

Learning Discriminative Relational Features for Sequence Labeling

submitted in partial fulfillment of the requirements
for the degree of

Doctor of Philosophy

of the
Indian Institute of Technology Bombay, India
and
Monash University, Australia
by

Naveen Sudhakaran Nair

Supervisors:

Prof. Ganesh Ramakrishnan (IIT Bombay)
Prof. Shonali Krishnaswamy (Monash University)



The course of study for this award was developed jointly by the Indian Institute of Technology Bombay, India and Monash University, Australia and given academic recognition by each of them. The programme was administered by The IITB-Monash Research Academy.

2013

To my Parents, Teachers and the Almighty

Thesis Approval

The thesis entitled

Learning Discriminative Relational Features for Sequence Labeling

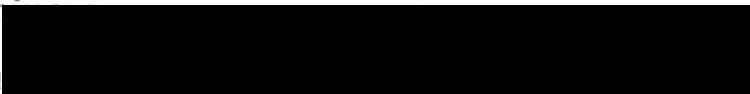
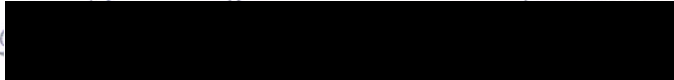
by

Naveen Sudhakaran Nair



is approved for the degree of

Doctor of Philosophy

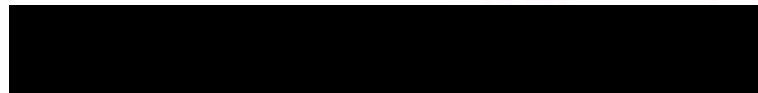
Exam

1. 
2. 
3. _____

Supervisors

1. 
2. 
3. _____
4. _____

Chairman



Date: 6 - DECEMBER - 2013

Place: IIT BOMBAY

Declaration

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute/the Academy and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Notice 1

Under the Copyright Act 1968, this thesis must be used only under normal conditions of scholarly fair dealing. In particular no results or conclusions should be extracted from it, nor should it be copied or closely paraphrased in whole or in part without the written consent of the author. Proper written acknowledgement should be made for any assistance obtained from this thesis.

Notice 2

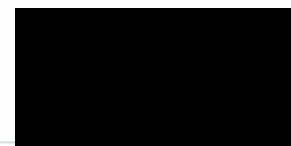
I certify that I have made all reasonable efforts to secure copyright permissions for third-party content included in this thesis and have not knowingly added copyright content to my work without the owners permission.

IIT BOMBAY

Place

6-DECEMBER-2013

Date



Signature



Name

Acknowledgments

The journey of acquiring a PhD degree, although self motivated work, cannot be accomplished without the help and guidance of many benevolent individuals around the researcher. I feel fortunate that I had many around me whose advices, insights, prayers and blessings made my thesis a reality. Its time that the great minds, who supported and encouraged me throughout this laborious journey, are appropriately thanked.

॥ माता पिता गुरु दैवम् ॥
mātā pitā guru daivam

It is my privilege to have a wonderful family who encouraged me tirelessly in my pursuit of education. My father, Sudhakaran Nair, who passed away during my childhood, had laid a strong foundation of education in me that helped me to excel in my studies. My mother, Anandavally Amma, was the one who supported me in every possible way during the course of my studies. I thank them for providing me a good background in education which has been a crucial milestone in my life. I also thank my wife, Chithra, who stood behind me during the course of PhD, and my son, Abhiraman, whose smiles bring happiness in me. I also thank my brother, in-laws, all other family members and my well wishers who have supported and prayed for my success.

"गुरु गोविंद दोऊ खड़े, किसको लागु पाय ।
बलिहारी गुरु आपने, गोविंद दिया बताय ॥"

"Guru govind dou khade, Kisako laagu paay |
Balihaari guru aapane, Govind diyaa bataay ||"

*"Guru and God both are here to whom should I first bow. All glory be unto the guru path to
God who did bestow". – Kabir Das*

No education is complete, unless it is practiced under the guidance of a teacher. I have been fortunate to have such profoundly knowledgeable and wonderful teachers to guide me throughout my research work. I consider this an opportune time to acknowledge their talent

and advices. At the outset, I would like to thank my advisors Prof. Ganesh Ramakrishnan and Prof. Shonali Krishnaswamy.

This thesis would not have been possible without continual inputs and suggestions from my Guru (Teacher) Prof. Ganesh Ramakrishnan. Prof. Ganesh Ramakrishnan, a man of immense energy and a true researcher, is also a dedicated teacher who is always inordinately interested in his students. He was the one who, with his persistent and invaluable comments, polished my intuitions and channeled them toward the right goals. He also encouraged me to enhance my communication skills. Apart from his academic life, in the midst of his busy schedule, he finds time for spiritual thoughts and philosophy and also advises his students to practice spirituality and to renounce the burdens of material life. I express my heartfelt gratitude to Prof. Ganesh Ramakrishnan for his guidance and support throughout my research work.

Prof. Shonali Krishnaswamy, a true professional and a very helpful guide, has showed immense patience in helping me to define the research problem and to solve diverse problem areas. She always gave me the freedom to choose a research path, while at the same time, directed it through the right path. I express my deepest gratitude to her for the guidance and support.

The next person I am indebted to is Prof. Krithi Ramakritham. A great mentor and a humble person, he always supported and inspired me to explore the possibilities of a wider research domain. He is always keen to discuss the details of my work to the finest granularity, which helped me to inch my work to perfection. I express my gratitude for his help and guidance which supported me in difficult times during the course of my PhD.

I also take extreme pleasure in expressing my thanks to Prof. Bala Srinivasan for his advice and support in my PhD work.

I take this opportunity to thank Prof. Mohan Krishnamurthy, CEO, IITB-Monash Research Academy, who always lent a helping hand in testing situations during the course of my PhD. He has been instrumental in improving my professional skills. I extend my sincere gratitude to him.

I also take this opportunity to thank my Research Progress Committee members, Prof. Saketha Nath and Dr. Mark Carman for their valuable comments which were vital to improving my research work. I would also like to thank all the examiners of my thesis for their valuable feedback. I thank Ms Sheba Sanjay, Communications and Training Services Consultant at IITB-Monash Research Academy, for doing the language review for my thesis.

I also thank all my teachers, from my school and college, for sharing their knowledge and inspiring me to pursue a long journey in the quest for knowledge.

I extend my gratitude to all the non academic staff and other professors at IIT Bombay, Monash University and IITB-Monash Research Academy, for sustaining a thriving echo system for research. My special thanks goes to the staff at IITB-Monash Research Academy, IIT Bombay, and Monash University for constantly reducing the burden on students in administrative matters. The most important names to be mentioned are Mrs. Mamta Bhattacharya, Mrs. Anasuya Banerji, Mrs. Parjat Chakraborty, Mr. Amrut Solanki, Mrs. Kuheli Mukerjee, Mr. Rahul Krishna, Mrs. Beena Pillai, Mr. Rahul Ingle, Mr. Bharat Ingle, Mr. Adrian Gertler, Mr. David Lau, Mr. Vijay Ambre, Mrs. Homsy Varghese, Mrs. Alpana Athavankar, Mrs. Nair, Mrs. Victor, Mrs. Subhadra, Mrs. Nandini Janardhan, Mrs. Sunanda Ghadge, Mr. Thushar, Mrs. Gaikwad, and Mrs. Gayathri.

I have been lucky to have a circle of friends, who were a great strength and help to me in my academic life. A special thanks to Ramkumar Rajendran, who supported and encouraged me a lot in my research. I also thank my other lab mates, Pratik Jawanpuria, Ajay Nagesh, Dipak Chowdary, Amrita Saha, and Dwi Rahayu, who were of great support in our joint endeavors. Other friends to be mentioned include Balamurali, Sharadwada Pan, Chitti Babu, Sidharth Gadkari, Bharat Padekar, Nilesh Raykar, Rupesh Bopade, Ramesh Kavali, Anu Thomas and Hariprasad Kodamana. I would always cherish the moments which I spent with them. I thank each and everyone for their support and help. There are so many other people who have influenced me throughout my academic career. I thank all of them.

Last but not least, I thank the Almighty for giving me the strength and the ability to undertake this long journey in the pursuit of knowledge. I conclude by quoting from 'The Gita':

कर्मण्येवाधिकारस्ते मा फलेषु कदाचन।
मा कर्मफलहेतुर्भूर्मा ते सङ्गोऽस्त्वकर्मणि।

"Let us continue to work without getting preoccupied and biased for positive results.

If our path is correct, and the intention novel, results will follow".

Date: _____

Naveen Sudhakaran Nair

Abstract

Sequence labeling is the task of assigning a class/state label to each instance in a sequence of observations; it is generally grouped under structured output classification problems. Typical sequence labeling algorithms learn probabilistic information about the neighboring states (along with the probabilistic information about the inputs¹) from the training data and find the globally best assignment for the entire query sequence at once. Hidden Markov Models (HMM) (Rabiner 1990), Conditional Random Fields (CRF) (Lafferty, McCallum & Pereira 2001) and Support Vector Machines on Structured Output Spaces (StructSVM) (Tsochantaridis et al. 2004) are some of the most popular sequence labeling approaches. All these models learn parameters for the state-observation relationships in a sequence and the transition relationships between states at successive steps. Inference is generally performed using a dynamic programming algorithm called the Viterbi algorithm (Forney 1973).

One of the problems in sequence labeling by conventional approaches is the limitation in discovering the interactions among inputs. Typical approaches tend to assume conditional independence between individual inputs, given the class label (van Kasteren et al. 2008). Although this enables a naive factorization of observation distribution, in several cases, where there are non-linear relationships among input variables, it results in loss of accuracy. Discovering the relational structure in input space could give a meaningful representation of the model and thereby improve the quality of the model in terms of labeling accuracy.

In this work, we propose to learn useful relational features that capture the relationships in input space. The space of relational features in such settings is exponential in the number of basic inputs. For instance, in the simple case of learning features that are conjunctions of basic inputs at any single sequence position, the feature space is of size

¹We use the terms inputs, emissions and observations interchangeably for input variables

2^N for N basic inputs. The size would be much larger if we consider complex relational features built from inputs at different relative positions. Since an exhaustive search in this exponentially large feature space is infeasible, most of the relational feature learning systems for sequence labeling such as tildeCRF follow a greedy search strategy. In this thesis, we study the possibility of efficiently learning and using discriminative relational features for sequence labeling. We pose the problem as learning relational features/rules in the form of definite clauses. For this, we identify classes of relational features based on their complexities and develop efficient learning approaches for those feature classes that we identify as relevant and useful.

We first investigate the problem of learning simple conjunctions of basic (propositional) input features for any given position in a sequence. This type of features is referred to as Simple Conjuncts (\mathcal{SC}). We start with developing a greedy feature induction approach for sequence labeling. Our greedy feature induction approach incrementally discovers the best model by employing a greedy hill climbing search in the space of features. In each iteration of the search, we derive a candidate model from the previous model, combine it with transition rules, evaluate in a custom implementation of HMM, prune low scoring candidate models (and their refinements) and select the best scoring model. There have been a few other approaches similar to our approach, but in different learning settings, that learn composite features for sequence labeling (McCallum 2003, Gutmann & Kersting 2006, Stewart et al. 2008). Although these approaches give better performance than conventional approaches, being greedy, they cannot guarantee optimal solutions. We therefore propose and develop a Hierarchical Kernels based approach for learning optimal \mathcal{SC} s relevant for each output label.

The Hierarchical Kernels approach, referred to as Hierarchical Kernel Learning for Structured Output Spaces (StructHKL), optimally and efficiently explores the hierarchical structure in the feature space for problems with structured output spaces such as sequence labeling. Here we extend the Hierarchical Kernel Learning (HKL) approach, originally introduced by Bach (2009) and Jawanpuria et al. (2011), to learn feature conjunctions for multi-class structured output classification. We build on the Support Vector Machines for Structured Output Spaces (StructSVM) model (Tsochantaridis et al. 2004, Tsochantaridis 2006) for sequence prediction problems and employ a ρ -norm hierarchical regularizer (Jawanpuria et al. 2011) for input/observation features and a conventional

2-norm regularizer for the state transition features. The hierarchical regularizer penalizes large features and thereby selects a small set of short features. StructHKL learns the input features and their weights simultaneously in an efficient way. We now look into the problem of learning complex relational features that are derived from inputs at multiple sequence positions.

Although the StructHKL algorithm optimally solves the objective of learning the most discriminative \mathcal{SC} s for sequence labeling, due to some theoretical requirements of the feature space, its applicability in learning complex relational features, that are derived from inputs at different relative positions, is non-trivial and challenging. Therefore, we determine feature classes that can be composed to yield complex ones, with the goal of formulating efficient yet effective relational feature learning procedures. We identify a self-contained class of features called Absolute Features (\mathcal{AF}), whose (unary/multiple) compositions yield complex relational features in another class called Composite Features (\mathcal{CF}). We seek to leverage optimal feature learning in all the steps of relational feature induction, which can be addressed either by (i) enumerating \mathcal{AF} s and discovering their compositions (\mathcal{CF}) using StructHKL or by (ii) developing methods to learn optimal \mathcal{AF} s (or \mathcal{CF} s directly).

As for the first option, the space of \mathcal{AF} s is prohibitively large, which makes enumeration in that space impractical. We thus selectively filter \mathcal{AF} s based on some relevance criteria (minimum support) and then make use of the StructHKL algorithm to learn compositions of selected features. However, the partial ordering of \mathcal{AF} s does not comply with the requirement of StructHKL that the descendant kernels in the partial ordering of features should be summable in polynomial time. Consequently, leveraging StructHKL to optimally learn features in the space of \mathcal{AF} s (and its super-space of \mathcal{CF} s) is infeasible.

For the second option to learn optimal \mathcal{CF} s directly, in the structured output classification model, we leverage a relational kernel that computes the similarity between instances in an implicit feature space of \mathcal{CF} s. To this end, we employ the relational subsequence kernel (Bunescu & Mooney 2006) at each sequence position (over a time window of inputs around the pivot position) for the classification model. While this way of modeling does not result in interpretability, relational subsequence kernels do efficiently capture the relational sequential information on the inputs. Although the main contribution of the thesis is feature learning for sequence labeling, we have also contributed in two related

problem domains, which we briefly introduce in the following paragraphs.

In general classification settings (with or without structured outputs), where it is not feasible to ground all variables, dynamic programming approaches have limitations in performing inference. We now derive a Satisfiability approach for fast and memory efficient inference in general horn clause settings, which prunes a major part of the possible groundings and performs inference in a small restricted space. Our approach finds a model in polynomial time, if it exists; otherwise finds a most likely interpretation given the evidence. We now briefly introduce our second related contribution, which is performing dimensionality reduction in classification settings by leveraging Hierarchical Kernel Learning.

Many real world classification problems are characterized by a large set of features that possibly contain a non-trivial amount of redundant and irrelevant information. Using the entire feature space as it is often leads to over-fitting and therefore less effective models. Dimensionality reduction techniques are typically used to reduce the dimension of the data either by projecting the features onto a collapsed space or by selecting a subset of features, both as preprocessing steps. These approaches suffer from the drawback that the dimensionality reduction objective and the objective for classifier training are decoupled (performed one after the other) and often, the approach for dimensionality reduction is greedy. A few approaches have been recently proposed to address the two tasks in a combined manner by attempting to solve an upper-bound to a single objective function (Zhu et al. 2010, Xu 2010). However, the main drawback of these methods is that the number of reduced dimensions is not learned, but taken as an input to the system. In this work, we propose an integrated learning approach for non-parametric dimension reduction by projecting the features from the original feature space to the space of disjunctions and discovering a sparse set of important disjunctions out of them. Here, in order to discover good disjunctive features, hierarchical kernels have been employed that efficiently and optimally perform feature selection and classifier training simultaneously in a maximum margin framework.

We demonstrate the efficiency of our feature induction approaches in improving prediction accuracy in the domain of activity recognition. The proposed satisfiability based inference approach and the dimensionality reduction approach are also evaluated on standard datasets.

Publications

Conference Papers

1. Naveen Nair, Amrita Saha, Ganesh Ramakrishnan, and Shonali Krishnaswamy. *Rule Ensemble Learning using Hierarchical Kernels on Structured Output Spaces*, In **26th AAAI Conference on Artificial Intelligence, 2012**, Toronto, Canada, Pages 1061-1067, AAAI Press, Palo Alto, Calif, ISBN: 1577355687.
2. Naveen Nair, Ajay Nagesh, and Ganesh Ramakrishnan. *Probing the Space of Optimal Markov Logic Networks for Sequence Labeling*, **22nd International Conference on Inductive Logic Programming, 2012**, Dubrovnik, Croatia, Pages 193-208, Springer Berlin Heidelberg, LNCS Vol 7842, 2013, ISBN: 978-3-642-38811-8.
3. Ajay Nagesh, Naveen Nair, and Ganesh Ramakrishnan. *Comparison between Explicit Learning and Implicit Modeling of Relational Features in Structured Output Spaces*, **23rd International Conference on Inductive Logic Programming, 2013**, Rio de Janeiro, Brazil.
4. Naveen Nair, Ganesh Ramakrishnan, and Shonali Krishnaswamy. *Enhancing Activity Recognition in Smart Homes Using Feature Induction*, In **13th International Conference on Data Warehousing and Knowledge Discovery (DaWaK), 2011**, Toulouse, France. Pages 406-418, Springer-Verlag Berlin, Heidelberg, LNCS Vol 6862, 2011 ISBN: 978-3-642-23543-6
5. Naveen Nair, Anandraj Govindan, Chander Iyer, Kiran TVS, and Ganesh Ramakrishnan. *Pruning Search Space for Weighted First Order Horn Clause Satisfiability*, In **20th International Conference on Inductive Logic Programming, 2010**, Florence, Italy. Pages 171-180, Springer-Verlag Berlin, Heidelberg, LNCS Vol 6489, 2011 ISBN: 978-3-642-21294-9

6. Amrita Saha, Naveen Nair and Ganesh Ramakrishnan, *Optimally Extracting Discriminative Disjunctive Features for Dimensionality Reduction*, Accepted for **International Conference on Management of Data, 2013**.

Journal Papers

1. Naveen Nair, Ajay Nagesh and Ganesh Ramakrishnan, *Learning Discriminative Relational Features for Sequence Labeling*, Invited to **Machine Learning Journal Special Issue on ILP**.

Workshop Papers

1. Naveen Nair, Amrita Saha, Ganesh Ramakrishnan, and Shonali Krishnaswamy, *Challenges in Learning Optimum Models for Complex First Order Activity Recognition Settings*, In **AAAI workshop on Activity Context Representation , 2012**, Toronto, Canada.

Competitions, Awards & Other Achievements

1. **First place in Task C** (recognition of arm gestures from noisy data in a sensor rich environment) of **Opportunity, 2011** - a bench marking contest for **activity recognition** conducted by Future and Emerging Technologies Open Call project of the **European Commission**.
2. Received **Microsoft Research India student travel grant** to attend **AAAI-2012** held at Toronto, Canada.
3. Presented the idea, **A Louder Scream: Using Mobile Phones to Call for Attention in Isolated Places**, *Naveen Nair*, at **IBM I-CARE colloquium, 2011**, Delhi, India.

Contents

	Page
Abstract	vi
List of Tables	xv
List of Figures	xvii
1 Introduction	1
1.1 Introduction to Sequence Labeling	1
1.2 Motivation	3
1.3 Objective	5
1.4 Contribution	10
1.4.1 Learning Discriminative Relational Features for Sequence Labeling .	11
1.4.2 Pruning Search Space for Weighted First Order Horn Clause Satisfiability	16
1.4.3 Optimally Extracting Discriminative Disjunctive Features for Dimensionality Reduction	16
1.5 Thesis structure	18
2 Related Work	20
2.1 Models for Sequence Labeling	20
2.1.1 Probability Based Sequence Labeling techniques	21
2.1.2 Max-Margin Methods for Sequence Labeling	21
2.2 Learning Relationships as Features	23
2.2.1 Greedy Feature Induction Approaches	23
2.2.2 Optimal Feature Induction for Binary Classification	24
2.3 Inference	26

2.3.1	The Viterbi Algorithm	26
2.4	Recent developments in Activity Recognition	27
3	Learning Discriminative Relational Features for Sequence Labeling	29
3.1	Introduction	29
3.2	First Order Definite Features	33
3.3	Greedy Feature Induction for Sequence Labeling	38
3.3.1	Logical Coverage Based Feature Induction for HMM	40
3.3.2	Probabilistic Feature Induction for HMM	41
3.4	Hierarchical Kernel Learning for Structured Output Spaces	44
3.5	Learning Complex Relational Features for Sequence Labeling	51
3.5.1	Constructing Composite Features from Enumerated Absolute Features	52
3.5.2	Leveraging Complex Relational Kernels for Sequence Labeling	52
4	Pruning Search Space for Weighted First Order Horn Clause Satisfiability	56
4.1	Background	56
4.2	Satisfiability in Horn Clauses	58
4.2.1	T_{Σ} Operator	59
4.2.2	<i>Modified</i> T_{Σ} Step	59
4.3	Modified_Weighted_SAT	60
5	Optimally Extracting Discriminative Disjunctive Features for Dimensionality Reduction	65
5.1	Introduction	66
5.1.1	Our Contribution	69
5.2	Optimal Non-Parametric Max Margin Dimensionality Reduction	71
5.2.1	Formal Specification of the Problem	73
6	Experiments and Results	79
6.1	Learning Discriminative Features for Sequence Labeling	79
6.1.1	Greedy Feature Induction for Sequence Labeling	80

6.1.2	Optimal Feature Induction using Hierarchical Kernels for Sequence Labeling	83
6.1.3	Learning Complex Relational Features for Sequence Labeling	89
6.2	Pruning Search Space for Satisfiability in Weighted Horn Clauses	92
6.3	Optimally Extracting Discriminative Disjunctive Features for Dimensionality Reduction	99
6.3.1	UCI data	99
6.3.2	20 Newsgroups data	102
7	Conclusion	105
	Appendix	109
A	Derivations and Proofs	109
A.1	Sufficiency Condition	109
A.2	Solution to the Reduced Problem	114
A.3	Kernels in StructHKL	115
A.4	Cutting Plane Algorithm	116
	Bibliography	117

List of Tables

Table		Page
6.1	Micro average accuracy and macro average accuracy of classification in percentage using standard HMM, B&B learning assisted HMM and greedy feature induction assisted HMM (macro-average and micro-average accuracies as scoring function separately) on UA dataset with 28 fold cross validation.	82
6.2	Micro average accuracy and macro average accuracy of classification in percentage using standard HMM, B&B learning assisted HMM, greedy feature induction assisted HMM, StructSVM, CRF, CRF with feature induction, RELHKL without transitions, RELHKL + StructSVM and the proposed StructHKL approach on UA dataset.	85
6.3	Micro average accuracy and macro average accuracy of classification in percentage using StructSVM, CRF, CRF with feature induction and the proposed StructHKL approach on PlaceLab dataset. (Std.HMM, B&B HMM, Greedy FIHMM, and RELHKL without transitions either failed to give comparable results or did not converge)	86
6.4	Micro average accuracy and macro average accuracy of classification in percentage using tildeCRF, \mathcal{CF} from enumerated \mathcal{AF} approach (enum \mathcal{AF}), StructSVM (with basic inputs as features) and relational subsequence Kernels for StructSVM approach (SubseqSVM) on UA data (exploring the space of \mathcal{CF} s).	91

6.5	Micro average accuracy and macro average accuracy of classification in percentage using various approaches on KU data. As a single sequence step in this data has only one input feature, the feature space is not rich enough to evaluate the efficiency of our approaches. For this reason, the performance of our approaches is slightly inferior, as observable from the table.	93
6.6	progression of sequence labeling results on UA data based on feature categories.	94
6.7	Comparison of number of groundings after i) Complete grounding and ii) Pruning on uwce knowledge base.	95
6.8	Performance comparison of satisfiability approaches on uwce knowledge base and different evidence sets.	95
6.9	Micro average accuracy and macro average accuracy of classification in percentage using two methods of inference, that is, the Viterbi algorithm and the HornSAT. A model is first trained using StructHKL and the two experiments for inference are performed. The time taken for inference for each of the approaches are also reported.	98
6.10	Comparison of accuracies (in percentage) of different dimensionality reduction approaches on the UCI dataset and the 20Newsgroups <i>alt.atheism</i> vs. <i>talk.religion.misc</i> problem. Here (\mathcal{L}_2) and (\mathcal{L}_1) refer to 2-norm and 1-norm SVM employed over the set of features selected by the Weka Feature Subset Selection Wrappers. For <i>IntegratedDim.Red.</i> the <i>Accuracy \pm RMS Error over Cross Validation Folds</i> has been reported. Baseline \mathcal{L}_2 and \mathcal{L}_1 refer to 2-norm and 1-norm SVM respectively, without applying any feature subset selection	100
6.11	Comparison of accuracies of different approaches on 20 <i>Newsgroups</i> dataset (MedLDA accuracy is not exactly reported in (Zhu et al. 2010) and therefore, it has been calculated from the relative improvement ratios reported in (Xu 2010). The results of the competitor approaches are the best ones obtained by the authors by cross validating on the parameter, number of topics.).	101

List of Figures

Figure		Page
1.1	Graphical representation of an HMM. The rectangles (blue) represent hidden states and the ellipses (orange) represent observable variables. y^t and x^t represent the output label and the input feature, respectively, at time t .	5
1.2	Multiple inputs at each time step.	8
1.3	Conjunctive features for sequence labeling. The enclosing ellipses depict conjunctions of the basic inputs represented by the enclosed ellipses. Only three activities are shown for simplicity.	9
1.4	Lattice of propositional conjunctive features related to a single label. . . .	10
1.5	Lattice of propositional features in a multi-label setting. Each sub-lattice is the space of features belonging to a particular label.	11
3.1	Subset relationships among various categories of features.	36
3.2	Feature induction assisted HMM model training	43
3.3	Active set algorithm for solving StructHKL.	50
3.4	Cutting plane algorithm for solving dual with a fixed $\boldsymbol{\eta}$	50
4.1	<i>Modified-T_Σ</i> algorithm	61
4.2	Modified_Weighted_MaxSAT algorithm	63
4.3	Weighted_HornSAT algorithm	64
5.1	Active set algorithm	77
6.1	Performance comparison of standard HMM, B&B learning assisted HMM and greedy feature induction assisted HMM with scoring function defined by macro-average accuracy (Greedy (macro)) and micro-average accuracy (Greedy (micro)) on UA dataset.	82

6.2	Performance comparison of different approaches on UA dataset.	86
6.3	Performance comparison of different approaches on PlaceLab subject one data (Std.HMM, B&B HMM, Greedy FIHMM, and RELHKL without transitions either gave worse results or did not converge).	87
6.4	Performance comparison of different approaches on PlaceLab subject two data (Std.HMM, B&B HMM, Greedy FIHMM, and RELHKL without transitions either gave worse results or did not converge).	88
6.5	Performance comparison of different approaches on UA data (exploring the space of \mathcal{CF} s).	92
6.6	Performance comparison of different approaches on KU data (exploring the space of \mathcal{CF} s).	93
6.7	progression of sequence labeling results on UA data based on feature categories.	94
6.8	Performance comparison of different satisfiability approaches on uwcse language KB. All 181 atoms are given as evidence.	96
6.9	Performance comparison of different satisfiability approaches on uwcse language KB. 87 atoms are given as evidence.	96
6.10	Performance comparison of different satisfiability approaches on uwcse AI KB. All 766 atoms are given as evidence. In this experiment, complete grounding resulted in an out-of-memory error	97
6.11	Performance comparison of the two inference approaches on activity recognition data (sequence labeling).	98

Chapter 1

Introduction

Our research objective is to efficiently learn non trivial input features for sequence labeling problems, and thereby improve the accuracy of labeling. We first give a brief overview of sequence labeling.

1.1 Introduction to Sequence Labeling

Structured output classification has gained profound interest in the machine learning community during the last decade (Tsochantaridis et al. 2004, Blaschko & Lampert 2012, Liu et al. 2012, Joachims et al. 2009, Taskar et al. 2006, Miao & Rao 2012, Zaki & Aggarwal 2006, Bo & Sminchisescu 2010). The goal of such work is to classify complex output structures such as sequences, trees, lattices or graphs, wherein the class label at each node/position of the structure has to be inferred based on observed evidence data. The possible space of structured outputs tends to be exponential and thus structured output classification is a challenging research task. We, in our research, focus on a specific structured output classification problem called sequence labeling. As in any classification setting, the sequence labeling domain also has complex relationships among entities, with uncertainties in these relationships. Accurate models can be constructed by exploiting these relationships. However, discovering relationships that enhance the discriminative power of classifiers is a hard task, since the relationship space is often too large. Conventional approaches restricts the space of relationships to transition relationships (between labels at neighboring positions) and emission relationships (label to observations at single time step) for computational simplicity. In domains that have multiple inputs with non-linear relationships, typical approaches either ignore the complex relationships or use heuristics to learn the relationships. In this work, we focus on exploiting complex

relationships in both the input as well as the output space in an efficient way in order to improve sequence labeling models. We begin with a brief introduction to the task of sequence labeling.

Sequence labeling is the task of assigning a class/state label to each instance in a sequence of observations. Typical applications of sequence labeling include activity recognition, natural language processing, bio-informatics and others. Activity recognition is the main motivating application domain considered for this research. In activity recognition systems for monitoring user activities (Wilson 2005), the activities being performed by a subject tend to follow a sequence structure. For instance, the subject is likely to have his lunch or dinner after cooking. A typical non-intrusive activity recognition setting consists of sensing devices, a processing unit, and the algorithms for learning and inference (van Kasteren et al. 2008). The objective, in such a setting, is to assess the sequence of activities performed by the subject based on the sensor observations. In the training phase, activities performed are manually annotated and the sensor readings are recorded. Probabilistic models are learned from the training data (van Kasteren et al. 2008). The trained model is later employed to infer the sequence of (hidden) activities from new sensor observations. Similarly, in the natural language processing task of part-of-speech tagging, a natural language sentence is viewed as a sequence of words with features extracted for each word/position (McCallum 2003). The objective is to assign a part-of-speech label to each word in the sentence. The words in the sentence are related to their neighboring words and thus the problem can be posed as a sequence labeling problem. For instance, a word’s part-of-speech is likely to be a noun if the preceding word is an article. A probabilistic model is trained from the labeled training data and later used for inferring the part-of-speech of words in unseen sentences. Sequence inferring approaches are also used in bio-informatics for discovering genetic sequences in genome analysis.

Although it is possible to classify each member in the sequence separately, it has been shown that incorporating the statistical influence of states in the neighborhood improves the accuracy of labeling (Rabiner 1990, Lafferty, McCallum & Pereir 2001). For instance, the activity at current time step can be better inferred if the activity at the previous time step is known. Another instance is that the part-of-speech of a word can be better inferred if the part-of-speech of neighboring words are known. Therefore, typical sequence labeling algorithms learn probabilistic information about the neighboring states

along with the probabilistic information about the input relationships from the training data. For inference, the globally best assignment for the entire sequence is found at once, which helps to resolve ambiguity in assigning labels. For example, assigning part-of-speech labels for words that have the same spelling, but belong to a different part-of-speech. Since sequence labeling approaches assign labels to an entire sequence at once, it is possible to assign an optimal label to each position that does not conflict with the neighborhood. For example, if a word can be either a noun or a verb, a preceding article will make the noun to be a more probable assignment. Hidden Markov Models (HMM) (Rabiner 1990) and Conditional Random Fields (CRF) (Lafferty, McCallum & Pereira 2001) are two conventional approaches popularly used in sequence labeling tasks. These models capture the state-input relationships (input/emission dependency) at each step and the transition relationships between states in successive steps. Support Vector Machines on Structured Output Spaces (StructSVM) (Tsochantaridis et al. 2004) is a large margin approach that is used popularly for sequence classification. Inference is generally performed using a dynamic programming algorithm called the Viterbi algorithm (Forney 1973).

In our research, we propose and develop approaches and algorithms to improve the prediction accuracy in sequence labeling. We evaluate our approaches on publicly available activity recognition datasets.

1.2 Motivation

In many domains of sequence labeling problems, there could be multiple inputs at each sequence position, which are typically represented as a vector with each individual input as an element. Therefore, at any particular sequence step, the current label should be determined using the vector of all individual inputs (jointly). However, since it is impractical to assume this vector as a single variable, due to its exponential size (2^N for N binary individual inputs), conventional approaches tend to ignore the information in the joint state of inputs. However, there may be vital non-linear interactions among inputs in many real world problems and therefore ignoring the joint state information or the input structure could lead to loss of accuracy. However, since the space of non-linear interactions/input structure is typically exponential in the number of basic inputs, discovering useful features from such a space is a challenging task. Conventional sequence labeling approaches such as the Hidden Markov Models (HMMs), the Conditional Random Fields

(CRFs), and the Support Vector Machines on Structured Output Spaces (StructSVM) have limitations in discovering the input structure. Our work focuses on discovering the input structure in sequence labeling settings. We also eliminate redundant or irrelevant basic inputs, which is an overhead in large settings. In the paragraphs that follow, we describe one of our motivating application domains, activity recognition, to exemplify the motivation. Since activity recognition domains have sparse, skewed and noisy¹ data, learning in such a setting is challenging. Therefore we discuss our approaches on activity recognition data.

Activity recognition systems help to monitor activities of users in domicile environments. An example area is monitoring the daily activities of elderly people living alone, in order to estimate their health condition (Wilson 2005, van Kasteren et al. 2008, Gibson et al. 2008), since the patterns of daily activities are indicators of health conditions. Such non-intrusive settings typically have on/off sensors installed at various locations in a house. Binary sensor values are recorded at regular time intervals. The joint state of these sensor values at each time step forms our observation/emission. The user activity at each time step forms the hidden state/label. The history of sensor readings and the corresponding activities (as manually identified) can be used to train prediction models such as the Hidden Markov Model (HMM) (Rabiner 1990), the Conditional Random Field (CRF) (Lafferty, McCallum & Pereira 2001) or StructSVM (Tsochantaridis 2006, Tsochantaridis et al. 2004), which could be later used to predict activities based on sensor observations. These approaches typically assume that the output label at a particular time is independent of all previous labels given the labels in the neighborhood and that the observation at any particular time step is independent of all other variables given the label at that time step. Figure 1.1 illustrates the independence assumption in an HMM setting. Prediction involves determining the label (activity) sequence that best explains the observation (joint state of sensors) sequence, for which dynamic programming (Forney 1973) is typically used.

Activity recognition datasets tend to be sparse; that is, one could expect very few sensors to be on at any given time instance². Moreover, in a setting such as activity recog-

¹Noise due to faulty sensors, communication lines and non uniform patterns adopted by subjects to perform tasks.

²Activity recognition domain, which has highly skewed, sparse and noisy data, is one of the challenging areas for sequence labeling. Therefore, we consider activity recognition as our application domain. Our

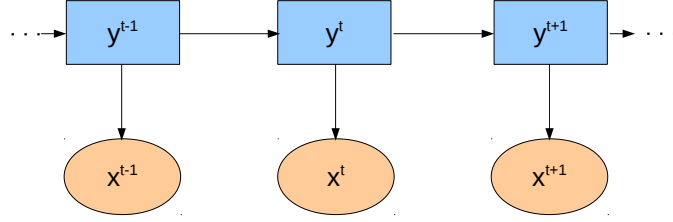


Figure 1.1: Graphical representation of an HMM. The rectangles (blue) represent hidden states and the ellipses (orange) represent observable variables. y^t and x^t represent the output label and the input feature, respectively, at time t .

dition, one can expect certain combinations of (sensor) readings to be directly indicative of certain activities. For example, sensors at *microwave* and *groceries cupboard* relate to the activity *cooking*. Another example is that firing of the *microwave* sensor followed by firing of the *plates cupboard* sensor relate to the activity *dining*. HMMs, CRFs and StructSVM attempt to capture these relations indirectly³. Our focus is to learn these relationships automatically in the form of relational features, which alleviates the problems of exponential input space and naive factorization, viz., determining a mapping between each label and the relational features derived from individual inputs at the same sequence position as well as relative sequence positions. There have been a few other approaches that learn relationships among input features at a sequence position (McCallum 2003, Gutmann & Kersting 2006, Stewart et al. 2008). However, these approaches are greedy, whereas our focus is to learn optimal features.

1.3 Objective

Real world problems have to deal with complex relationships among entities and the uncertainties in these relationships. Learning and prediction systems in such domains should have adequate techniques to understand, represent and use the relationships as well as deal with the uncertainties. These complex relationships are quite often represented in approaches work on other domains such as Natural Language Processing, which has relatively less sparse, skewed and noisy data.

³For example, the sequence relationships among inputs at different positions are indirectly captured in HMMs through the label sequence.

the form of relational clauses or features and the uncertainties are typically learned in the form of probabilities or probabilistic weights. Sequence labeling is one of the classification settings, where such relationships and uncertainties exist. In our research, we exploit such complex relationships, in the form of relational features, to improve the efficiency of sequence labeling models. We start with the objectives of conventional sequence labeling models and then formally state our objective.

The objective in a sequence labeling task is to assign a class label to each instance in a sequence of inputs/observations. Typical sequence labeling algorithms learn probabilistic information (probabilities or probabilistic weights) about the neighboring states along with the probabilistic information about the inputs/observations. HMM, CRF, StructSVM and other sequence labeling models follow this, with different conventions for representing parameters. In HMMs, the score is the joint probability distribution of input and output sequences. From the independence assumptions as illustrated in Figure 1.1, one can factorize the joint probability distribution of the sequence of inputs and labels into three factors: the initial state distribution, the transition distribution, and the emission distribution (Rabiner 1990). These parameters are learned by maximizing the joint probability of the paired input and label sequences in the training data.

In CRFs (Lafferty, McCallum & Pereira 2001), parameters that maximize the conditional probability of a sequence of states given a sequence of inputs are learned. These parameters are later used to identify the (hidden) label sequence that best explains a given sequence of inputs/observations.

StructSVM (Tsochantaridis et al. 2004, Tsochantaridis 2006) is a maximum margin framework for structured output spaces such as sequence labeling. It generalizes the standard Support Vector Machines (SVM) with the margin defined as the difference in the likelihood scores of the original output sequence with any other possible output sequence.

In general, the objective of learning sequence labeling models is to learn feature weights that make the score of the true output sequence greater than any other possible output sequence, given an input sequence (Tsochantaridis et al. 2004, Tsochantaridis 2006).

We now discuss the limitations of typical sequence labeling approaches and then explain our objective of discovering the input structure in sequence labeling tasks. For simplicity of exposition, we discuss the limitations in the simple setting of HMM.

In an HMM set-up with multiple possible observations, the observation/input variable is a vector of basic individual inputs. An example from the activity recognition domain is shown in Figure 1.2. This results in an exponential input space which is computationally feasible only in small settings. Providing tuples consisting of individual inputs to the model is a solution to this. However, discovering such tuples is challenging and popular approaches such as HMM, CRF and StructSVM have limitations in learning such relations. Therefore, often independence is assumed among individual inputs, given a label, to simplify the representation and computation. However, in complex problem settings, where there are non-linear interactions among input variables, assuming conditional independence would affect the performance in terms of accuracy of labeling. To alleviate both the issues of exponential input space and independence assumption, we identify the need to find a mapping between labels and their relevant compositions of dependent inputs⁴. The objective is to automatically discover features that categorize each label. These label specific features can be basic inputs or features derived from basic inputs. For example, in activity recognition, a few dependent sensors in conjunction with information regarding the previous activity may jointly decide whether an activity has happened in the current time. That is, we avoid the non relevant inputs and use useful relevant inputs or their compositions to improve the prediction accuracy. This also helps to reduce the effect of noise while doing inference. Figure 1.3 illustrates an example of the model desired in the domain of activity recognition, where features are constructed from activity specific conjunctions of sensors and all the transition relations. Only three labels are assumed for simplicity of exposition.

We propose learning the emission structure in the form of relational features that maximize probabilistic coverage (probability by which examples are covered) of the training data. In our problem, since we assume that all the transitions are important, the model learned should allow all inter state transitions. Moreover, since the support for transition rules/features is much higher than that for emission rules/features, the set of rules returned when emission and transition rules are learned together would be dominated by the transition rules. Therefore, we learn the structure of emission distribution while preserving all the n^2 transition probabilities, where n is the number of labels.

In the previous paragraphs, we identified the need for finding the relational structure

⁴This work has been presented at DaWaK, 2011

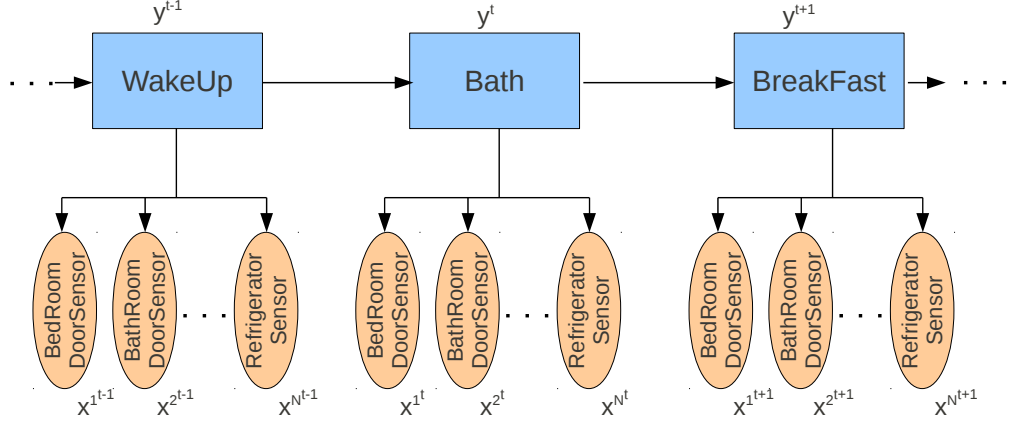


Figure 1.2: Multiple inputs at each time step.

among input features and mapping them to corresponding labels. However, the space of relational features is exponential in the number of basic inputs, making the discovery of useful features a difficult task. For instance, in the simple case of learning features that are conjunctions of basic inputs at any single sequence position, the feature space is of size 2^N for N basic inputs. The space of such features follows an ordering (lattice⁵). In multi-label settings, the size of feature space gets multiplied by the number of labels. A feature lattice for a single label in an activity recognition setting with four sensors is illustrated in Figure 1.4. Figure 1.5 shows the feature lattice in a multi-label case with three labels and four sensors. In complex relational settings that capture relationships between input attributes in relative sequence positions, features can be constructed from conjunctions and unifications of ground/first order predicates. Each first order predicate is a place holder for a group of ground predicates. A detailed discussion on first order predicates will ensue in another chapter. Therefore, the space of relational features is much larger than the simpler propositional setting. Discovering useful emission features for sequence labeling from this large space is a challenging problem. An exhaustive search is infeasible in real world settings. In this thesis, we investigate the problem of efficiently learning propositional and relational features for sequence labeling.

In the previous paragraphs, we identified the need to discover label specific relational

⁵A lattice is a partially ordered set which has a least upper bound and a greatest lower bound defined on every pair of nodes/elements

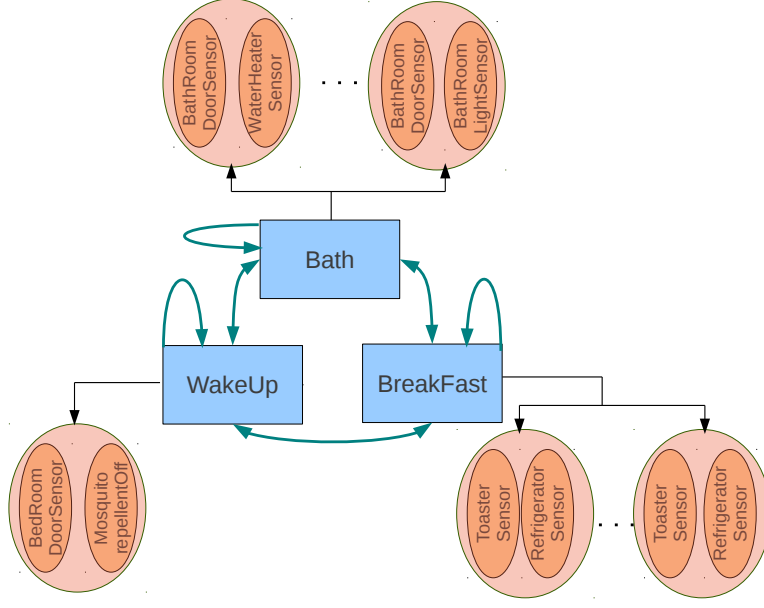


Figure 1.3: Conjunctive features for sequence labeling. The enclosing ellipses depict conjunctions of the basic inputs represented by the enclosed ellipses. Only three activities are shown for simplicity.

features for sequence labeling. Manual imposition of such a mapping is neither novel nor feasible in large settings. An efficient feature induction approach that can automatically capture this mapping has to be employed. We define features or rules to represent label specific compositions of inputs. In relational sequence labeling problems, the rules are first order logical statements and are constructed by conjoining and unifying predicates. For example,

```
prepareDinner(t1) :- microwave(t1), prevRelPosWindowNear(t1,t2),
                                                             platesCupboard(t2)
```

is a first order relation in the form of a definite clause⁶, where the variables $t1$ and $t2$ are shared among predicates. The rule above states that the activity at time $t1$ is `prepareDinner` if the sensor fired at $t1$ is the `microwave` sensor and there is a previous time instance $t2$ near $t1$ such that a `platesCupboard` sensor is fired at $t2$. Therefore the rules/features in first order are not only from conjunctions⁷. In this work, we restrict our discussion to function-free definite features. Since a class specific feature can be

⁶A definite clause is a first order logical statement where all the atoms are negated except one (head).

⁷Although we can theoretically ground the predicates and write conjunctive rules for first order relations, it is not feasible in real world problems.

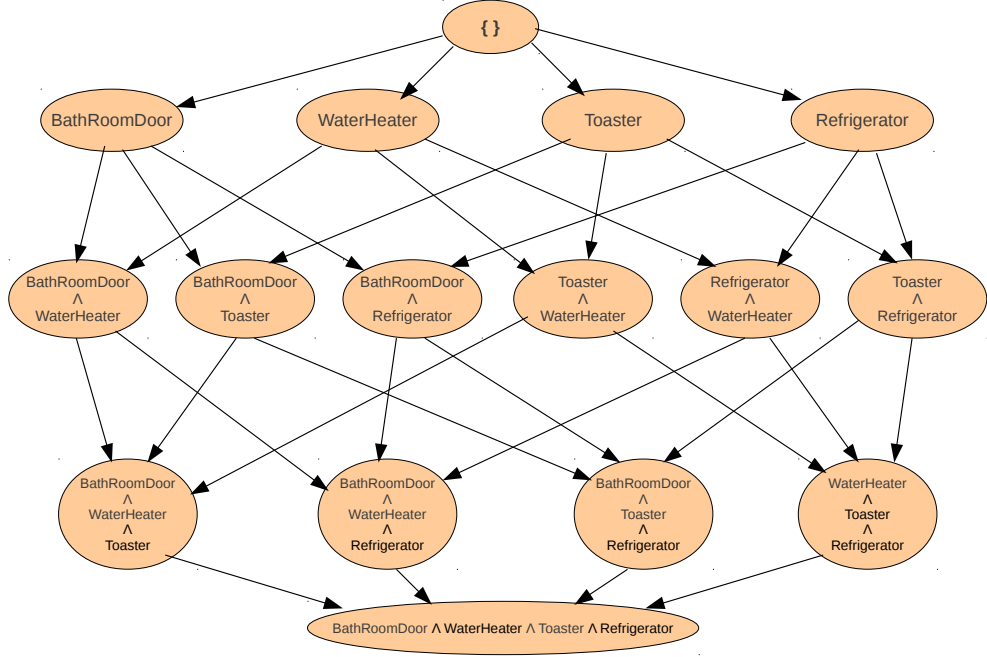


Figure 1.4: Lattice of propositional conjunctive features related to a single label.

constructed by conjoining the body literals of a definite clause/rule whose head depicts the class label, we use the terms definite clause and feature interchangeably. In this thesis, we first propose and develop a heuristic based greedy search algorithm to discover useful features for sequence labeling. We then investigate the possibility of learning optimal relational features for sequence labeling. On this account, we first categorize relational features based on their complexity and develop optimal learning approaches for those categories we identify as useful and tractable. We present a solution to deal with the problems of naive factorization and the exponential input space that we discussed in the previous sections. We now move to the main contributions of this thesis.

1.4 Contribution

In the previous sections, we posed the problem of efficiently learning relational input features for sequence labeling. Our objective is to improve prediction accuracy by learning discriminative relational features. However, since the feature space is exponentially large, discovering useful features is a challenging task. Here, we briefly introduce our solutions to the problem. We also discuss here our contributions in two related areas *viz.* weighted

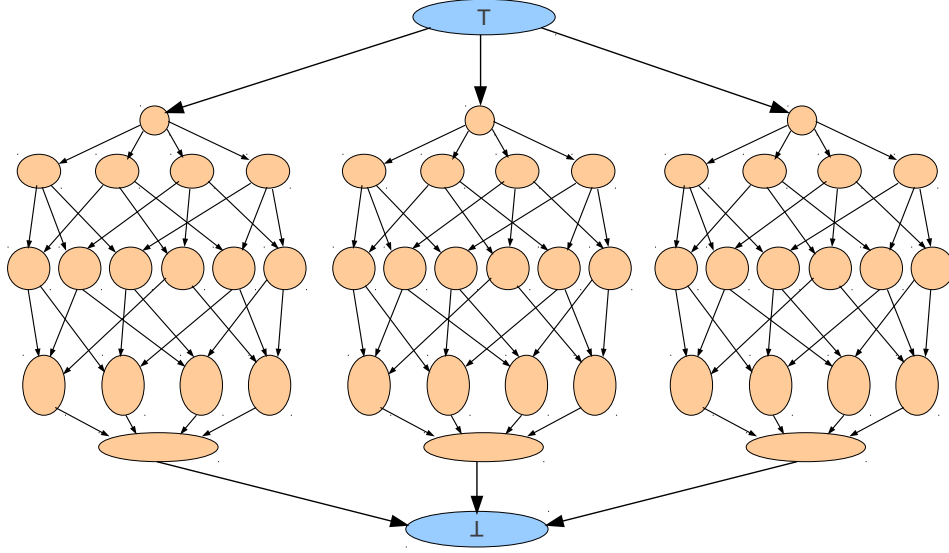


Figure 1.5: Lattice of propositional features in a multi-label setting. Each sub-lattice is the space of features belonging to a particular label.

satisfiability in first order horn clause settings and integrated supervised dimensionality reduction using hierarchical kernel learning.

1.4.1 Learning Discriminative Relational Features for Sequence Labeling

We intend to learn discriminative features that discern classes or labels, which can be viewed as decision rules of the form *if condition then decision* (Rivest 1987). The condition consists of the composition of a small number of simple boolean statements concerning the values of the individual input variables. This can essentially represent relational features derived from individual inputs at single or multiple sequence positions. The decision part specifies the function/class to be learned. This type of rules with a single boolean predicate as the decision variable generally is referred to as a definite clause/rule. A class/label specific feature can be constructed from the condition part of the definite clause/rule whose head depicts the class label. Therefore, we use the terms (definite) clause or feature interchangeably. In this work, we restrict our discussion to function free definite clauses. Based on the complexity (and utility), we categorize definite clauses/features into Simple

Conjuncts (\mathcal{SC}), Absolute Features (\mathcal{AF}), Primary Features (\mathcal{PF}), Composite Features (\mathcal{CF}) and Definite Features (\mathcal{DF}). While simple conjuncts are features derived from basic inputs at a single sequence position, other categories are capable of representing features derived from basic features at different relative positions. A detailed description of these categories and their relationships with each other is presented in section 3.2. We now discuss our proposed feature learning approaches for categories of features that we identify as relevant and useful.

To begin with, we learn features from the space of Simple Conjuncts, which is the feature space explored by McCallum (2003). Before delving into our optimal learning approach, we first briefly discuss a greedy feature induction approach for sequence labeling, that we proposed and developed.

An efficient feature induction approach that can automatically capture the mapping between labels and derived features (conjunctions) is desired. Inductive Logic Programming (ILP), a branch of machine learning, is a learning paradigm capable of learning such mappings or rules. Given some background knowledge and a set of facts as examples, ILP systems derive a hypothesis (structure) that entails all the positive examples and none of the negative examples. It starts with an initial hypothesis and refines the hypothesis by searching in a lattice of clauses based on a heuristic scoring function. Since in real world problems, the support for any input of a label and the support for inter state transitions are much fewer than that for same state transitions, in learning both the emission and transition dependencies using traditional systems, rules defining transitions within the same state tend to dominate. Such a model tends to predict fewer inter state transitions and thus affects the accuracy of inference, as observed in our experiments. Hence we focus only on the induction of emission rules and combine them with the set of n^2 interstate transitions while learning the parameters of the model.

In order to learn emission rules/features (for each label), we first employ a Branch and Bound (B&B) search on the lattice of emission features using an ILP system, Aleph. We then combine the learned emission features with transition features in a custom implementation of HMM and learn the parameters. However, typical ILP systems for structure learning do a Branch and Bound search in the lattice of clauses evaluating scores based on positive and negative examples covered, and therefore suffer from accuracy loss when used to construct features for HMM. We therefore propose and implement a greedy feature

induction approach that adapts the structure of HMM using HMM evaluation on a held out part of training data as scoring function⁸ to learn emission features for each label. Our experimental results suggest a performance improvement over both the traditional HMM and the B&B learning assisted HMM in terms of accuracy.

Although the greedy feature induction approach yields better prediction models than traditional approaches, an optimal model is not guaranteed. An exhaustive search for the optimal features is not feasible in real world problems, because the feature space is exponential in the number of basic features. Our objective is to efficiently explore the space of \mathcal{SC} s to discover discriminative features for sequence labeling. We therefore propose and develop a Hierarchical Kernels based approach for learning optimal \mathcal{SC} s for each label. Hierarchical Kernel Learning (HKL) was originally introduced by Bach (2009) for high-dimensional non-linear variable selection, by exploiting the hierarchical structure of the problem. Their approach extends the multiple kernel learning framework to the space of kernels that has a directed acyclic graph structure and performs kernel selection through a sparsity inducing norm. Jawanpuria et al. (2011) leveraged the HKL framework to learn rule ensembles for binary classification tasks. Our approach, referred to as Hierarchical Kernel Learning for Structured Output Spaces (StructHKL)⁹, optimally and efficiently learns discriminative features for multi-class structured output classification problems such as sequence labeling. We build on the Support Vector Machines for Structured Output Spaces (StructSVM) model (Tsochantaridis et al. 2004, Tsochantaridis 2006) for sequence prediction problems, wherein, we consider all possible \mathcal{SC} s to form the input features while the transition features are constructed from all possible transitions between the state labels. A ρ -norm hierarchical regularizer is employed to select a sparse set of \mathcal{SC} s. Since we need to preserve all possible transitions, a conventional 2-norm regularizer is employed for state transition features. The exponentially large input feature space is searched using an active set algorithm and the exponentially large set of constraints is handled using a cutting plane algorithm. In general, StructHKL can be used in structured output classification problems to learn from complex feature spaces that can be ordered as directed acyclic graph and where the summation of descendant kernels can be computed in polynomial time.

⁸This work has appeared in DaWaK, 2011 (Nair, Ramakrishnan & Krishnaswamy 2011).

⁹This work has appeared in AAAI, 2012 (Nair, Saha, Ramakrishnan & Krishnaswamy 2012).

As stated before, in this work, we further study the possibility of efficiently learning and using discriminative relational features (that capture sequential information among input variables) for sequence labeling. The StructHKL algorithm optimally solves the objective of learning the most discriminative \mathcal{SC} s for sequence labeling. However, due to the theoretical requirements for the feature space, as discussed briefly in the previous paragraph, its applicability in learning complex relational features, that are derived from inputs at different relative positions, is non-trivial and challenging. Therefore, from our feature categories, we determine simple feature classes that can be composed to yield complex ones, with the goal of formulating efficient yet effective relational feature learning procedures. We identify a class of simple features called Absolute Features (\mathcal{AF}). \mathcal{AF} s are self-contained, in the sense that an \mathcal{AF} stands by itself to convey an information. In other words, every variable used in an \mathcal{AF} is tied to a property or is in a relationship with other variables. Further, we identify a powerful class of features termed as Composite Features (\mathcal{CF}) that are constructed using conjunctions of \mathcal{AF} s. Please read 3.2 for definitions of \mathcal{CF} s and \mathcal{AF} s. Since \mathcal{CF} s are conjunctions of \mathcal{AF} s, it is trivial to observe that StructHKL can be employed to efficiently construct \mathcal{CF} s from \mathcal{AF} s. Therefore, optimal relational discriminative features can be learned either by (i) enumerating \mathcal{AF} s and discovering their useful compositions (\mathcal{CF}) using StructHKL or by (ii) developing methods to learn optimal \mathcal{AF} s (or \mathcal{CF} s directly).

The space of \mathcal{AF} s is prohibitively large and therefore it is not feasible to enumerate all \mathcal{AF} s in a domain. We therefore, propose to selectively enumerate \mathcal{AF} s based on some relevance criteria such as the support of the \mathcal{AF} in the training set. We define an \mathcal{AF} as *strongly relevant* if it helps the classification model to discern classes optimally. On the other hand, we consider a feature to be *weakly relevant* if it covers at-least a threshold percentage of examples. Since discovering *strongly relevant* \mathcal{AF} s is a hard task, we discover *weakly relevant* \mathcal{AF} s using Inductive Logic Programming tools. We employ pattern mining approaches to discover a relevant set of \mathcal{AF} s. Specifically, we use a relational pattern miner called Warmr (Dehaspe & Toivonen 1999, Dehaspe & Toironen 2000). Warmr uses a modified version of Apriori algorithm (Agrawal & Srikant 1994) to find frequent patterns (\mathcal{AF} s) which have minimum support, as specified by the user. Once a set of relevant \mathcal{AF} s are enumerated, StructHKL can be employed to learn useful compositions of \mathcal{AF} s and their parameters to get the final model. This can be viewed as

projecting the space of complex relational features such as \mathcal{CF} s into the space of \mathcal{SC} s and leveraging StructHKL¹⁰. We now briefly discuss the second approach of learning optimal \mathcal{AF} s (or \mathcal{CF} s directly).

An \mathcal{AF} is formed by combining one or more predicates which share variables. Thus, \mathcal{AF} s are constructed from its primary clauses by unifying different variables present in those clauses. Therefore, the partial ordering of \mathcal{AF} s does not comply with the requirement of StructHKL that the descendant kernels should be summable in polynomial time. This limits the possibility of leveraging StructHKL to optimally learn features in the space of \mathcal{AF} s (and its super-space of \mathcal{CF} s). For this reason, in the structured output classification model, we leverage a relational kernel that computes the similarity between instances in an implicit feature space of \mathcal{CF} s¹¹. To this end, we employ the relational subsequence kernel (Bunescu & Mooney 2006) at each sequence position/pivot (over a time window of observations around it) for the classification model. Relational subsequence kernels have been used to extract relations between entities in natural language texts (Bunescu & Mooney 2006). The features are (possibly non-contiguous) sequences of words and word classes. In our problem, at each temporal step (pivot position), we would like to learn composite features which capture relational information about basic inputs at positions relative to the pivot position. This sequence information would provide a rich feature space for the algorithm to learn a more expressive model. However, explicitly enumerating such a feature space is not feasible due to the high dimensionality of the feature space. Relational subsequence kernels implicitly capture the effectiveness of this rich feature space. We also show that the feature space of \mathcal{CF} s are captured by our relational subsequence kernels. While this way of modeling does not result in interpretability, relational subsequence kernels do efficiently capture the relational sequential information on the inputs. In this thesis, we also discuss our contribution in two related problem domains *viz.* inference in probabilistic first order logical systems and dimensionality reduction in classification settings, which we briefly introduce in the following subsections.

¹⁰This work has appeared in ILP, 2012 (Nair, Nagesh & Ramakrishnan 2012).

¹¹This work has appeared in ILP, 2013 (Nagesh et al. 2013).

1.4.2 Pruning Search Space for Weighted First Order Horn Clause Satisfiability

In first order settings, it is possible to have complex models that have complex first order structure, huge set of groundings, and the like. In sequence labeling, if it is feasible to ground all the variables, a dynamic programming algorithm called the Viterbi algorithm (Forney 1973) is the best choice. In all other cases (sequence labeling or general), we suggest using our contribution introduced in the following paragraph¹².

Many Statistical Relational Learning models pose inference as weighted satisfiability solving. Performing logical inference after completely grounding clauses with all possible constants is computationally expensive. If a set of horn clauses are fully satisfiable, then a minimal model can be found by selective grounding and using the T_Σ operator (referred to as the immediate consequence operator T_P in (Hogger 1990)) in polynomial time. However, weighted unsatisfiable problems need to find the most likely state based on the weights. We propose and develop an extension to the minimal model approach wherein we find (i) the relevant set of ground horn clauses which has a potential to be part of a contradiction and (ii) an interpretation close to the result. The MaxSAT algorithm (Selman et al. 1993) can be used on this subset of clauses, (optionally) starting from the interpretation returned, to get the most likely state. We also prove that local search for optimality in the pruned space cannot affect the satisfiability of the rest of the clauses. We prove theoretically and empirically that the optimal solution is guaranteed to exist in the pruned space. The approach finds a model, if it exists, in polynomial time; otherwise it finds an interpretation that is most likely given the weights¹³. We now move to the problem of dimensionality reduction.

1.4.3 Optimally Extracting Discriminative Disjunctive Features for Dimensionality Reduction

Support Vector Machines (SVM) and its variants are amongst the current state-of-the-art approaches to classification. These non-parametric maximum margin supervised learning frameworks have provided algorithms that yield optimal solutions to classification prob-

¹²Since we do not have access to such a sequence labeling data to demonstrate the validity of our approach, we present our work in a general setting

¹³This work has appeared in ILP, 2010 (Nair, Govindan, Jayaraman, TVS & Ramakrishnan 2011)

lems with binary, multi-label and structured output spaces. However, since many real world application domains are characterized by a large set of features that possibly contain a non-trivial amount of redundant and irrelevant information, using the entire feature space as it is often leads to over-fitting and therefore less effective classifier models. To alleviate this problem, a significant amount of research (Blei, Ng & Jordan 2003, Teh et al. 2004, Blei, Griffiths, Jordan & Tenenbaum 2003, Jolliffe 1986*b*, Ye & Ji n.d.) has been invested to reduce the dimensionality of the data either by projecting the features onto a collapsed space or selecting a subset of features, both as preprocessing steps. These approaches suffer from the drawback that the dimensionality reduction objective and the objective for classifier training are decoupled (the two tasks are performed one after the other) and often, the approach for dimensionality reduction is greedy. Recently, there have been some efforts to address the two tasks in a combined manner by attempting to solve an upper-bound to a single objective function (Zhu et al. 2010, Xu 2010). However the main drawback of these methods is that they are all parametric, in the sense that the number of reduced dimensions is taken as an input to the system. In this work, we propose an integrated learning approach for non-parametric dimension reduction by projecting the features from the original feature space to the space of disjunctions and discovering a sparse set of important disjunctions out of them. For datasets with nominal features, it is quite natural to consider disjunctions (or sets of synonymous features) as dimensions. Here, in order to discover good disjunctive features, hierarchical kernels, that efficiently and optimally perform feature selection and classifier training simultaneously in a maximum margin framework, have been employed¹⁴.

We evaluate the performance of our feature induction approaches on publicly available activity recognition datasets. Our experiments show improvements over other standard and state-of-the-art sequence labeling techniques. We also demonstrate the effectiveness of our satisfiability approach and the dimensionality reduction approach. Our experiments show that our satisfiability based inference approach reduces search space substantially and helps maxSAT to converge in short time. We also present results of our dimensionality reduction approach on standard datasets.

To summarize, we pose the problem of learning complex relational features in order to improve the efficiency of sequence labeling systems. In order to get more insight into

¹⁴This work has been accepted for COMAD, 2013

the feature space, we categorize features based on their complexity (and utility) and prove that complex features can be constructed from simpler ones. Among those features, we identify \mathcal{SC} s and \mathcal{CF} s as interesting for sequence labeling. While \mathcal{SC} s are derived from basic inputs at a single sequence position, \mathcal{CF} s are derived from basic inputs at multiple sequence positions. We first focus on learning \mathcal{SC} s. To discover discriminative features for sequence labeling, we first develop a greedy feature induction approach. Since greedy approaches cannot guarantee optimal solutions, we propose a Hierarchical Kernel Learning approach for structured output spaces (StructHKL), which efficiently learns optimal \mathcal{SC} s for sequence labeling. Since the construction of \mathcal{CF} s does not comply with the requirement of StructHKL that the descendant kernels of any node should be summable in polynomial time, StructHKL cannot be trivially leveraged to learn \mathcal{CF} s. We therefore propose and develop two strategies: (i) to enumerate all \mathcal{AF} s and learn \mathcal{CF} s by combining \mathcal{AF} s using StructHKL. (ii) to incorporate relational subsequence kernels in the structured output classification framework so that the relational sequential information on the inputs is captured implicitly. We also present two related contributions, that is (i) an approach for faster inference in relational settings and (ii) an integrated non-parametric dimensionality reduction approach. The next section outlines the structure of the thesis.

1.5 Thesis structure

This thesis is organized thus:

Chapter 2 discusses some of the related works on sequence labeling, feature induction and the application area, activity recognition.

Our main contributions are discussed in the chapter 3. Here we discuss our categorization of features and identify important categories suitable for sequence labeling tasks. We identify \mathcal{SC} s and \mathcal{CF} s as two categories particularly interesting for us. We start with our approaches for learning \mathcal{SC} s, for which we first propose a greedy feature induction approach. To learn optimal \mathcal{SC} s, a Hierarchical Kernels based approach for structured output classification (StructHKL) is proposed. We then present two strategies to efficiently learn \mathcal{CF} s for sequence labeling.

We present our two related contributions, the satisfiability approach and the dimensionality reduction approach in chapters 4 and 5 respectively.

In Chapter 6, we discuss our experimental setup, datasets and results. We evalu-

ate our proposed feature induction approaches in publicly available activity recognition datasets. The results for satisfiability based inference and the dimensionality reduction approach are also discussed.

We conclude this dissertation in Chapter 7 and discuss directions of future work.

Chapter 2

Related Work

Here, we look into some of the related works in the area of sequence labeling and feature induction. We also briefly talk about inference in sequence labeling. A brief account of popular approaches in the domain of activity recognition (which is one of the motivating problems for this research work) is also discussed. We start with a brief discussion on popular sequence labeling approaches.

2.1 Models for Sequence Labeling

The objective in learning sequence labeling models is to learn functions of the form $\mathcal{F} : \mathcal{X} \rightarrow \mathcal{Y}$ from the training data, where \mathcal{X} and \mathcal{Y} are input and output sequence spaces, respectively. Typically, a discriminant function $F : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$ is learned from training data that consists of pairs of input and output sequences. The prediction is performed using the decision function $\mathcal{F}(X; \mathbf{f})$:

$$\mathcal{F}(X; \mathbf{f}) = \arg \max_{Y \in \mathcal{Y}} F(X, Y; \mathbf{f}), \quad (2.1)$$

where $F(X, Y; \mathbf{f}) = \langle \mathbf{f}, \boldsymbol{\psi}(X, Y) \rangle$ represents a score which is a scalar value based on the features $\boldsymbol{\psi}$ involving input sequence X and output sequence Y values and parameterized by a parameter vector \mathbf{f} . In sequence prediction, features are constructed to represent emission (observation) and transition distributions. Given this objective, we can classify sequence labeling techniques into probability based and max-margin based, which we discuss in the following sections.

2.1.1 Probability Based Sequence Labeling techniques

In probability based sequence labeling methods, the parameters are characterized by probabilities. Hidden Markov Models (HMM) (Rabiner 1990) and Conditional Random Fields (CRF) (Lafferty, McCallum & Pereira 2001) are traditionally used in sequence prediction problems and have a similar objective as discussed above, with probabilities and probabilistic weights as parameters, respectively. Their ability to capture the state transition dependencies along with the observation dependencies makes these approaches robust in noisy and sparse data. In an HMM setup, probability parameters that maximize the joint probability of input and output training sequences are learned during the training phase. In contrast, CRF learns parameters that maximize the conditional probability of the output sequence given the input sequence. The probability distributions for HMM and CRF are shown in equations (3.1) and (3.2) respectively. Although HMM and CRF are rich enough to represent complex relationships, they have limitations in automatically learning these relationships. Since the concepts in HMM and CRF are the basic building blocks of our research, we discuss them in greater detail in chapter 3. Prediction is usually performed by a dynamic programming algorithm called the Viterbi Algorithm (Forney 1973). We now discuss the Support Vector Machines on Structured Output Spaces (StructSVM) (Tsochantaridis et al. 2004), which is a max margin method that can be used for sequence labeling.

2.1.2 Max-Margin Methods for Sequence Labeling

Tsochantaridis et al. (2004) generalize the SVM framework to perform classification on structured outputs. This builds on the conventional SVM formulation that assumes output as a single variable which can be a binary label or multi-class. The conventional SVM does not consider the dependencies between output variables and is not suitable for structured data such as sequential data, labeled trees, lattices, or graphs. StructSVM generalizes multi-class Support Vector Machine learning to incorporate features constructed from input and output variables and solves classification problems with structured output data. We now briefly explain the StructSVM approach in the specific case of sequence prediction.

Loss functions in structured outputs have to measure the amount by which the prediction deviates from the actual value and hence the zero-one classification loss is not

sufficient. In sequence prediction, the predicted sequence of labels that are different from the actual labels in a few time steps should be penalized less than those that differ from the actual labels in majority of the time steps. While any decomposable loss-function that holds the above property fits in this approach, the micro-average of wrong predictions is used in our research work. A loss function is represented as $\Delta : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$. $\Delta(Y, \hat{Y})$ is the loss value when the true output is Y and the prediction is \hat{Y} .

The SVM formulation for structured output spaces can thus be written as

$$\begin{aligned} \min_{\mathbf{f}, \xi} \quad & \frac{1}{2} \|\mathbf{f}\|^2 + \frac{C}{m} \sum_{i=1}^m \xi_i, \quad s.t. \quad \forall i : \xi_i \geq 0 \\ \forall i, \forall Y \in \mathcal{Y} \setminus Y_i : \quad & \langle \mathbf{f}, \boldsymbol{\psi}_i^\delta(Y) \rangle \geq 1 - \frac{\xi_i}{\Delta(Y_i, Y)}. \end{aligned} \quad (2.2)$$

where m is the number of examples, C is the regularization parameter, ξ 's are the slack variables introduced to allow errors in the training set in a soft margin SVM formulation, and $X_i \in \mathcal{X}$ and $Y_i \in \mathcal{Y}$ represent the i^{th} input and output sequences respectively¹. $\langle \mathbf{f}, \boldsymbol{\psi}_i^\delta(Y) \rangle$ represents the value $\langle \mathbf{f}, \boldsymbol{\psi}(X_i, Y_i) \rangle - \langle \mathbf{f}, \boldsymbol{\psi}(X_i, Y) \rangle$. Equation 2.2 represents the slack scaling approach proposed by Tsochantaridis et al. (2004). We follow slack scaling in our paper, as it is believed to be more accurate and better behaved than margin scaling (Sarawagi & Gupta 2008).

In cases where the sequence length is large, the number of constraints in (2.2) can be extremely large. To solve this problem, an algorithm based on the cutting plane method is proposed by Tsochantaridis et al. (2004) (*c.f.* algorithm 1 in (Tsochantaridis et al. 2004)) to find a polynomially sized subset of constraints that ensures a solution very close to the optimum.

In a related work, Taskar, Guestrin & Koller (2004) combine the advantages of kernel based methods (capable of handling high dimensional feature spaces and with strong theoretical guarantees) and probabilistic graphical models (capable of explicitly representing correlations between labels in structured output spaces) in a Maximum Margin framework for Markov Networks (M^3N). For Markov Networks that can be triangulated, the resultant quadratic objective is reduced to a polynomial size formulation, whereas, for non triangulated Markov Networks, an approximate reformulation (based on a relaxation technique used in belief propagation algorithms) is proposed. In another work, Taskar,

¹Subscript i here is to denote i^{th} example sequence and should not be confused with the i^{th} element of a vector

Chatalbashev & Koller (2004) focus on a subclass of Markov Networks called Associative Markov Networks (AMN). AMNs have clique potentials that favor the same labels for all variables in the clique. The paper proposes an approximation to the M^3N objective to solve a linear program relaxation of the AMN objective. All these works propose maximum margin methods for structured output classification. Whereas, our work focuses on learning the relational structure in input space for improving structured output prediction. The next section discusses some of the existing approaches that looked into learning relationships to improve accuracy of prediction.

2.2 Learning Relationships as Features

In our research work, we focus on learning relationships in the form of features for sequence labeling models. We could categorize feature induction methods (generally) into (i) Greedy and (ii) Optimal. We discuss the existing feature induction approaches in the subsections that follow.

2.2.1 Greedy Feature Induction Approaches

McCallum (2003) proposes feature induction methods that iteratively construct feature conjunctions that increase an objective. This approach starts with no features and at each step, considers a set of candidate features (conjunctions or atomic). Features whose inclusion will lead to a maximum increase in the objective are selected. Weights for the new features are trained. The steps are iterated until convergence. The approach learns conjunctive features from ground basic inputs². McCallum (2003) trains a CRF model and uses conditional log-likelihood as the objective for the greedy induction. This effectively solves the problem of the incorrect assumption, that individual observations are independent, while not dealing with exponential observation space.

²In Feature Induction for CRF, the approach starts with no feature, creates a list of candidate features (singleton or conjunctions of existing features) with the highest gain, evaluates all candidate features, adds those features with the highest gain, adjusts the parameters of the CRF model and repeats until convergence criterion is met. If ground basic inputs at multiple sequence positions are provided, McCallum’s as well as our greedy approach can learn their conjunctions. However, the approach does not learn relational features (if ground basic inputs are not provided/feasible to provide) from multiple relative sequence positions.

Although greedy feature induction approaches have been shown to improve performance, they cannot guarantee an optimal solution. An exhaustive search to find the optimal solution is expensive due to the exponential size of the search space. A brief account of previous approaches that tried to improve sequence labeling by learning structure in input/output (not necessarily at the same sequence step) is given below.

A Logical Hidden Markov Model is discussed in (Kersting et al. 2006), which deals with sequences over logical atoms. A model selection approach for the Logical Hidden Markov Model is proposed in (Kersting 2005), which is based on the Expectation Maximization algorithm and Inductive Logic Programming principles. Our approach differs from their approach in the sense that our objective is to explore the relationships among multiple inputs at relative sequence steps to improve the efficiency of sequence labeling. Thon (2010) and Thon et al. (2011) elaborate on relational markov processes which are concerned with efficient parameter learning and inference. They assume that a structure has been provided upfront. Similarly, a relational Bayesian network learning is discussed in (Schulte et al. 2012) with the goal of learning the parameters given the structure of the bayes-net.

TildeCRF (Gutmann & Kersting 2006) has an objective similar to our approach, where the relational structure and parameters of a CRF for sequence labeling are learned. TildeCRF uses relational regression trees and gradient tree boosting to learn the structure and parameters³. However, TildeCRF gives no guarantee that the learned structure is optimal. To the best of our knowledge, there has not been any approach for learning optimal input structure for sequence labeling. We, therefore, discuss an approach for optimal induction of feature conjunctions in a binary classification setting.

2.2.2 Optimal Feature Induction for Binary Classification

Jawanpuria et al. (2011) propose Rule Ensemble Learning using Hierarchical Kernels where they make use of the Hierarchical Kernel Learning (HKL) framework introduced by Bach (2009) to learn, simultaneously, sparse rule ensembles and their optimal weights. We will refer to their approach as RELHKL. The regularizer used in HKL discourages

³We propose two learning approaches for the feature space explored in TildeCRF. In one, we derive convex formulations for a significant portion of the learning steps. Our other approach solves the problem optimally but the learned feature space is not interpretable.

selection of rules that involve a large number of basic features. Jawanpuria et al. (2011) prove that although HKL discourages selection of large rules, it redundantly selects all the rules that are subsets of the chosen rules. As a solution, they generalize HKL with a convex formulation using a $(1,2]$ norm (ρ -norm) regularizer that ensures a set of sparse and non redundant rules. A mirror descent based active set algorithm is employed to solve the convex formulation. We briefly discuss their approach in the paragraphs that follow.

The prime objective of Rule Ensemble Learning (REL) is to learn a small set of simple rules and their optimal weights. The set of rules that can be constructed from basic features follow a partial order and can be visualized as a lattice (conjunction lattice when the features are conjunctions of basic features). The set of indices of the nodes in the lattice are represented by \mathcal{V} . The model is additive in nature and the weighted sum of the features decides the output. To learn a sparse sets of rules, the regularizer $\Omega(\mathbf{f})$ is modified in the following way (Jawanpuria et al. 2011),

$$\Omega(\mathbf{f}) = \sum_{v \in \mathcal{V}} d_v \| \mathbf{f}_{D(v)} \|_{\rho} \quad (2.3)$$

where \mathbf{f} is the feature weight vector corresponding to the feature nodes in the lattice, $d_v \geq 0$ is a prior parameter showing the usefulness of the feature conjunctions, $\mathbf{f}_{D(v)}$ is the vector with elements as $\| f_w \|_2 \quad \forall w \in D(v)$, $D(v)$ the set of descendant nodes of v and $\| \cdot \|_{\rho}$ represents the ρ -norm. In rule ensemble learning d_v is defined as $\beta^{|v|}$, where β is a constant. The optimization problem, with hinge loss, can now be written as,

$$\begin{aligned} \min_{\mathbf{f}, \mathbf{b}, \xi} \quad & \frac{1}{2} \Omega(\mathbf{f})^2 + C \sum_{i=1}^m \xi_i, \\ \text{s.t.} \quad & y_i \left(\sum_{v \in \mathcal{V}} \langle f_v, \psi_v(\mathbf{x}_i) \rangle - \hat{b} \right) \geq 1 - \xi_i, \\ & \xi_i \geq 0 \end{aligned} \quad (2.4)$$

where y_i and \mathbf{x}_i are the output label and input vector respectively for i^{th} example. \hat{b} is the bias. Other notations are as defined in previous subsections.

Since the 1-norm induces sparsity (Rakotomamonjy et al. 2008, Bach 2009), for most of the $v \in \mathcal{V}$, $\| \mathbf{f}_{D(v)} \|_{\rho} = 0$ and this implies $f_w = 0, \forall w \in D(v)$. Since norms between $(1, 2)$ promote sparsity (Szafranski & Rakotomamonjy 2008, Jawanpuria et al. 2011), even if $\| \mathbf{f}_{D(v)} \|_{\rho}$ is not forced to zero, many of the $f_w = 0$ for $w \in D(v)$.

At optimality, only a few features are expected to be non-zero. The solution obtained with these non zero features will be the same as the solution obtained with the original set of features. Therefore for computational efficiency, an active set algorithm can be employed (*c.f.* algorithm 1 of (Jawanpuria et al. 2011)) which starts with an initial set of non zero features. At every step, it solves an optimization problem; a sufficiency condition is checked and it terminates if satisfied. Otherwise the nodes violating the sufficiency condition are added to the active set and the algorithm moves on to the next iteration.

The RELHKL approach is specific to the single variable binary classification problems and cannot be trivially applied to problems involving multi class structured output data. We now briefly discuss inference in sequence labeling.

2.3 Inference

Inference in sequence labeling is to assign a label for each sequence step based on the sequence of observations. This amounts to selecting the correct sequence from an exponential number of candidate sequences. The Viterbi algorithm (Forney 1973), a dynamic programming algorithm, is typically employed for efficiently solving the inference problem in sequence labeling. We give a brief overview of the Viterbi algorithm below.

2.3.1 The Viterbi Algorithm

The Viterbi algorithm (Forney 1973) is a dynamic programming algorithm used extensively in sequence labeling tasks to find the most probable sequence of hidden states. The Viterbi algorithm works in two phases, the forward phase and the backward phase. In each step of the forward phase, the algorithm determines the best path so far to reach each of the states at that sequence step, based on the observations and parameters. In the backward phase, the algorithm determines the best scoring label at the last sequence step and traces back the best path to this label. The running time of the Viterbi algorithm is in the order of $O(T \times n^2)$, where T is the sequence length and n is the number of state labels.

The Viterbi algorithm efficiently computes the label sequence for sequence labeling tasks, in settings where all variables can be grounded. However, in cases where it is not feasible to ground all the variables, we need to employ clever methods that avoid

complete grounding (thus saves memory and time). Also in instances where additional constraints exist in the output space, the Viterbi algorithm has limitations. There are a few approaches that solve problem settings with additional constraints on the output space. For example, Roth & Yih (2007, 2005) posed finding the sequence of output labels as an Integer Linear Programming problem, wherein, additional constraints on the output variables can be enforced. They evaluate their approach, which extends Conditional Random Field or Markov Random Field models to support general output constraint structures, on Natural Language Processing tasks such as semantic role labeling and named entity recognition. However, these approaches are not developed to solve inference problems in general first order settings. In chapter 4, we present an approach that prunes the search space for first order inference using satisfiability. The next section gives a brief account of some of the existing approaches in our motivating problem domain of activity recognition.

2.4 Recent developments in Activity Recognition

Automatic activity recognition has been an active research area in the current era of pervasive systems. Various approaches have been proposed. Wilson (2005) experimented with particle filter and context aware recognition for recognizing ADLs at the MIT Laboratory. Gibson et al. (2008) discussed the idea of clustering sensors for recognizing activities and concluded that trivially imposing clusters differs from reality. A relational transformation based tagging system using ILP concepts is proposed by Landwehr et al. (2009). The approach starts with an initial tag to all the sequences and then improves by learning a list of transformation rules which can re-tag based on context information. The approach is purely logical and not probabilistic. Wang et al. (2007) identify the minimal set of sensors that can jointly predict all activities in the domain. Binsztok et al. (2004) discussed learning HMM structure (number of states and allowed transitions) from the data for clustering sequences. Landwehr et al. (2006) construct kernel functions from features induced by an ILP approach. The search for features is directed by a Support Vector Machine performance using the current kernel. Mauro et al. (2010) aim to classifying relational sequences using relevant patterns discovered from labelled training sequences. Here, the whole sequence is labelled and not the individual components of the sequence. Patterns in each dimension of multi dimensional sequences are discovered and a feature

vector is constructed. Then an optimal subset of the features is selected using a stochastic local search guided by a naive Bayes classifier.

Many of these learning approaches are for general classification. However, in the case of sequential, skewed, and noisy⁴ activity recognition data where temporal dependencies dominate over static dependencies, most of the learning approaches that globally normalize⁵ the parameters do not fit well. We find a solution to this problem by identifying relevant conjunctions of sensors for each activity as input features. We now discuss our proposed approaches for feature induction in the following chapter.

⁴Noise due to faulty sensors, communication lines and non uniform patterns adopted by subjects to perform tasks.

⁵In sparse sequence labeling domains, since the support for same state transitions are much higher than emissions and inter state transitions, if we learn all the emission and transition features simultaneously, where the parameters are compared relative to each other to select the best, the same state transition features get more importance and hence dominate over other features.

Chapter 3

Learning Discriminative Relational Features for Sequence Labeling

3.1 Introduction

In our research work, we investigate the potential of learning input structure to improve sequence labeling models. To this end, we identify the categories of features relevant for discerning labels at each position in a sequence. Before going into the details of feature categories, we restate our motivation for the need to learn higher order features and objectives formally .

In a sequence labeling setting with multiple basic inputs at each sequence position, the joint state of these basic inputs at time t forms our observation/emission and we represent it as \mathbf{x}^t . The label at time t is represented by y^t . Prediction models such as the Hidden Markov Model (HMM) (Rabiner 1990), the Conditional Random Field (CRF) (Lafferty, McCallum & Pereira 2001) or StructSVM (Tsochantaridis 2006, Tsochantaridis et al. 2004) are trained from historical input/output data. These approaches typically assume that y^t is independent of all previous activities given y^{t-1} and \mathbf{x}^t is independent of all other variables given y^t . Figure 1.1 illustrates the independence assumption in an HMM setting.

In HMMs, the score is the joint probability distribution, $p(X, Y)$, of the input sequence X and the output sequence Y . From the independence assumptions as illustrated in Figure 1.1, one can factorize the joint probability distribution of the sequence of inputs (X) and labels (Y) into three factors: the initial state distribution $p(y^1)$, the transition distribution $p(y^t|y^{t-1})$, and the emission distribution $p(x^t|y^t)$ (Rabiner 1990). Here x^t and

y^t represent the input variable and the class variable at time t respectively. Therefore,

$$p(X, Y) = \prod_{t=1}^T p(y^t | y^{t-1}) p(x^t | y^t) \quad (3.1)$$

where T is the length of the sequence and $p(y^1 | y^0)$ is used instead of $p(y^1)$ to simplify the notation. Parameters for the distributions are learned by maximizing the joint probability, $p(X, Y)$, of the paired input and label sequences in the training data.

In CRFs (Lafferty, McCallum & Pereira 2001), parameters that maximize the conditional probability, $p(Y|X)$, of a sequence of states Y given a sequence of inputs X are learned. The conditional probability can be computed as,

$$p(Y|X) = \frac{1}{Z(X)} \exp \sum_{t=1}^T \phi_t(y^t, X) + \phi_{t-1}(y^{t-1}, y^t, X). \quad (3.2)$$

where $\phi_t(y^t, X)$ and $\phi_{t-1}(y^{t-1}, y^t, X)$ are potential functions and $Z(X)$ is the partition function.

These parameters are later used to identify the (hidden) label sequence that best explains a given sequence of inputs/observations. StructSVM (Tsochantaridis et al. 2004, Tsochantaridis 2006) is a maximum margin framework for structured output spaces such as sequence labeling. It generalizes the standard Support Vector Machines (SVM) with the margin defined as the difference in the scores of the original output sequence with any other possible output sequence.

In general, the objective of learning sequence labeling models is to learn functions of the form

$$\mathcal{F} : \mathcal{X} \rightarrow \mathcal{Y} \quad (3.3)$$

from the training data, where \mathcal{X} and \mathcal{Y} are input and output sequence spaces, respectively. Typically, a discriminant function

$$F : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R} \quad (3.4)$$

is learned from training data that consists of pairs of input and output sequences. Then inference can be performed using the decision function,

$$\mathcal{F}(X; \mathbf{f}) = \arg \max_{Y \in \mathcal{Y}} F(X, Y; \mathbf{f}), \quad (3.5)$$

where

$$F(X, Y; \mathbf{f}) = \langle \mathbf{f}, \boldsymbol{\psi}(X, Y) \rangle \quad (3.6)$$

represents a score which is a scalar value based on the features $\boldsymbol{\psi}$, involving input sequence X and output sequence Y values, and parameterized by a parameter vector \mathbf{f} . The features for sequence labeling are constructed to represent emission (observation) and transition distributions. That is, $\boldsymbol{\psi}$ consists of features describing emission relationships and transition relationships. The training objective is to learn weights that make the score $F : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$ of true output sequence Y greater than any other possible output sequence, given an input sequence X (Tsochantaridis et al. 2004, Tsochantaridis 2006).

We now discuss the limitations of typical sequence labeling approaches and then explain our objective of discovering the input structure in sequence labeling tasks. For simplicity of exposition, we discuss the limitations in the simple setting of HMM.

In an HMM set-up, the probability distribution of the observation/input given label, $p(x^t|y^t)$, is represented as an emission matrix. In domains with multiple possible observations, the input variable is a vector $\mathbf{x}^t = (x^{1^t}, x^{2^t}, \dots, x^{N^t})^\top$, where x^{i^t} represents the value of i^{th} input at time t and N is the number of basic inputs. An example is shown in Figure 1.2. This results in 2^N values for input \mathbf{x}^t which is computationally feasible only in small settings. Providing tuples consisting of individual inputs to the model is a solution to this. However, these approaches (HMM, CRF, StructSVM) have limitations in learning such relations. Therefore, often independence is assumed among individual inputs, given a label, to simplify the representation and computation of $p(\mathbf{x}^t|y^t)$. Conditional probability, when independence is assumed among inputs, is

$$p(\mathbf{x}^t|y^t) = \prod_{i=1}^N p(x^{i^t}|y^t) \quad (3.7)$$

In complex problem settings, where there are non-linear interactions among input variables, assuming conditional independence would affect the performance in terms of the accuracy of labeling. To alleviate both the issues, we identify the need to find a mapping between labels and the relevant compositions of inputs. The label specific compositions of inputs are referred to as features. Features can be derived from inputs at a single sequence position or across relative sequence positions. Our work underlines the notion that if a few dependent inputs (at same position or relative positions) in conjunction with

the information regarding the previous label in a sequence step can jointly decide the label at the current step, then it is better to consider only those. For instance, in the simple setting of features derived from inputs at single sequence position, we avoid the non relevant x^i 's and use compositions (or conjunctions) of relevant x^j 's to improve the prediction accuracy. This also helps to reduce the effect of noise while doing inference.

We propose learning the emission structure that maximizes probabilistic coverage (probability by which examples are covered) of the training data. In our problem, since we assume that all transitions are important, the model learned should allow all inter state transitions. Moreover, the transition feature space is not exponential. Therefore, we learn the structure of emission distribution while preserving all the n^2 transition probabilities. In the next paragraph and the section to follow, we discuss features in detail.

As stated above, we are interested in label specific features derived from multiple inputs/observations at any single sequence position or at different relative positions. Definite clauses/rules (clauses having all negated atoms except one (head)) can be used to represent such feature mappings. Since a class specific feature can be constructed from the body literals of a definite clause whose head depicts the class label, we use the terms definite clause and (label specific) feature interchangeably. While a label specific input feature can be visualized as a definite clause, the transition relationships can also be visualized as definite clauses. For example, the clause $0.5: activity(t, eatBreakFast) \leftarrow activity(t-1, prepareBreakFast)$ represents a simple transition relationship, which says that the activity at time t is *eatBreakFast* if the previous activity was *prepareBreakFast* and 0.5 is the degree of belief attributed to the rule. We now focus on input features, which are more complex than transition features. For simplicity, we drop the probabilistic weights while defining our feature categories. In this chapter, we first categorize definite features based on their complexity and identify the categories relevant for improving sequence labeling models. We then investigate leveraging Inductive Logic Programming (ILP) approaches to supplement or complement sequence labeling models. We then identify the limitations of traditional ILP systems to learn features for sequence labeling and propose a greedy feature induction approach that alleviates the limitations of traditional systems. Since the greedy approach cannot guarantee an optimum model, we propose to leverage optimum feature learning methods in feature learning steps for sequence labeling. To this end, we propose and develop a Hierarchical Kernels based approach to learn optimal features

derived from basic inputs at a single sequence position, which is referred to as StructHKL. We then develop strategies to leverage StructHKL to learn complex relational features derived from inputs at multiple relative sequence positions. We also propose and develop approaches to learn complex relational features efficiently by leveraging relational sub-sequence kernels. The categorization of first order definite features is discussed in the following section.

3.2 First Order Definite Features

Most of the Inductive Logic Programming (ILP) systems and the Statistical Relational Learning systems (SRL) learn clauses by searching in a space (often a lattice) of clauses in the domain. The search space, due to computational reasons, is typically controlled by language restrictions, which define the type of clauses to be learned. One common way to model a classification problem is to learn definite clauses (clauses having one head predicate conditioned on the values of zero or more body predicates). Since we are interested in such a setup, we confine our discussion to the space of definite clauses. We use the terms first order definite clause and first order feature interchangeably, as one can be derived from the other. We start by defining categories of predicates and then discuss the complexity based classification of features.

Similar to the *structural* and *property* predicates in 1BC clauses (Flach & Lachiche 1999), we define two types of predicates, *viz.* (*inter*) *relational* and *evidence* predicates. A *relational* predicate is a binary predicate that represents the relationship between *types* or between a *type* and its parts, where a *type* is an entity or object that has a meaning and described by itself or its attributes¹. For example, the relational predicate `prevRelPosWindowNear(t1,t2)` states the relationship between two sequence positions `t1` and `t2` that `t2` is in a previous position window of `t1` and both are near to each other. An *evidence* predicate is an assertion of a situation or a property of a *type* or part of it². For example, `microwave(t)` states that the microwave was on at time `t`³. We use the

¹Example of a type is a sequence position, which has a meaning by itself. Another example is an object such as cookware, where it has component parts or properties (such as “can contain”)

²Evidence predicate here is a predicate that states something about an entity and should not be confused with that used in Markov Logic Networks. A property is some conclusive information about a type.

³Typically, evidence predicates are unary and relational predicates are n-ary, $n > 1$

following set of definite clause examples for illustration of the concepts we discuss in the rest of this chapter.

1. `prepareDinner(t) :- microwave(t)`
2. `prepareDinner(t) :- microwave(t), platesCupboard(t)`
3. `prepareDinner(t1) :- prevRelPosWindowNear(t1,t2), platesCupboard(t2)`
4. `prepareDinner(t1) :- prevRelPosWindowNear(t1,t2)`
5. `prepareLunch(t1) :- prevRelPosWindowNear(t1,t2), platesCupboard(t2),`
`microwave(t2)`
6. `prepareLunch(t1) :- prevRelPosWindowNear(t1,t2), platesCupboard(t2),`
`prevRelPosWindowNear(t1,t3), microwave(t3),`
`greater(t2,t3)`
7. `prepareDinner(t1) :- microwave(t1), prevRelPosWindowNear(t1,t2),`
`platesCupboard(t2)`

We now categorize definite features based on complexity into Simple Conjuncts (\mathcal{SC}), Absolute Features (\mathcal{AF}), Primary Features (\mathcal{PF}), Composite Features (\mathcal{CF}) and Definite Features (\mathcal{DF}). The definitions of \mathcal{SC} , \mathcal{AF} and \mathcal{CF} are used in this thesis, while the other categories are presented for supporting these definitions. The reader may skip the definitions for \mathcal{PF} s and \mathcal{DF} s.

Simple Conjuncts (\mathcal{SC}):

\mathcal{SC} s are simple conjunctions of basic features (including unary conjunctions) observed at a single sequence position. In other words, \mathcal{SC} s are conjunctions of evidence predicates (without any relational predicate). Clauses 1 and 2 above are Simple Conjuncts. None of the other clauses are \mathcal{SC} s.

Absolute Features (\mathcal{AF}):

In absolute features (clauses), new local variables can only be introduced in a *relational* predicate, where a local variable is a variable not present in the head predicate.

Unlike in 1BC clauses, any number of new local variables can be introduced in a *relational* predicate. Any number of *relational* and *evidence* predicates can be conjoined to form an \mathcal{AF} such that the resultant \mathcal{AF} is minimal and the local variables introduced in *relational* predicates are consumed⁴ by some other *relational* or *evidence* predicates. Here a minimal clause is one which cannot be constructed from smaller clauses that share no common variables other than that in the head. So clauses 1, 3, 5 and 6 above are \mathcal{AF} s whereas clauses 2 (not minimal), 7 (not minimal) and 4 (since variable $\mathbf{t2}$ is not consumed) are not.

Primary Features (\mathcal{PF}):

Primary features (clauses) are absolute features (clauses) that have at-most one *evidence* predicate for every new local variable introduced. This is similar to elementary features in (Flach & Lachiche 1999) except that elementary features allow only one new local variable in a *structural* predicate. Clause 1 and 3 are \mathcal{PF} s whereas the other clauses do not conform to the restrictions imposed.

Composite Features (\mathcal{CF}):

Composite Features (clauses) are definite clauses that are formed by the conjunction of one or more \mathcal{AF} s without unification of body literals. Only the head predicates are unified. As in \mathcal{AF} s, every local variable introduced in a relational predicate should be consumed by other relational or evidence predicates. Clauses 1 (also qualifies for \mathcal{SC} , \mathcal{PF} and \mathcal{AF}), 2 (also qualifies for \mathcal{SC}), 3 (also qualifies for \mathcal{PF} and \mathcal{AF}), 5 (also a \mathcal{AF}), 6 (also a \mathcal{AF}) and 7 are \mathcal{CF} s where as 4 is not.

First Order Definite Features (\mathcal{DF}):

First order definite features (clauses) are features with none of the above restrictions. Therefore, all the given examples are \mathcal{DF} s.

In all the feature categories discussed above, the body of a rule can contain query predicates (labels/response variables). However, in our problem, we consider the cases where the body contains only evidence predicates. We now state some of the relationships between these categories of features. Some of the proofs come directly from the definitions;

⁴Consumption of a variable means that it is used by another predicate.

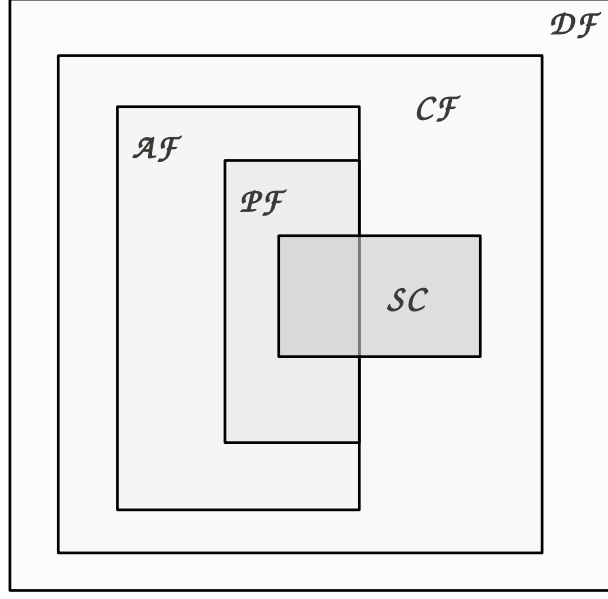


Figure 3.1: Subset relationships among various categories of features.

but are included for completion. The subset relationships among these categories of features are also illustrated in Figure 3.1

Claim 1. The set of primary features is a proper subset of the set of Absolute Features. That is, $\mathcal{PF} \subset \mathcal{AF}$.

Proof. From definition, \mathcal{PF} s are \mathcal{AF} s with the restriction that a new local variable introduced should be transitively consumed by a single *evidence* predicate. Hence $\mathcal{PF} \subseteq \mathcal{AF}$. Now, consider the clause $A(a) :- B(a,b), C(b), D(b)$. Since it follows all the requirements of an \mathcal{AF} , it is an absolute feature. However, since there are two *evidence* predicates for the local variable b , it does not qualify to be a \mathcal{PF} . Hence, $\mathcal{PF} \neq \mathcal{AF}$. \square

Claim 2. The set of absolute features is a proper subset of the set of composite features. That is, $\mathcal{AF} \subset \mathcal{CF}$.

Proof. From definition, \mathcal{CF} s are conjunctions of one or more \mathcal{AF} s. Therefore, all \mathcal{AF} s are \mathcal{CF} s (unary conjunctions). Now, consider the \mathcal{CF} clause $A(a) :- B(a,b), C(b), B(a,c), D(c)$. Since this is a conjunction of two \mathcal{AF} s ($A(a) :- B(a,b), C(b)$ and $A(a) :- B(a,c), D(c)$), this is not minimal, hence, not an \mathcal{AF} . Hence, $\mathcal{AF} \neq \mathcal{CF}$. \square

Claim 3. The set of simple conjuncts is a proper subset of the set of composite features. That is, $SC \subset CF$.

Proof. From definition, CF s are conjunctions of one or more AF s. Also SC s are conjunctions of evidence predicates at a single sequence position. Since an SC with a single evidence predicate is also an AF , conjunctions of such single predicate SC s are the same as conjunctions of single predicate AF s; thus all SC s are CF s. Now, consider the clauses 3, 5 and 7 which are CF s, but not SC s. Therefore, $SC \neq CF$. \square

Claim 4. The set of composite features is a proper subset of the set of full first order definite features. $CF \subset DF$.

Proof. From definition, DF s are first order definite clauses without any restrictions imposed for CF s; therefore, $CF \subseteq DF$. Now consider the clause $A(a) :- B(a, b)$, which is a first order relation that does not qualify as a CF , as the variable b introduced is not consumed. Therefore, $CF \neq DF$. \square

Claim 5. Every AF can be constructed from PF s using unifications.

Proof. The difference a AF has with PF is that it can have more than one *evidence* predicate for each local variable introduced. Let l_p be a *relational* literal in the body of an AF clause which introduces only one local variable. Let l_1, l_2, \dots, l_{p-1} be the set of *relational* literals in the body, which l_p depends on. Let there be $P \geq 0$ number of dependency chains starting from l_p to some *evidence* predicates, each of which is represented as l_{p+1}^i, \dots, l_k^i . We define l_p as a pivot literal if $P > 1$. For simplicity, we assume there is only one pivot in a clause. Now, we can construct P PF clauses from this with the body of the i^{th} clause as $l_1, l_2, \dots, l_{p-1}, l_p, l_{p+1}^i, \dots, l_k^i$, where l_k^i is a *evidence* predicate. It is trivial to see that these P clauses can be unified to construct the original AF . For multiple pivot literal clauses, the above method can be applied recursively until PF clauses are generated. The proof can be extended to pivot literals with multiple new local variables by using a dependency tree structure in place of chain. \square

Claim 6. Every CF can be constructed from AF s by conjunctions.

Proof. By definition, a clause qualifies as a CF only if it is constructed from the conjunction of one or more AF s. \square

Claim 7. \mathcal{CF} s are first order \mathcal{DF} s with local variable reuse restriction.

Proof. \mathcal{AF} s include maximal⁵ clauses generated only with unification (without conjunctions) of \mathcal{PF} s. These clauses have the restriction that all local variables introduced should be consumed transitively. \mathcal{CF} s capture all possible conjunctions of \mathcal{AF} s and therefore can generate any definite clause that is consistent with the local variable consumption restriction. \square

We briefly discuss below some of the existing complexity based categorization of first order definite features in the following paragraph.

1BC clauses and elementary clauses introduced by Flach & Lachiche (1999) are similar to \mathcal{AF} s and \mathcal{PF} s respectively, with the restriction that a structural (relational) predicate can have only one new local variable. Simple clauses are defined by McCreath & Sharma (1998) as the clause with at-most one sink literal, where a sink literal is one which has no other literal dependent on it. Simple clauses need not have a sink for a local variable and thus differs from \mathcal{PF} s. From the feature categories discussed above, we identify \mathcal{SC} s and \mathcal{CF} s as two categories particularly interesting for us. While \mathcal{SC} s are simple conjunctive features derived from inputs at a single sequence position, \mathcal{CF} s can be constructed from inputs at multiple sequence positions relative to the current position. We first present our contributions that learn \mathcal{SC} s and then discuss our learning approaches for \mathcal{CF} s. In the following paragraph, we present an analysis of traditional feature induction approaches in either complementing or supplementing the sequence labeling model, the HMM. We also present a greedy feature induction approach, that differs from traditional Inductive Logic Programming approaches in the scoring function and learning setting.

3.3 Greedy Feature Induction for Sequence Labeling

As motivated in the introduction, our objective is to learn higher order features efficiently from spaces exponential in the number of basic features. To this end, we start with a brief discussion on employing Inductive Logic Programming systems (either complementing or supplementing conventional Hidden Markov Model (HMM)) to learn the models for sequence labeling. We then identify the limitations of such systems on sparse, skewed and noisy data (such as activity recognition) and propose our feature induction assisted HMM.

⁵Maximal with respect to unifications and minimal with respect to conjunctions.

We use a greedy search strategy to learn emission features in the form of simple conjuncts (SC). SC s discovered by a greedy search in the lattice of clauses/features are combined with transition relationships in a custom implementation of HMM and parameters are learned. In this thesis, we use the terms features and rules interchangeably, because a label specific feature can be constructed from the body part of a definite clause rule whose head depicts the label. In sequence labeling, we therefore have emission rules and transition rules. Emission rules are those whose head is the label and the body comprises inputs. Transition rules are those whose head and body are label/output predicates. In the HMM setting, we learn parameters for each emission feature (defined as above) and the transition features. The joint probability of input and output sequences can still be factorized into emission and transition parts.

As discussed above, we are interested in finding a mapping between labels and relevant conjunctions of individual inputs. The mapping can be expressed as relationships in the form of definite clause rules (“Label if a particular set of inputs are on”), which are traditionally represented in the form $A \leftarrow B, C, \dots$ where A, B, C, \dots are binary predicates. Traditional structure learning systems are capable of discovering rules of the above form.

Our objective is to learn an emission structure that maximizes probabilistic coverage (probability by which examples are covered) of the training data. Our experiments with Inductive Logic Programming (ILP) systems alone failed to learn useful rules and resulted in a bad classifier model. The reason for this is that since, in real world problems such as activity recognition, the support for any emission of an activity and the support for inter state transitions are much fewer than that for the same state transitions, in learning both the emission and transition dependencies using traditional systems, rules defining transitions within the same state tend to dominate. Such a model tends to predict fewer inter state transitions, and thus affects the accuracy of prediction/labeling. Hence we focus only on the induction of emission rules and combine them with the set of n^2 interstate transitions while learning the parameters of the model. We first study the applicability of ILP systems that use a Branch & Bound search for discovering emission rules. We identify the limitations of this approach and then propose our greedy feature learning approach.

3.3.1 Logical Coverage Based Feature Induction for HMM

Since the real world settings such as activity recognition have sparse, skewed and noisy data, learning the complete model using standard structure learning systems often does not yield efficient models in terms of accuracy of labeling⁶. We therefore, learn emission features using standard Branch & Bound structure learning systems, combine them with the transition relationships, and learn parameters for the final model. ILP is one of the traditional structure learning paradigms that learn first order relations among entities. For example, Aleph (Srinivasan 2007) is a popular ILP system that in each iteration, selects a positive example, builds the most specific clause based on the example, searches for a more general clause that has the best score, and removes examples made redundant by the current clause. We use Aleph as a benchmark system for our experiments.

B&B systems, when used for learning emission rules, evaluate each refinement of clauses using scoring functions based on positive and negative examples. Since real world data is vulnerable to noisy information, an exact model is hard to get. Since the examples covered by a refinement are removed in each step, rules that are learned in subsequent iterations have less confidence than those learned initially, which leads to a less efficient model. Since the objective of traditional systems is to logically cover all the positive examples with clauses which is different from the actual objective of building a probabilistic model (HMM), the approach suffers from accuracy loss. We have experimented with this approach using Aleph combined with a customized implementation of HMM. Each rule returned by aleph is a definite rule, which associates a subset of inputs to a label. A new attribute (feature) is constructed with each such subset. Therefore, the number of attributes equals the number of rules learned. The learned logical model and the training data are passed to a customized implementation of HMM for constructing the probabilistic model. Later, the probabilistic model is used for inference.

Our experiments reveal that, HMM with B&B structure learning for feature construction yields better results than HMM without structure learning only in a small scale. This is because, in traditional systems, the objective is to logically cover all the positive examples. We identify the limitations of this approach as (i) The scores used by traditional systems such as Aleph are largely based on the number of positive and negative

⁶As stated earlier, in learning complete model using standard structure learning systems, the same state transitions tend to dominate.

examples covered by the current model⁷. (ii) Discovery of each of the clauses leads to the removal of positive examples covered. (iii) Logical coverage in place of probabilistic coverage. We now discuss our proposed greedy feature induction assisted HMM model construction that addresses these limitations.

3.3.2 Probabilistic Feature Induction for HMM

After analyzing the limitations of off-the-shelf branch & bound structure learning to assist HMM model construction, we propose a greedy hill climbing feature induction approach wherein we evaluate, in each refinement step, the current model in an HMM setting, which is based on a probabilistic score⁸. That is, the score which has to be maximized is an HMM evaluation on part of the training data. We call this approach the Probabilistic Feature Induction assisted HMM model construction (FIHMM). The score can be either micro-average accuracy⁹ or macro-average accuracy (referred to as time slice accuracy and average class accuracy, respectively, by van Kasteren et al. (2008)) of the current model. Micro-average accuracy is the fraction of sequence steps whose class labels are predicted correctly and macro-average accuracy is the average percentage of time a class is classified

⁷One example of scoring function is $pos - neg$, where pos and neg are the number of positive and negative examples covered by the clause, respectively

⁸With respect to the search for each feature, the search strategy and scoring procedure of B&B are optimal for the objective. However, with respect to the overall search for the best set of features, the B&B strategy will generally be sub-optimal. Nevertheless, we chose greedy hill climbing (which is sub-optimal with respect to the search for each feature and overall set of features) for two reasons viz., 1. While B&B adopts different scoring functions for individual feature search and overall feature set search, our approach uses the same scoring function for both of these. 2. The scoring function in the greedy hill climbing strategy is the accuracy obtained using a probabilistic model (HMM) for which no reasonable bounds are known.

⁹Here we consider the problem of improving the overall accuracy of labeling. The score is a heuristic score and therefore we chose the one that is similar to our objective. Nevertheless, any other scoring function such as that used in cost sensitive learning can be used here.

correctly as given in the expressions adapted from (van Kasteren et al. 2008).

$$\text{Micro average accuracy} : \frac{\sum_{t=1}^T [\text{inferred}(t) = \text{true}(t)]}{T} . \quad (3.8)$$

$$\text{Macro average accuracy} : \frac{1}{n} \sum_{c=1}^n \left\{ \frac{\sum_{t=1}^{T_c} [\text{inferred}_c(t) = \text{true}_c(t)]}{T_c} \right\} . \quad (3.9)$$

where $[a = b]$ is an indicator giving 1 when true and 0 otherwise, T is the total number of time steps, n is the number of classes and T_c is the number of time steps for the class c .

In data that are skewed towards some labels, predicting a frequent label for all the time slices gives better micro-average accuracy but a bad macro-average accuracy. Therefore, if the data set is skewed and some critical classes have less support, we suggest maximizing the macro-average accuracy. In all other cases, we suggest maximizing micro-average accuracy. This is because the macro-average accuracy computation does not consider the size of a particular class and its maximization leads to a situation where unimportant classes that occur seldom have more impact on the overall efficiency of the model. The most typically used performance evaluation measure among the two is micro-averaged accuracy. However macro-average accuracy being too low is considered to be a poor performance. We have performed separate experiments with micro-average accuracy and macro-average accuracy as the scoring function. Trying a combination of both is a future work direction. We now discuss the overall learning algorithm for model construction.

During the training phase, we pursue a greedy hill climbing search in the lattice to find a model. The pseudo code for our approach is given in Figure 3.2. Initially, the features for each label are constructed with each of the individual inputs and an initial model is trained. In every iteration, candidate models are constructed by removing the features of each label one at a time as shown in step 8 of the pseudo code. Step 10 constructs new features by combining the features removed in step 8 with other features of the label and a new candidate model is trained. The best scoring model among all the candidate models, if better than the previous model, is saved. To evaluate a model, an HMM is constructed from the current emission model and the transition distribution. Each of the conjunctions discovered forms a column in the emission probability matrix and the conditional probabilities are learned for these conjunctions given label. Further,

```

1. procedure FIHMM_MODEL_CONSTRUCTION
2.    $featureSet \leftarrow$  features representing each individual input
3.    $currentModel \leftarrow$  model trained with  $featureSet$ 
4.   repeat
5.      $previousModel \leftarrow currentModel$ 
6.     for each label  $i$  do
7.       for each feature  $j$  of label  $i$  do
8.          $modelDel(i, j) \leftarrow$  model trained with  $j^{th}$  feature of  $i^{th}$  label dropped
9.         for each feature  $k$  of label  $i$  do
10.           $modelAdd(i, j, k) \leftarrow$  model trained with features  $j$  and  $k$  combined
11.          . to form new feature of label  $i$ 
12.        end for
13.      end for
14.       $currentModel \leftarrow \arg \max \left\{ \arg \max_{i,j} modelDel(i, j).accuracy, \right.$ 
15.      .  $\left. \arg \max_{i,j,k} modelAdd(i, j, k).accuracy \right\}$ 
16.    until  $currentModel.accuracy \leq previousModel.accuracy$ 
17.  return  $previousModel$ 
18. end procedure

```

Figure 3.2: Feature induction assisted HMM model training

only those features that are mapped to a label have to be considered during inference. Each iteration either deletes or adds a feature to the final model based on the HMM evaluation on part of the training data. Unlike traditional approaches, no examples are removed during the iterations. In each iteration, the existing logical model is refined, probabilistic parameters are learned and the model is evaluated on part of the training data. The process is repeated until convergence.

Greedy feature induction approaches, though better than conventional approaches, do not give any guarantee on optimality of the model learned. In the next section, we introduce our hierarchical kernel based optimal feature learning approach for structured output spaces (StructHKL). We then investigate the possibility of efficiently learning complex features for sequence labeling by leveraging the StructHKL approach.

3.4 Hierarchical Kernel Learning for Structured Output Spaces

In section 3.2, we have categorized relational features and identified Simple Conjuncts (\mathcal{SC}) and Composite Features (\mathcal{CF}) as powerful features for sequence labeling tasks. In the previous section, we discussed our greedy feature construction approach for learning \mathcal{SC} s as features for Hidden Markov Models. Since greedy approaches cannot guarantee optimal models, we propose a Hierarchical Kernels based feature learning approach for structured output spaces, which can learn optimal \mathcal{SC} s. We refer to this approach as Hierarchical Kernel Learning for Structured Output Spaces (StructHKL).

In this section, we derive our approach in general settings of features (for structured output spaces) that follow a partial order and then discuss the possibility of learning \mathcal{SC} s using this approach. The size of such an ordering is 2^N for N basic inputs. We develop our approach, that derives from the norm employed in RELHKL (Jawanpuria et al. 2011) and uses the loss function of StructSVM (Tsochantaridis et al. 2004, Tsochantaridis 2006) to discover optimal discriminative features from the partial order for solving sequence labeling problems. We build on the Support Vector Machines for Structured Output Spaces (StructSVM) model (Tsochantaridis et al. 2004, Tsochantaridis 2006) for sequence prediction problems. The StructSVM objective posed by Tsochantaridis et al. (2004) is given in equation (2.2). We modify the feature vector to include all possible label specific features (that are compositions of basic features), while preserving all possible transitions. The following paragraph explains the notations we use in the derivations of our approach.

We have presented the training and inference objectives of sequence labeling problems in equations (3.3), (3.4) and (3.5), where the features and feature weights are represented by ψ and \mathbf{f} respectively. Elements of ψ correspond to the input/observation (emission) features and the transition features. We represent the emission and transition parts of the vector ψ as ψ_E and ψ_T respectively. We assume that both ψ_E and ψ_T are vectors of dimension equal to the dimension of ψ with zero values for all elements not in their context. That is, ψ_E has dimension of ψ , but has zero values corresponding to the transition elements. In a similar spirit, we split the feature weight vector \mathbf{f} to \mathbf{f}_E and \mathbf{f}_T . Similarly, \mathcal{V} , the indices of the elements of ψ , is split into \mathcal{V}_E and \mathcal{V}_T . Our proposed approaches are to discover discriminative input features (ψ_E) that capture complex rela-

tionships among input variables. To learn emission features, ψ_E has to include all possible (composition/conjunction) features, which has a partial ordering. The top node in the partial order is the empty node and the bottom node is constructed from all the basic features. The nodes at level 1, denoted by B , are basic features themselves. For the sake of visualization, we assume there is a partial order for each label. Therefore, elements of ψ_E vector correspond to the nodes in the partial ordering of features for each label. As followed in (Jawanpuria et al. 2011), $D(v)$ and $A(v)$ represent the set of descendants and ancestors of the node v in the lattice. Both $D(v)$ and $A(v)$ include node v . The hull and the sources of any subset of nodes $\mathcal{W} \subset \mathcal{V}$ are defined as $\text{hull}(\mathcal{W}) = \bigcup_{w \in \mathcal{W}} A(w)$ and $\text{sources}(\mathcal{W}) = \{w \in \mathcal{W} | A(w) \cap \mathcal{W} = \{w\}\}$ respectively. In other words, the hull of a set of nodes is the set of all ancestors of the set and sources of a set of nodes are those nodes in the set that have no ancestor in the set other than itself. The size of set \mathcal{W} is denoted by $|\mathcal{W}|$. $\mathbf{f}_{\mathcal{W}}$ is the vector with elements as $f_v, v \in \mathcal{W}$. Also let the complement of \mathcal{W} denoted by \mathcal{W}^c be the set of all features belonging to the same label that are not in \mathcal{W} . Further let the input/observation at p^{th} sequence step of the i^{th} example be \mathbf{x}_i^p , where \mathbf{x}_i^p is a vector of binary values. Each element of the vector represents the value of an input at the position p . For instance, in activity recognition, these binary values represent the values of sensors fixed at locations such as groceries cupboard, bathroom door *etc.* at p^{th} time step. Similarly, output/label at p^{th} time step of the i^{th} example is represented by y_i^p . Let y_i^p can take any of n values. Other notations are consistent with that given in the previous sections.

To use the hierarchical ρ -norm regularizer on the feature weights corresponding to the emission nodes, we separate the regularizer term into those corresponding to emission and transition features. Since all the state transitions are to be preserved, a conventional 2-norm regularizer is used for transition features. The new SVM formulation is,

$$\begin{aligned}
& \min_{\mathbf{f}, \xi} \frac{1}{2} \Omega_E(\mathbf{f}_E)^2 + \frac{1}{2} \Omega_T(\mathbf{f}_T)^2 + \frac{C}{m} \sum_{i=1}^m \xi_i, \\
& \forall i, \forall Y \in \mathcal{Y} \setminus Y_i : \langle \mathbf{f}, \psi_i^\delta(Y) \rangle \geq 1 - \frac{\xi_i}{\Delta(Y_i, Y)} \\
& \forall i : \xi_i \geq 0
\end{aligned} \tag{3.10}$$

where $\Omega_E(\mathbf{f}_E)$ is defined in (2.3) as $\sum_{v \in \mathcal{V}_E} d_v \|\mathbf{f}_{E D(v)}\|_\rho$, $\rho \in (1, 2]$ and $\Omega_T(\mathbf{f}_T)$ is the 2-norm

regularizer $(\sum_j f_{Tj}^2)^{\frac{1}{2}}$

The 1-norm in $\Omega_E(\mathbf{f}_E)$ forces many of the $\|\mathbf{f}_{ED(v)}\|_\rho$ to be zero. Even in cases where $\|\mathbf{f}_{ED(v)}\|_\rho$ is not forced to zero, the ρ -norm forces many of node v 's descendants to zero. Since transition feature space is not exponential, no sparsity is desired and therefore a 2-norm regularizer is sufficient for transition. The above SVM setup has two inherent issues which makes it a hard problem to solve. The first is that the regularizer, $\Omega_E(\mathbf{f}_E)$, consists of ρ -norm over descendants of each lattice node, which makes it exponentially expensive. The second problem is the exponential number of constraints for the objective. The rest of the section discusses how to solve the problem efficiently.

By solving (3.10), we expect most of the emission feature weights to be zero. As illustrated by Bach (2009) and Jawanpuria et al. (2011), the solution to the problem when solved with the original set of features is the same but requires less computation when solved only with features having non zero weights at optimality. Therefore, an active set algorithm can be employed to incrementally find the optimal set of non zero weights (Bach 2009, Jawanpuria et al. 2011). In each iteration of the active set algorithm, since the constraint set in (3.10) is exponential, a cutting plane algorithm has to be used to find a subset of constraints of polynomial size so that the corresponding solution satisfies all the constraints with an error not more than ϵ . We now modify (3.10) to consider only the active set of features \mathcal{W} .

$$\begin{aligned} \min_{\mathbf{f}, \xi} \quad & \frac{1}{2} \left(\sum_{v \in \mathcal{W}} d_v \|\mathbf{f}_{ED(v) \cap \mathcal{W}}\|_\rho \right)^2 + \frac{1}{2} \|\mathbf{f}_T\|_2^2 + \frac{C}{m} \sum_{i=1}^m \xi_i, \\ \forall i, \forall Y \in \mathcal{Y} \setminus Y_i : \quad & - \left(\sum_{v \in \mathcal{W}} \langle f_{Ev}, \psi_{Evi}^\delta(Y) \rangle + \sum_{v \in \mathcal{V}_T} \langle f_{Tv}, \psi_{Tvi}^\delta(Y) \rangle + \frac{\xi_i}{\Delta(Y_i, Y)} - 1 \right) \leq 0 \\ \forall i : \quad & -\xi_i \leq 0 \end{aligned} \tag{3.11}$$

where $\rho \in (1, 2]$

The active set algorithm can be terminated when the solution to the small problem (reduced solution) is the same as the solution to the original problem; otherwise the active set has to be updated. We follow a similar approach to that defined in (Jawanpuria et al. 2011) for deriving a sufficiency condition to check optimality, which we discuss in the following paragraphs.

Applying lemma 26 in (Micchelli & Pontil 2005), the regularizer term corresponding

to the emission weights in (3.10) can be written as,

$$\Omega_E(\mathbf{f}_E)^2 = \min_{\gamma \in \Delta_{|\mathcal{V}_E|,1}} \min_{\lambda_v \in \Delta_{|D(v)|,\hat{\rho}} \forall v \in \mathcal{V}_E} \sum_{w \in \mathcal{V}_E} \delta_w^{-1}(\gamma, \lambda) \|f_{Ew}\|_2^2$$

where, $\hat{\rho} = \frac{\rho}{2-\rho}$, $\Delta_{d,r} = \left\{ \boldsymbol{\eta} \in \mathbb{R}^d \mid \boldsymbol{\eta} \geq 0, \sum_{i=1}^d \eta_i^r = 1 \right\}$, and $\delta_w^{-1}(\gamma, \lambda) = \sum_{v \in A(w)} \frac{d_v^2}{\gamma_v \lambda_{wv}}$.

Using the variational characterization and representer theorem (Rakotomamonjy et al. 2008), the partial dual (wrt. $\mathbf{f}, \boldsymbol{\xi}$) of (3.10) can be derived as,

$$\min_{\gamma \in \Delta_{|\mathcal{V}_E|,1}} \min_{\lambda_v \in \Delta_{|D(v)|,\hat{\rho}} \forall v \in \mathcal{V}_E} \max_{\boldsymbol{\alpha} \in \tau(\mathcal{Y}, C)} G(\gamma, \lambda, \boldsymbol{\alpha}) \quad (3.12)$$

where

$$G(\gamma, \lambda, \boldsymbol{\alpha}) = \sum_{i,Y \neq Y_i} \alpha_{iY} - \frac{1}{2} \boldsymbol{\alpha}^\top \left(\sum_{w \in \mathcal{V}_E} \delta_w(\gamma, \lambda) \boldsymbol{\kappa}_{Ew} \right) \boldsymbol{\alpha} - \frac{1}{2} \boldsymbol{\alpha}^\top \boldsymbol{\kappa}_T \boldsymbol{\alpha}$$

and

$$\tau(\mathcal{Y}, C) = \{ \boldsymbol{\alpha} \in \mathbb{R}^{m(n^l-1)} \mid \alpha_{i,Y} \geq 0, m \sum_{Y \neq Y_i} \frac{\alpha_{iY}}{\Delta(Y, Y_i)} \leq C, \forall i, Y \}$$

The kernel functions $\boldsymbol{\kappa}_{Ew}$ and $\boldsymbol{\kappa}_T$ are those corresponding to emission kernel at node w and the transition respectively. This is briefly discussed in the appendix section A.3. Now, let the duality gap with $(\gamma, \lambda, \boldsymbol{\alpha})$ in (3.12) be given by

$$\begin{aligned} & \max_{\hat{\boldsymbol{\alpha}} \in \tau(\mathcal{Y}, C)} G(\gamma, \lambda, \hat{\boldsymbol{\alpha}}) - \min_{\hat{\gamma} \in \Delta_{|\mathcal{V}_E|,1}} \min_{\hat{\lambda}_v \in \Delta_{|D(v)|,\hat{\rho}} \forall v \in \mathcal{V}_E} G(\hat{\gamma}, \hat{\lambda}, \boldsymbol{\alpha}) \\ & \leq \frac{1}{2} \Omega_E(\mathbf{f}_E)^2 + \frac{1}{2} \Omega_T(\mathbf{f}_T)^2 + \frac{C}{m} \sum_i \xi_i \\ & \quad - \left(\min_{\hat{\gamma} \in \Delta_{|\mathcal{V}|,1}} \min_{\hat{\lambda}_v \in \Delta_{|D(v)|,\hat{\rho}} \forall v \in \mathcal{V}_E} \sum_{i,Y \neq Y_i} \alpha_{iY} - \frac{1}{2} \sum_{w \in \mathcal{V}_E} \delta_w(\gamma, \lambda) \boldsymbol{\alpha}^\top \mathbf{K}_{Ew} \boldsymbol{\alpha} - \frac{1}{2} \boldsymbol{\alpha}^\top \mathbf{K}_T \boldsymbol{\alpha} \right) \end{aligned}$$

From this, we can derive a sufficient condition for the reduced solution with \mathcal{W} to have a duality gap less than ϵ as,

$$\begin{aligned} & \max_{u \in \text{sources}(\mathcal{W}^c)} \sum_{i,Y \neq Y_i} \sum_{j,Y' \neq Y_j} \boldsymbol{\alpha}_{\mathcal{W}iY}^\top \sum_{p=1}^{l_i} \sum_{q=1}^{l_j} 2 \left(\prod_{k \in u} \frac{\psi_{Ek}(\mathbf{x}_i^p) \psi_{Ek}(\mathbf{x}_j^q)}{b^2} \right) \\ & \quad \left(\prod_{k \notin u} \left(1 + \frac{\psi_{Ek}(\mathbf{x}_i^p) \psi_{Ek}(\mathbf{x}_j^q)}{(1+b)^2} \right) \right) \boldsymbol{\alpha}_{\mathcal{W}jY'} \\ & \leq \Omega_E(\mathbf{f}_{E\mathcal{W}})^2 + \Omega_T(\mathbf{f}_{T\mathcal{W}})^2 + 2(\epsilon - e_{\mathcal{W}}) \end{aligned} \quad (3.13)$$

where $e_{\mathcal{W}} = \Omega_E(\mathbf{f}_{\mathbf{E}\mathcal{W}})^2 + \Omega_T(\mathbf{f}_{\mathbf{T}\mathcal{W}})^2 + \frac{C}{m} \sum_i \xi_i + \frac{1}{2} \boldsymbol{\alpha}_{\mathcal{W}}^\top \boldsymbol{\kappa}_{\mathbf{T}} \boldsymbol{\alpha}_{\mathcal{W}} - \sum_{i, Y \neq Y_i} \alpha_{\mathcal{W}iY}$.

If the current solution satisfies the above condition in any iteration of the active set, the algorithm terminates; else the active set is updated by adding the nodes in $\text{sources}(\mathcal{W}^c)$ which violate the condition. To solve the optimization problem efficiently, we now derive the complete dual of (3.10) from the partial dual (3.12) as,

$$\min_{\boldsymbol{\eta} \in \Delta_{|\mathcal{V}|,1}} g(\boldsymbol{\eta}) \quad (3.14)$$

where $g(\boldsymbol{\eta})$ is defined as,

$$\max_{\boldsymbol{\alpha} \in \boldsymbol{\tau}(\mathcal{Y}, C)} \sum_{i, Y \neq Y_i} \alpha_{iY} - \frac{1}{2} \boldsymbol{\alpha}^\top \boldsymbol{\kappa}_{\mathbf{T}} \boldsymbol{\alpha} - \frac{1}{2} \left(\sum_{w \in \mathcal{V}} \zeta_w(\boldsymbol{\eta}) (\boldsymbol{\alpha}^\top \boldsymbol{\kappa}_{\mathbf{E}w} \boldsymbol{\alpha})^{\bar{\rho}} \right)^{\frac{1}{\bar{\rho}}}, \quad (3.15)$$

and $\zeta_w(\boldsymbol{\eta}) = \left(\sum_{v \in A(w)} d_v^\rho \eta_v^{1-\rho} \right)^{\frac{1}{1-\rho}}$ and $\bar{\rho} = \frac{\rho}{2(\rho-1)}$.

Since (3.14) is a 1-norm regularized problem, many of the η s are expected to be zero at optimality. A zero value for η at node v makes the weights $\zeta_w(\boldsymbol{\eta})$ of all of v 's descendant nodes w to be equal to zero and essentially discourages selection of kernels near the bottom of the lattice. It can be shown that the maximization term is similar to a $\hat{\rho}$ -norm ($\hat{\rho} = \frac{\rho}{2-\rho}$) MKL formulation (Kloft et al. 2009). If $\rho \in (1, 2)$, the kernel κ_w may not be selected even in cases when $\zeta_w(\boldsymbol{\eta}) \neq 0$ (Jawanpuria et al. 2011). Therefore the formulation ensures that large conjunctive features are not selected and that selection of a feature does not warrant selection of its subsets.

The solution to the dual problem in (3.14) with \mathcal{V} restricted to the active set \mathcal{W} gives the solution to the restricted primal problem given in (3.11). The active set algorithm, adapted from Jawanpuria et al. (2011), is briefly outlined in Figure 3.3¹⁰. It starts with the top nodes in the lattice and iteratively adds new nodes that violate the sufficiency condition. The algorithm terminates when no new nodes violate the sufficiency condition. Parameters are updated in each step of the active set by solving (3.14). We follow the mirror descent (Ben-Tal & Nemirovskiaei 2001) approach to solve (3.14) as done by Jawanpuria et al. (2011).

¹⁰Although, the derivations and procedures in this work and that in Jawanpuria et al. (2011) are different, the structure of active set algorithm is same. It has been included here for completion.

Let $\bar{\alpha}$ be the optimal solution to (3.15) with some η , then the i^{th} sub-gradient is computed as

$$(\nabla g(\eta))_i = -\frac{d_i^\rho \eta_i^{-\rho}}{2\bar{\rho}} \left(\sum_{w \in \mathcal{V}_{\mathbf{E}}} \zeta_w(\eta) (\bar{\alpha}^\top \kappa_{\mathbf{E}w} \bar{\alpha})^{\bar{\rho}} \right)^{\frac{1}{\bar{\rho}}-1} \left(\sum_{w \in D(i)} \zeta_w(\eta)^\rho (\bar{\alpha}^\top \kappa_{\mathbf{E}w} \bar{\alpha})^{\bar{\rho}} \right) \quad (3.16)$$

To compute the gradient, $\bar{\alpha}$ is to be obtained by solving (3.15) using the cutting plane method. The cutting plane algorithm, adapted from Tsochantaridis et al. (2004), is outlined in Figure 3.4¹¹. The algorithm starts with no constraints for (3.11) and in each step, adds a constraint that most violates the margin. The dual problem (3.15) is then solved and the process is continued. The algorithm stops when there are no more margin violations. We develop a modified version of Viterbi algorithm (Forney 1973) to add constraints in each iteration of the cutting plane algorithm (Inference for predicting the label sequence for test data is also performed in a similar way). Derivations in detail are explained in Appendix section A.4.

In this section, we derived an approach that discovers an optimal set of discriminative features from the feature space of emission features. We use the active set algorithm to handle exponential feature space. Each active set iteration solves the dual formulation using mirror descent algorithm. The subproblem in the mirror descent step is solved efficiently by a cutting plane algorithm that handles the exponential constraint space. In general, StructHKL can be used in structured output classification problems to learn from complex feature spaces that can be ordered as a directed acyclic graph and where the summation of descendant kernels can be computed in polynomial time in the number of basic inputs (Bach 2009). We briefly discuss the possibility of learning \mathcal{SC} s using StructHKL.

An \mathcal{SC} is defined as a conjunction of basic boolean inputs at any single sequence position. The space of conjunctions can be ordered in the form of a partial order, with an empty node as the top node and the conjunction of all basic inputs as the bottom node. This space of features is similar to the space explored by Jawanpuria et al. (2011), but for a different problem setting. While Jawanpuria et al. (2011) explored this space in a binary classification setting, we explore this space (we have this ordering for each label)

¹¹Although, the derivations and procedures in this thesis and that in Tsochantaridis et al. (2004) are different, the structure of cutting plane algorithm is the same. It has been included here for completion.

Input: Training data D , Oracle for computing kernels, Maximum tolerance ϵ

1. Initialize $\mathcal{W} = \text{Top nodes}$ in the lattice as the active set
2. Compute $\boldsymbol{\eta}, \boldsymbol{\alpha}$ by solving (3.14) using mirror descent
3. **while** sufficiency condition is not satisfied, **do**
4. Add sufficiency condition violating nodes to active set \mathcal{W}
5. Recompute $\boldsymbol{\eta}, \boldsymbol{\alpha}$ by solving (3.14)
6. **end while**
7. **Output:** Active-set $\mathcal{W}, \boldsymbol{\eta}, \boldsymbol{\alpha}$

Figure 3.3: Active set algorithm for solving StructHKL.

Input: kernels, C , ϵ_{margin}

1. $S_i \leftarrow \phi \quad \forall i = 1, \dots, m$
2. **repeat**
3. **for** $i = 1, \dots, m$ **do**
4. Define $H(Y) \equiv \left[1 - \langle \mathbf{f}, \psi_i^\delta(Y) \rangle\right] \Delta(Y_i, Y)$
5. Compute $\hat{Y} = \arg \max_{Y \in \mathcal{Y}} H(Y)$.
6. Compute $\xi_i = \max\{0, \max_{Y \in S_i} H(Y)\}$.
7. **if** $H(\hat{Y}) > \xi_i + \epsilon_{\text{margin}}$, **then**
8. $S_i \leftarrow S_i \cup \{\hat{Y}\}$.
9. $\boldsymbol{\alpha} \leftarrow$ optimize dual over S , $S = \bigcup_i S_i$.
10. **end if**
11. **end for**
12. **until** no S_i has changed during the iteration.

Figure 3.4: Cutting plane algorithm for solving dual with a fixed $\boldsymbol{\eta}$.

in a multi-class structured output classification setting. In the case of \mathcal{SC} s, the kernel κ_v at node v for sequence positions p and q (of examples i and j respectively) $\kappa_v(x_i^p, x_j^q)$ is the kernel induced by the v^{th} conjunction in the partial order evaluated at the p^{th} and q^{th} positions of examples i and j respectively. Therefore, $\kappa_v(x_i^p, x_j^q)$ is the product of v^{th} conjunction evaluated at (i, p) and (j, q) , which is equal to the product of basic boolean inputs in v evaluated at (i, p) and (j, q) . As in (Jawanpuria et al. 2011), for a sub-space formed by descendants of a node in the ordering, this sum of products can be written as

products of sums. For instance, $\sum_{v \in \mathcal{V}} \kappa_v(x_i^p, x_j^q) = \prod_{k \in B} (1 + \psi_k(x_i^p) \psi_k(x_j^q))$, where B is the set of nodes in level 1 (basic inputs). Therefore, StructHKL can be employed to discover \mathcal{SC} s efficiently. We now look into the category of complex relational features that capture sequential information among input variables to build efficient sequence labeling models.

3.5 Learning Complex Relational Features for Sequence Labeling

Our objective is to learn complex relational features (\mathcal{CF}), that are derived from inputs at different relative positions. TildeCRF (Gutmann & Kersting 2006) is an existing approach that explores such feature space, using relational regression trees. However the approach pursued is greedy. Here we look into the possibility of leveraging optimal feature learning approaches to explore such feature space.

Although, the StructHKL algorithm optimally solves the objective of learning the most discriminative \mathcal{SC} s for sequence labeling, its applicability in learning complex relational features, that are derived from inputs at different relative positions, is non-trivial and challenging. In section 3.2, we have identified \mathcal{CF} s as powerful set of features that capture relationships among inputs at multiple sequence positions. Composite Features (\mathcal{CF}) are particularly interesting in our case, because they capture most of the semantics of definite clauses and do not allow clauses that are not complete in its meaning. For instance, the example clause 4 in section 3.2 does not give any information about the newly introduced time variable T2, and thus is not a \mathcal{CF} . In section 3.2, we have proved that composite features (\mathcal{CF}) can be constructed from absolute features (\mathcal{AF}) with unary/multiple conjunctions without unifications and that \mathcal{AF} s can be constructed from primary features (\mathcal{PF}) with unifications. Therefore, the space of \mathcal{CF} s can be defined as a partial order over \mathcal{PF} s with unifications and conjunctions. However, since \mathcal{CF} s (and \mathcal{AF} s) share local variables across predicates in the condition part and the refinement of such a clause is performed by operators such as unification and anti-unification, StructHKL cannot be applied to learn \mathcal{CF} s (and \mathcal{AF} s). It is easy to observe that, if \mathcal{AF} s can be constructed by other approaches, StructHKL can be employed to efficiently construct \mathcal{CF} s from \mathcal{AF} s. We identify two possibilities to meet this objective, *viz.* (i) enumerating \mathcal{AF} s and discovering their useful compositions (\mathcal{CF}) using StructHKL or (ii) developing

methods to learn optimal \mathcal{AF} s (or \mathcal{CF} s directly). We now study the former choice of constructing \mathcal{CF} s from a set of enumerated \mathcal{AF} s.

3.5.1 Constructing Composite Features from Enumerated Absolute Features

Since the ordering of \mathcal{CF} s is based on complex refinement operators such as unification and anti-unification, the ordering does not comply with the requirement of StructHKL that the summation over descendant kernels should be able to be computed in polynomial time. Therefore, as \mathcal{CF} s are conjunctions of \mathcal{AF} s, we enumerate all \mathcal{AF} s and construct an ordering of their conjunctions. Considering \mathcal{AF} s as individual basic features (features at level 1), the new setting can be viewed as the projection of \mathcal{CF} s into the space of \mathcal{SC} s. StructHKL can be employed to discover \mathcal{CF} s from this new ordering.

The space of \mathcal{AF} s is prohibitively large and therefore it is not feasible to enumerate all \mathcal{AF} s in a domain. We therefore, propose to selectively enumerate \mathcal{AF} s based on some relevance criteria such as support of the \mathcal{AF} in the training set. We define an \mathcal{AF} as strongly relevant if it helps the classification model to discern classes optimally. On the other hand, we consider a feature to be weakly relevant if it covers at-least a threshold percentage of examples. Since discovering strongly relevant \mathcal{AF} s is a hard task, we discover weakly relevant \mathcal{AF} s using Inductive Logic Programming tools (Dehaspe & Toivonen 1999, Dehaspe & Toironen 2000). Once a set of relevant \mathcal{AF} s are enumerated, StructHKL can be employed to learn useful compositions of \mathcal{AF} s and their parameters to get the final model. We now discuss leveraging relational kernels to implicitly learn relational features efficiently.

3.5.2 Leveraging Complex Relational Kernels for Sequence Labeling

As discussed in the previous subsection, since the partial ordering of \mathcal{AF} s does not comply with the requirements of StructHKL, it is not feasible to leverage StructHKL for learning features in the space of \mathcal{AF} s (and its super-space of \mathcal{CF} s). For this reason, in the sequence labeling model, we leverage a relational kernel that computes the similarity between instances in an implicit feature space of \mathcal{CF} s. To this end, we employ the relational subsequence kernel (Bunescu & Mooney 2006) at each sequence position (over a

time window of inputs around the pivot position) for the classification model. We now briefly discuss about the relational subsequence kernels in the following paragraph.

Relational subsequence kernels have been used to extract relations between entities in natural language text (Bunescu & Mooney 2006), where the relations are between protein names in biomedical texts. The features are (possibly non-contiguous) sequences of word and word classes anchored by the protein names at their ends. They extend the string kernels (Lodhi et al. 2002) for this task.

We have defined \mathcal{CF} s as features that capture the subset of features at the current position as well as its relative positions. To implicitly capture this feature space, we employ a relational subsequence kernel at each position of the input sequence, with the current position as the pivot position. Suppose we consider an input \mathbf{x}_i^p at position p for example i . Let the previous k positions relative to p have inputs $\mathbf{x}_i^{p-1}, \dots, \mathbf{x}_i^{p-k}$ and next l positions relative to p have inputs $\mathbf{x}_i^{p+1}, \dots, \mathbf{x}_i^{p+l}$. Let there be N basic features at a time-step t denoted by $x^{1^t} \dots x^{N^t}$ ¹². Essentially our sequence for the particular time-step pivoted at p denoted by Q^p is as follows :

$$Q^p = \{x^{1^{p-k}}, \dots, x^{N^{p-k}}\}, \dots, \{x^{1^{p-1}}, \dots, x^{N^{p-1}}\}, \{x^{1^p}, \dots, x^{N^p}\}, \{x^{1^{p+1}}, \dots, x^{N^{p+1}}\} \dots \{x^{1^{p+l}}, \dots, x^{N^{p+l}}\}$$

Given two sequences Q^p and Q^q , we define the relational subsequence kernel $SSK(Q^p, Q^q)$ as elaborated in (Bunescu & Mooney 2006). This kernel will implicitly enumerate all possible common subsequences between Q^p and Q^q . We now show that the feature space of \mathcal{CF} s are indeed that captured by our relational subsequence kernel.

Claim 7. Relational subsequence kernels implicitly enumerate all the features in the feature space defined by Composite Features (\mathcal{CF}) given a constant context window.

Proof. By definition the relational subsequence kernel $SSK(Q^p, Q^q)$ will implicitly enumerate all possible common subsequences between Q^p and Q^q . \mathcal{CF} s are conjunctions of features in the present time-step with features present in time-steps before and after the current time-step. Since we are considering all the sub-sequences in the given context (time) window in the relational kernel, we implicitly enumerate space of \mathcal{CF} s. \square

We now discuss the incorporation of relational subsequence kernels in the StructSVM formulation.

¹²Ignoring the example number i for simplicity

We define the kernel for StructSVM framework below, which represents the kernel resulting from the difference in values for the original and the candidate sequences. This stands for the inner product, $\langle \boldsymbol{\psi}_i^\delta(Y), \boldsymbol{\psi}_j^\delta(Y') \rangle$ with $\boldsymbol{\psi}_j^\delta(Y)$ defined as $\boldsymbol{\psi}(X_i, Y_i) - \boldsymbol{\psi}(X_i, Y)$. The kernel is,

$$\kappa^\delta((X_i, Y_i, Y), (X_j, Y_j, Y')) = \kappa_T^\delta(Y_i, Y, Y_j, Y') + \kappa_E^\delta((X_i, Y_i, Y), (X_j, Y_j, Y')) \quad (3.17)$$

where $\kappa_T^\delta(\cdot)$ and $\kappa_E^\delta(\cdot)$ stand for the transition and emission parts of the kernel $\kappa^\delta(\cdot)$ and are defined below.

$$\kappa_T^\delta(Y_i, Y, Y_j, Y') = \kappa_T(Y_i, Y_j) + \kappa_T(Y, Y') - \kappa_T(Y_i, Y') - \kappa_T(Y_j, Y), \quad (3.18)$$

$$\begin{aligned} \kappa_T(Y_i, Y_j) &= \sum_{p=1}^{l_i-1} \sum_{q=1}^{l_j-1} \Lambda(y_i^p, y_j^q) \Lambda(y_i^{p+1}, y_j^{q+1}) \\ &= \sum_{p=2}^{l_i} \sum_{q=2}^{l_j} \Lambda(y_i^{p-1}, y_j^{q-1}) \Lambda(y_i^p, y_j^q), \end{aligned} \quad (3.19)$$

$\Lambda(y_i^p, y_j^q) = 1$ if $y_i^p = y_j^q$; 0 otherwise. and

$$\kappa_E^\delta((X_i, Y_i, Y), (X_j, Y_j, Y')) = \sum_{p=1}^{l_i} \sum_{q=1}^{l_j} \kappa_E(\mathbf{x}_i^p, \mathbf{x}_j^q) \left(\Lambda(y_i^p, y_j^q) + \Lambda(y_i^p, y_j'^q) - \Lambda(y_i^p, y_j'^q) - \Lambda(y_i^p, y_j^q) \right) \quad (3.20)$$

Relational subsequence kernels can be employed to compute $\kappa_E(\mathbf{x}_i^p, \mathbf{x}_j^q)$, where the kernel is computed from subsequences extracted from a window of sequence positions around the positions p and q for examples i and j respectively, where p and q are pivots.

The dual of (2.2) with the new kernel can be written as,

$$\begin{aligned} \max_{\alpha} \quad & \sum_i \sum_{Y \in S_i} \alpha_{iY} - \frac{1}{2} \sum_i \sum_{Y \in S_i} \sum_j \sum_{Y' \in S_j} \alpha_{iY} \alpha_{jY'} \left(\kappa_T^\delta(Y_i, Y, Y_j, Y') + \kappa_E^\delta((X_i, Y_i, Y), (X_j, Y_j, Y')) \right) \\ \text{s.t.} \quad & \forall i, \forall Y \in S_i, \quad \alpha_{iY} \geq 0 \\ & \forall i, \quad m \sum_{Y \in S_i} \frac{\alpha_{iY}}{\Delta(Y_i, Y)} \leq C. \end{aligned} \quad (3.21)$$

Now the margin violation cost function for a candidate output sequence Y for example i (for the cutting plane algorithm) can be written as,

$$\begin{aligned}
H(Y) &= \left(1 - \langle \psi_i^\delta(Y), \mathbf{f} \rangle\right) \Delta(Y_i, Y) \\
&= \left(1 - \sum_j \sum_{y' \in S_j} \alpha_{jY'} \langle \psi_i^\delta(Y), \psi_j^\delta(Y') \rangle\right) \Delta(Y_i, Y) \\
&= \left(1 - \sum_j \sum_{y' \in S_j} \alpha_{jY'} \kappa^\delta((X_i, Y_i, Y), (X_j, Y_j, Y'))\right) \Delta(Y_i, Y) \tag{3.22}
\end{aligned}$$

where S_j is the active constraint set for example j .

The dual objective and the margin violation cost function can be plugged into the cutting plane algorithm to solve the objective. While this way of modeling does not result in interpretability, relational subsequence kernels do efficiently capture the relational sequential information on the inputs.

Although the main contribution of the thesis is feature learning for sequence labeling, we have also contributed in two related problem domains, that is, satisfiability based inference in probabilistic first order logical systems and dimensionality reduction in classification settings, which we discuss in the following chapters. The next chapter discusses our efficient search space reduction in satisfiability checking for inference in first order systems.

Chapter 4

Pruning Search Space for Weighted First Order Horn Clause Satisfiability

We have discussed our feature induction techniques in Chapter 3. In this chapter, we discuss our contribution in another area, that is, satisfiability based inference in probabilistic first order settings.

Inference in sequence labeling requires assigning a label to each observation instance in the sequence. As observations at successive time instances are related, it is intuitive that assigning labels to all instances at once yields better solutions. Globally assigning labels to all the instances is selecting a solution (a sequence of labels) from an exponential number of possible solutions (for a sequence of length T with n possible labels for each time step, there are n^T possible solutions), which is a difficult task. The problem could be more complex in first order settings, where models can have a complex first order structure, a huge set of groundings, and the like. If it is feasible to ground all the variables, this can be solved in polynomial time using a dynamic programming algorithm called the Viterbi algorithm (Forney 1973). In all other cases (sequence labeling or general), we derive a Satisfiability approach for fast and memory efficient inference¹. In our implementation for sequence labeling, we employ the Viterbi algorithm for predicting the actual sequence of labels. We first explain the background of our work in the following section.

4.1 Background

Representing sets of objects and their relationships in a compact form has been the focus of researchers for more than a decade. Weighted first order formulas proved to be one

¹Since we could not find such a sequence labeling data to demonstrate the validity of our approach, we present our work in a general setting

such representation also allowing inference and learning in a structured way. Satisfiability testing is one of the techniques for inference in these sets of formulas. We summarize some of the works that addressed the problem of satisfiability, in the next paragraph.

Traditional SAT solvers in propositional logic try to find an assignment for all the literals that makes all the clauses true. They return a model if it exists or return unsatisfiable. SAT solvers such as DPLL (Davis et al. 1962) give exact solutions but employ backtracking and take exponential time in the worst case. Local search methods for satisfying the maximum number of clauses (Max-SAT) have been implemented in GSAT (Selman et al. 1992), WalkSAT (Selman et al. 1993) *etc.*. As there could be many contradictions in real world, it is better to perform weighted satisfiability. Weighted Max-SAT problems assign weights to the clauses and aim to minimize the sum of the weights of unsatisfied clauses. MiniMaxSAT (Heras et al. 2008) uses a depth-first branch-and-bound search approach for satisfiability.

Satisfiability of first order logic (universally quantified Conjunctive Normal Form (CNF)) can be done by grounding all the clauses (exponential memory cost) and then running satisfiability as in a propositional case. Since, many learning techniques require the repeated use of inference and satisfiability, a complete grounding of the clauses becomes a bottle neck. Weighted satisfiability solvers are used for MPE/MAP inference in relational domains (Richardson & Domingos 2006). However, the complete grounding issue remained unsolved. A LazySAT approach (Singla & Domingos 2006) that does not ground all clauses was proposed for satisfiability in domains where a majority of ground atoms are false. Their approach, a variation of MaxWalkSAT, keeps track of all the clauses that can be affected when a literal in an unsatisfied clause is flipped². Recently, in (Shavlik & Natarajan 2009), the ground clauses that are satisfied by the evidence are excluded.

²In lazySAT, the unsatisfied clauses are obtained by simply going through each possible grounding of all the first order clauses and materializing the groundings that are unsatisfied. In contrast, our approach grounds clauses that have a potential to be unsatisfied. As we have restricted our case to horn clauses (where all clauses are satisfied if all atoms are false), we can explore the search space in a directed way. In contrast to LazySAT and walkSAT, we get an initial truth assignment in the end of pruning, which is very close to the final solution. Our approach is different from lazySAT, also in the sense that, we ground all relevant clauses before the maxWalkSAT step. The LazySAT paper reports running time for LazySAT that is comparable with maxWalkSAT. Whereas, our approach converges much quicker than maxWalkSAT.

The approach, which depends only on the evidence set, processes each clause independently and does not find the dependent clauses transitively. Mihalkova & Richardson (2009) cluster query literals and perform inference for cluster representatives. Queries are clustered by computing signatures using a recursive procedure based on adjacent nodes. Inference is performed for each cluster representative by running the MaxWalkSAT on the corresponding Markov Network constructed recursively. Alen Fern mentions the applicability of Forward Chaining in horn clauses (Fern 2009) but has not given any algorithm or proof for doing so. In the case of contradicting clauses, it is not straight forward to do forward chaining. We state the objective of our work in the next paragraphs.

We address the issue of complete grounding by restricting our domain to first order horn clauses and pruning the search space for satisfiability. Our approach caters to several real world applications that use the horn clausal language. We formally discuss our approach in the next section.

4.2 Satisfiability in Horn Clauses

The objective of Satisfiability checking is to find an interpretation that makes all (or maximum number of) the clauses in a system to be true. In horn clauses, if a set of horn clauses are fully satisfiable, then a minimal model can be found using T_{Σ} operator (referred to as the immediate consequence operator T_P in (Hogger 1990)) in polynomial time. However weighted unsatisfiable problems require to find the most likely state based on the weights. We propose an extension to the minimal model approach in which we find (i) the relevant set of ground horn clauses which has the potential to be part of a contradiction and (ii) an interpretation close to the result. The MaxSAT algorithm can be used on this subset of clauses, (optionally) starting from the interpretation returned, to get the most likely state. We also prove that the local search for optimality in the pruned space cannot affect the satisfiability of the rest of the clauses. We first give the intuition for our approach before going into the details.

If any of the atoms in the body part of a horn clause is false, then the clause is satisfied because of its inherent structure of containing one positive atom at the most, while all others are negative. The groundings of a set of first order horn clauses (Σ) with all the constants give a large set in which majority of the atoms are false in the real world. This makes a large subset of these clauses satisfied by default. We can neglect

these clauses and restrict our attention to the clauses that have a potential to be a part of a contradiction. We call this set, the relevant set (RS).

We propose an algorithm, *Modified* T_Σ , to identify the relevant set along with the truth assignments that are almost close to the result. A local search for optimality can be done on this set, starting with the interpretation returned, rather than considering the huge set of clauses and arbitrary truth assignments. Next, we explain T_Σ before going to the Modified version.

4.2.1 T_Σ Operator

T_Σ Operator provides a procedure to generate an interpretation from another. It builds on the concept that for satisfiability in horn clauses, all the unit clauses should be True and if the body of a clause is True, then the head should also be True. Let I_k be the interpretation at the k^{th} step of the operation. Then,

$$I_{k+1} = I_k \cup T_\Sigma(I_k) \quad (4.1)$$

$$\text{where,} \quad T_\Sigma(I) = \{A : A \leftarrow \text{body} \in \Sigma \quad \text{and} \quad \text{body} \subseteq I\} \quad (4.2)$$

If we start with $I = \emptyset$, and iteratively apply the above function assignment (with respect to the set of clauses), we will eventually converge at an interpretation that is the minimal model of the formulae if one exists. If there is no model for this set, the operation will reach a contradiction and will return *Unsatisfiable*.

In weighted satisfiability problems, if the given set is unsatisfiable, we need to get a most likely state based on the weights. MaxSAT algorithms can do this optimization. Since applying MaxSAT to the complete groundings is expensive, we improve the above method to prune the search space for MaxSAT. The *Modified* T_Σ Step described in the next section helps us to prune the search space.

4.2.2 *Modified* T_Σ Step

The *Modified* T_Σ operation returns a model, if one exists; Otherwise, it returns the set of clauses to be used by a local search algorithm and an initial interpretation for the local search. The method is outlined in Figure 4.1 and is explained below.

Start with applying T_Σ to the set of ground clauses until it converges in a model or some contradiction is attained. In the former case, we can stop and return the current

interpretation as the solution. If we land up in a contradiction, we get an atom whose truth value determines the set of clauses satisfied. We assign true to the atom and proceed further till no more clauses can be visited. All the clauses discovered by *Modified_T Σ* irrespective of whether satisfied or not form the relevant set. The interpretation which is obtained at the end of the algorithm can optionally be used as an initial truth assignment for the optimization step. Note that the truth values for evidences given are always true and cannot be changed.

Any Weighted satisfiability algorithm can be applied on the Relevant Set of clauses and the (optional) initial truth values to get a minimum cost interpretation. We now discuss the weighted satisfiability approach using *Modified_T Σ* .

4.3 Modified_Weighted_SAT

In the new approach, the *Modified_T Σ* operation is used to find the relevant subset as well as an initial truth assignment. Then, the weighted MaxSAT version presented in Figure 4.2 is used. Figure 4.3 shows the overall algorithm.

Lemma 4.1. *All the unsatisfied clauses will be in RS.*

Proof. A horn clause c' is unsatisfied if $c'.body \subseteq \{TS \cup DB\}$ and $c'.head \notin \{TS \cup DB\}$. Step 6 in *Modified_T Σ* adds all clauses c' of the form $(c'.head \vee \neg True)$ to RS irrespective of whether it is satisfied or not. Step 17 in *Modified_T Σ* adds all clauses c' where $c'.body \subseteq \{TS \cup DB\}$. This covers both the cases of $c'.head$ is *True* and $c'.head$ is *False*. All other clauses c'' where $c''.body \not\subseteq \{TS \cup DB\}$ are satisfied by default. So set of unsatisfied clauses is a subset of RS . \square

Lemma 4.2. *Any flip done in any maxSAT step to make an unsatisfied clause satisfiable only affects the satisfiability of clauses in RS.*

Proof. Let us prove this by contradiction.

Suppose a clause, $c' = (l_1 \vee \neg l_2 \vee \neg l_3 \vee \dots \vee \neg l_n)$ is not satisfied by the current assignments in $\{TS \cup DB\}$. This happens only when $l_1 \notin \{TS \cup DB\}$ and $\forall i = 2 \dots n \quad l_i \in \{TS \cup DB\}$. To make c' satisfied, there are two cases. case 1: flip l_1 , case 2: flip any of l_2, l_3, \dots, l_n .

case 1: Flip l_1 (*False* to *True*). Assume that flipping l_1 will affect the state of a clause $c'' \notin RS$. Since $c'' \notin RS$, $c''.body \not\subseteq \{TS \cup DB\}$. Otherwise step 17 in *Modified_T Σ*

Input: Σ , the set of first order clauses with weights; DB , evidence set given.

Output: RS , the set of clauses to be considered for optimization; TS , truth assignments of all atoms in RS except those in DB .

1. $TS := \emptyset$
2. $RS := \emptyset$
3. **for each** unit clause c in Σ **do**
4. **for each** grounding c' of c **do**
5. **if** $c' \notin RS$ **then**
6. Add c' to RS
7. **end if**
8. **if** $c'.head \notin \{TS \cup DB\}$ **then**
9. Add $c'.head$ to TS
10. **end if**
11. **end for**
12. **end for**
13. **repeat**
14. **for each** non unit clause c in Σ **do**
15. **for each** grounding c' of c where $c'.body \subseteq \{TS \cup DB\}$ **do**
16. **if** $c' \notin RS$ **then**
17. Add c' to RS
18. **end if**
19. **if** $c'.head \notin \{TS \cup DB\}$ **then**
20. Add $c'.head$ to TS
21. **end if**
22. **end for**
23. **end for**
24. **Until** no new clauses are added to the set RS
25. Return $\{RS, TS\}$

Figure 4.1: *Modified- T_Σ* algorithm

would have covered c'' and it would have been in RS . Also all the unit clauses are covered by step 6 in *Modified- T_Σ* .

Now let $c''.head = l_1$. Since flipping $c''.head$ to *True* changes the state of c'' , $c''.body \subseteq \{TS \cup DB\}$. If this is the case, c'' should have been covered by step 17 in *Modified.T_Σ* and would have been in RS . Hence the assumption that $c'' \notin RS$ is wrong.

Now let $l_1 \in c''.body$ and flipping it to *True* changes the state of c'' . Then $c''.body \setminus l_1 \subseteq \{TS \cup DB\}$. But applying our approach to c' would have made $l_1 \in TS$ and transitively $c''.body \subseteq \{TS \cup DB\}$ and $c'' \in RS$. Hence the assumption that $c'' \notin RS$ is wrong.

case 2: Flip any $l_i \in \{l_2, l_3, \dots, l_n\}$ (*True* to *False*). Assume that flipping l_i will affect the state of a clause $c'' \notin RS$. Since $c'' \notin RS$, $c''.body \not\subseteq \{TS \cup DB\}$. Otherwise step 17 in *Modified.T_Σ* would have covered c'' and it would have been in RS . Also all the unit clauses are covered by step 6 in *Modified.T_Σ*.

Now let $c''.head = l_i$. Since flipping $c''.head$ to *False* changes the state of c'' , $c''.body \subseteq \{TS \cup DB\}$. If this is the case, c'' should have been covered by step 17 in *Modified.T_Σ* and would have been in RS . Hence the assumption that $c'' \notin RS$ is wrong.

Now let $l_i \in c''.body$ and flipping it to *False* changes the state of c'' . Then before flipping, $c''.body \subseteq \{TS \cup DB\}$ which must have been covered by step 17 in *Modified.T_Σ* and $c'' \in RS$. Hence the assumption that $c'' \notin RS$ is wrong. \square

Lemma 4.3. *If α is the cost of an optimal solution to RS , then α is the cost of an optimal solution to Σ*

Proof. let μ and $\hat{\mu}$ be the cost of optimal solutions to Σ and RS respectively. That is μ should be the sum of costs of RS and $\Sigma \setminus RS$. Increase in cost occurs only because of contradictions and this is in the set RS (proved in claim 1). The best possible solution to the non contradicting part is zero. We get a minimum cost solution for the RS part using MaxSAT and any modification to that can result in (proved in claim 2 that this doesn't affect $\Sigma \setminus RS$) an increase in cost only in RS . Therefore $\mu = 0 + \hat{\mu}$ and thus $\mu = \hat{\mu}$. \square

Our approach has some similarities to the LazySAT approach (Singla & Domingos 2006) that adds clauses to the active set lazily (as and when required) in a general Markov Logic Network framework. LazySAT, a variation of WalkSAT, keeps track of all the clauses that can be affected when a literal is flipped. In lazySAT, unsatisfied clauses are selected by going through all possible groundings and adding those groundings that are unsatisfied. In contrast, our approach grounds clauses that have the potential to be unsatisfied, and does not require to go through all possible groundings. As we have restricted our case to

<p>Input: Σ_g, all grounded clauses with weights; TS, initial truth assignment; DB, the evidence given; $target$, the upper bound of cost.</p> <p>Output: TS, An interpretation that is the best solution found.</p> <ol style="list-style-type: none"> 1. $atms := \text{atoms in } \Sigma_g$ 2. repeat 3. $cost := \text{sum of weights of unsatisfied clauses}$ 4. if $cost \leq target$ 5. Return Success, TS 6. end if 7. $c := \text{a randomly chosen unsatisfied clause}$ 8. for each atom $a \in c$ and $a \notin DB$ do 9. compute $\Delta Cost(a)$, the cost incurred if a is flipped 10. end for 11. $a_f := a$ with lowest $\Delta Cost(a)$ 12. $TS := TS$ with a_f flipped 13. $cost := cost + \Delta Cost(a_f)$ 14. until the $cost$ is no more decreasing 15. Return Failure, TS
--

Figure 4.2: Modified_Weighted_MaxSAT algorithm

horn clauses (where all clauses are satisfied when all atoms are false), we can explore the search space in a more directed way. In contrast to LazySAT and walkSAT, we get an initial truth assignment at the end of pruning, which is very close to the final solution. Our approach takes less time for pruning and the walkSAT step. HornSAT is different from lazySAT, also in the sense that, we ground all relevant clauses before the maxWalkSAT step.

In the next chapter, we discuss our second related contribution, the integrated non-parametric dimensionality reduction approach.

<p>Input: Σ, the set of first order clauses with weights; DB, evidence set given; target, maximum cost expected for the optimization step if required.</p> <p>Output: TS, An interpretation when combined with DB gives the (local) optimum solution.</p> <ol style="list-style-type: none"> 1. $\{RS, TS\} := \text{Modified_}T_{\Sigma}(\Sigma, DB)$ 2. if $\{TS \cup DB\}$ is a model for Σ then 3. Return TS 4. else 5. $TS := \text{Modified_Weighted_MaxSAT}(RS, TS, DB, \text{target})$ 6. end if 7. Return TS
--

Figure 4.3: Weighted_HornSAT algorithm

Chapter 5

Optimally Extracting Discriminative Disjunctive Features for Dimensionality Reduction

In this chapter, we discuss our contribution in a related area, that is, integrated non-parametric dimensionality reduction using hierarchical kernel learning.

Many classification and regression settings have redundant and irrelevant data, which could negatively impact the efficiency of models learned. Therefore, dimensionality reduction is a relevant area of research in machine learning applications. This is also true in case of sequence labeling. In chapter 3, we have discussed our feature induction approaches, that discard irrelevant inputs and construct higher order features from relevant basic inputs. However, our main objective in that chapter was to construct higher order features. In this chapter, we extend hierarchical kernel learning in the domain of dimensionality reduction. Here, we discuss the problem and solution in general classification settings, which can be extended to sequential domains. However, since, we could not find good sequential data (large number of inputs that have the curse of dimension) to validate our approach, we limit our discussion to binary classification settings. We now give a brief introduction to dimensionality reduction approaches.

Several real world applications domains are characterized by a large set of features containing a non-trivial amount of redundant and irrelevant information. Therefore, using the entire feature space often leads to over-fitting and therefore less effective classifier models. To alleviate this problem, significant research has been invested in pre-processing approaches to reduce dimensionality of the data, either by selecting a subset of features or by projecting the features onto a smaller space. Most of these approaches suffer from the drawback that the dimensionality reduction objective and the objective for classifier training are decoupled (the two tasks are performed one after the other) and often, the

approach for dimensionality reduction is greedy. Recently, there have been some efforts to address the two tasks in a combined manner by attempting to solve an upper-bound to a single objective function (Zhu et al. 2010),(Xu 2010). However, the main drawback of these methods is that they are all parametric, in the sense that the number of reduced dimensions needs to be provided as an input to the system. Assuming that all the input features have been transformed into Boolean features, we propose an integrated non-parametric learning approach to supervised dimensionality reduction by exploring a search space of all possible disjunctions of features¹ and discovering a sparse subset of disjunctions that minimize a regularized loss function. For datasets with nominal features, it is quite natural to consider disjunctions (or sets of synonymous features) as dimensions. Here, in order to discover good disjunctive features, we employ algorithms from hierarchical kernel learning to achieve simultaneously, efficient feature selection and optimal classifier training in a maximum margin framework. We demonstrate the effectiveness of our proposed method by evaluating on bench-mark datasets.

5.1 Introduction

In building machine learning models using features, it may happen that several features might be either irrelevant or contain redundant information, which could befuddle the model learner or lead to over-fitting and consequently, a less effective model. Therefore, a small set of relevant and non-redundant features that effectively discern classes is desired. Identifying the best feature subspace for classification comes under the broad area of dimensionality reduction (DR) techniques, which can be divided into methods for feature subset selection and methods for feature extraction.

Feature subset selection (FSS) is the process of selecting a subset of features that embodies relevant and non-redundant information for use in model construction. Some approaches like Relief (Kira & Rendell 1992), FOCUS (Almuallim & Dietterich 1991) and wrapper methods (Kohavi & John 1997) select the subset of features (often greedily) based on some local relevance criteria such as information gain (Hall 1998), or chi-squared test (Jin et al. 2006), *etc.*, or some global objective such as the \mathcal{L}_1 norm in an SVM classifier. Sparse SVM (Tan et al. 2010) is one recent approach that selects a sparse subset

¹Disjunctions effectively capture the information contained in statistically synonymous basic features. Moreover, disjunctions are tractable.

of features efficiently by posing \mathcal{L}_0 norm objective as a mixed integer programming and employing a cutting plane algorithm combined with multiple kernel learning to solve a convex relaxation of the objective. Another recent feature selection approach is the one that is proposed by Zhai et al. (2012), where feature groups that have high correlation are identified from high dimensional data. Their approach performs embedded feature selection by encoding correlation measures as constraints and solves the problem using a cutting plane method. Both these approaches optimize feature selection and classification objectives simultaneously. We now discuss the background for feature extraction, which is the main focus of this chapter.

On the other hand, feature extraction approaches attempt to discover a lower-dimensional embedding of the feature space that will approximately retain the statistical relation between the instances and the class label as in the original space. Any approach for dimensionality reduction can be adjudged parametric or non-parametric respectively depending on whether the number of reduced dimensions of the embedding is considered as an input parameter or whether it is estimated within the approach. Further, each line of work can be classified as supervised, weakly-supervised or unsupervised. Unsupervised parametric methods include projective methods like Principal Component Analysis (PCA) (Jolliffe 1986a), Kernel PCA (Schölkopf et al. 1997) and its variants, manifold methods like Multi-Dimensional Scaling (Cox & Cox 2001), Laplacian Eigen-Maps (Belkin & Niyogi 2002), discriminant analysis techniques such as Linear Discriminant Analysis (Ye & Ji n.d.), Kernel Discriminant Analysis (Mika et al. 1999), Hybrid Discriminant Analysis (Yu et al. 2007) (a combination of Principal Component Analysis and Linear Discriminant Analysis which is claimed to lead to more robust models), and Continuous Latent Variable methods like Latent Semantic Indexing, Probabilistic Latent Semantic Indexing (Hofmann 1999), Latent Dirichlet Allocation (Blei, Ng & Jordan 2003).

Motivated by the use of dimensionality reduction techniques in predictive learning problems, there has been considerable amount of work on adapting these methods in supervised or weakly-supervised settings by exploiting some user provided supervision or incorporating prior background knowledge. Some of the natural extensions of unsupervised dimensionality reduction techniques in this regard are those of supervised and semi-supervised versions of PCA, multidimensional scaling, self-organizing map and laplacian Eigen-map. More specifically, supervised latent variable models include supervised

latent dirichlet allocation (sLDA) (Blei & Mcauliffe 2007), hierarchical supervised LDA (HsLDA) (Perotte et al. 2011) which extends sLDA to the case of hierarchical supervision, labeled-LDA (Ramage et al. 2009) which focuses on multi-labeled supervision for multi-labeled collections. An Empirical comparison of HsLDA against sLDA has shown that the former gives a better recall but at the price of poorer precision because of an increased number of false positives (Perotte et al. 2011). There have also been a few attempts at building a discriminative framework for supervised dimensionality reduction, mainly driven by the observation that the parameter estimates obtained in the parametric generative counterparts that employ maximum likelihood or Bayesian posterior inference do not necessarily lead to optimum models for predictive tasks, for example, discLDA (Lacoste-Julien et al. 2008) and Kernel Dimension Reduction (Fukumizu et al. 2003). Some of these methods make assumptions which may not be appropriate in reality. For example sLDA assumes a normal distribution for the response variable and further assumes it to be linearly dependent on its empirical mixture proportions. On the other hand, discLDA assumes that the mixture proportions of each class after a linear transformation should be close to each other. This assumption seems very restrictive and also appears to go directly against classification requisites.

The approach generally adopted for dimensionality reduction in nonparametric settings is to employ stochastic processes instead of distributions, which are flexible in the sense of accommodating infinite number of variables and hence being able to estimate the number of reduced dimensions implicitly. Two such stochastic variants of the unsupervised approaches are Hierarchical Dirichlet Processes (HDP) (Teh et al. 2004) and hierarchical LDA (Blei, Griffiths, Jordan & Tenenbaum 2003) that use nested Chinese Restaurant Processes. Both these approaches assume that the data has a hierarchical structure to it. Other than this, there have been some approaches to determine the size of learned ontologies, in the area of topic-modeling, by studying the change in average cosine distances between topics with respect to the increase in the number of topics. Arun et al. (2010) additionally consider information from the topic-document matrix (in contrast to HDP which takes into account only the topic-word matrix) and propose a measure for the estimation of the ‘correct’ number of topics based on Kullback-Leibler divergence of the singular value distributions of these matrices. There have been some attempts at extending these nonparametric methods to a supervised setting. For example (Xie & Pas-

sonneau n.d.) introduces different levels of supervision to an HDP, where the supervision itself dictates the number of topics and concentration of information within topics.

The above methods that treat dimensionality reduction as an isolated problem, allow for the use of any classification or regression model building on the discovered subspace. However, since dimensionality reduction and classifier training are decoupled from each other, these approaches cannot generally guarantee optimality of feature selection with respect to the classifier objective. There have been some attempts (Li et al. 2003) to develop models that integrate dimensionality reduction with model building, and which have shown the ability to discover predictive topic representations that are more suitable for supervised prediction tasks. Maximum Entropy Discrimination Latent Dirichlet Allocation (medLDA) (Zhu et al. 2010) is a maximum margin variant of maximum-entropy discrimination LDA which integrates the maximum margin criterion with LDA by optimizing a single objective function with a set of expected margin constraints. A more recent approach, Multi-Modal Probabilistic Latent Semantic Analysis (MMpLSA) (Xu 2010), that evolved along the same lines, has integrated PLSA (in place of its Bayesian version LDA) with the maximum margin criterion and has shown to perform better than its predecessors (Blei & Mcauliffe 2007, Lacoste-Julien et al. 2008, Zhu et al. 2010). Further, these studies have shown that building another classifier using the induced dimensions does not introduce much performance gain. The main limitation of these methods is that they are parametric – they need the number of reduced dimensions as input and have no intrinsic mechanism for estimating this number within the system.

There has not been much discussion on the optimality of the models built subsequently from the reduced set of features produced by dimensionality reduction techniques. Chechik (2008) solves the maximum margin objective in the dual but does not guarantee optimality, while medLDA (Zhu et al. 2010) and MMpLSA (Xu 2010) have both solved a tight bound approximation of the original objective in the interest of tractability.

5.1.1 Our Contribution

Our work falls in the league of integrated maximum margin linear dimensionality reduction approaches for model building in a supervised classification² setting. For simplicity,

²While our approach can be very naturally extended to the regression setting by changing the loss function, we have not empirically studied supervised dimensionality reduction for regression in this work.

we assume that all our basic features (attributes themselves) are Boolean. For example, the presence or absence of a word in the dictionary can be a basic Boolean feature. Nevertheless, our approach can be applied to settings with nominal features by constructing one Boolean feature for each value of the nominal feature. Similarly, a numeric attribute can be discretized into different intervals and Boolean features be constructed for each of these intervals.

We intend to learn fewer features that capture the redundant information present in basic features by grouping and representing them as a single disjunctive feature. For instance in text classification, disjunction of synonymous words can be treated as a single feature. The disjunctive feature is relevant to the classification problem at hand if and only if any one or more of the basic features that share same meaning are relevant. For example, words such as *beautiful* and *gorgeous* could convey the same sense about an entity and therefore a single feature that is a disjunction of these will be relevant if, and only if, any one or both of them are relevant. The preferred disjunctive features should be maximum in the sense that we try to include as many synonymous basic features as possible and exclude any non-synonymous basic feature in a disjunction. For example, *ugly* is not a synonym of *beautiful* and therefore we would generally not expect *ugly* and *beautiful* to co-occur in the same disjunctive feature. Our objective is to construct an optimal set of relevant and non-redundant features for classification, with hinge loss as the objective. As noted above, any approach with the dimensionality reduction approach decoupled from the classifier training has limitations in finding optimum models. In this thesis, we propose an integrated supervised approach for dimensionality reduction in a maximum margin framework.

To the best of our knowledge, there has not been any approach in discriminative learning that integrates non-parametric dimensionality reduction with optimal model building for classification. There has been some work on employing maximum margin based nonparametric dimensionality reduction in a multi-task setting (Argyriou et al. 2007) and in the area of learning underlying shared structures amongst classes (Amit et al. 2007) in a multi-class setting. Although these methods solve their objectives optimally, they are not directly comparable to our work, since in the case of binary classification or 1-task case, their dimensionality reduction approach reduces to a trivial feature selection using 1-norm regularization. Our main contribution in this thesis is an integrated

optimal and efficient classifier learning and dimensionality reduction technique based on the Hierarchical Kernel Learning (HKL) setting (Bach 2009). Although we discuss our approach in the context of general SVMs for binary classification, our approach can be trivially extended to other variants. We conclude this section by briefly introducing the hierarchical kernel learning framework.

Hierarchical kernel learning (HKL) (Bach 2009) approaches have gained interest recently due to their ability to learn kernels in a large kernel space. Bach (2009) has introduced HKL framework that efficiently explores an exponential kernel space where individual kernels can be decomposed into base kernels (Bach 2009). Their approach selects kernels from the space of all possible kernels embedded in a directed acyclic graph using a graph based sparsity inducing norm. The complexity of HKL is polynomial in the number of selected kernels. The regularizer used discourages complex kernels and, thereby, helps to learn a small set of simple kernels. In this thesis, we leverage the HKL framework to simultaneously perform dimensionality reduction and classifier training. We prune irrelevant features and group redundant features in the form of disjunctions, which also adds a logical significance to it. The sparsity inducing hierarchical regularizer used in HKL selects a sparse set of non-synonymous disjunctions. From our experiments on standard datasets, we observe that the disjunctions discovered by our method are more refined than the topics identified by competitor methods. In the following paragraph, we discuss the proposed approach and algorithm for learning disjunctions for dimensionality reduction in a hierarchical kernel learning setting.

5.2 Optimal Non-Parametric Max Margin Dimensionality Reduction

We now formally define our problem of simultaneously performing dimensionality reduction and classifier training and present an efficient polynomial time algorithm to solve the objective optimally. We consider features that do not discern classes as irrelevant and therefore can be discarded. For example, in sentiment classification, a set of similar-meaning words such as *method*, *algorithm*, and others might not help in discerning classes and can be omitted. On the other hand, multiple features capturing the same information (synonyms) are redundant and might result in an ineffective classifier. For example, words

such as *beautiful*, *exquisite*, *gorgeous*, *charming*, and the like, capture similar information about the entity being discussed and should probably be clubbed together in the same dimension.

For effectively capturing the meaning of a group of synonymous basic features without redundancy, we explore the space of disjunctive features that are disjunctions (\vee) of basic features. For instance, in document classification, synonymous words *beautiful*, *exquisite*, and *gorgeous* can be used to construct a disjunctive feature and the feature is instantiated when any one or more of the component features are active. We refer to such features as DisjunctProjs (**Disjunctive Projections**). The space of all possible DisjunctProjs can be visualized as a lattice, with a structure similar to the subset lattice, where the top node is the empty node, the nodes at the next level are the individual basic features and so on. The bottom node in the lattice is the disjunction of all the basic features. Upward and downward refinements of a node can be defined in terms of deletion or addition of a basic feature from or to the node respectively.

We aim at automatically selecting good maximal DisjunctProjs from the ordering. A *good* DisjunctProj is a disjunction which does not contain any statistically different feature. A maximal DisjunctProj is a disjunction of the maximum number of basic features capturing (statistically) similar information about the classes being discriminated against each other. Therefore, a good and maximal DisjunctProj corresponds to a disjunction of synonymous basic features in which no more basic features can be added without affecting its meaning. With this understanding, if a DisjunctProj is not effective for classification, we assume that the feature will not become more effective by the addition of a new basic feature to the disjunction. For example, if *beautiful* \vee *ugly* is not good, then *beautiful* \vee *ugly* \vee *gorgeous* may not be good in general. Therefore, in the ordering, if a node is not selected, we expect that none of its descendants be selected either. Now let us assume that we have a good DisjunctProj in the form of *beautiful* \vee *exquisite* \vee *gorgeous*. This is maximal if adding a new word to the disjunction results in a bad DisjunctProj for classification. Therefore, if *ugly* is added, in the new DisjunctProj formed by this addition, *ugly* can be considered as noise. Next, we formally define our problem.

5.2.1 Formal Specification of the Problem

We pose our requirement as a maximum margin optimization problem which is expected to select a sparse set of good DisjunctProjs from the ordering and learn their optimal feature weights simultaneously. Let each element of vector $\boldsymbol{\psi}$ corresponds to a node in the disjunction lattice and \mathbf{f} the corresponding weights. Let \mathcal{V} be the set of indices to the nodes in the lattice. A node $\psi_v(\cdot)$ in the ordering is a disjunction of a set of basic features and can be represented as $\bigvee_{\hat{v} \in v} \psi_{\hat{v}}(\cdot)$, where \hat{v} stands for a basic feature present in v .

To select a sparse set of DisjunctProjs from the exponential feature space, we employ a hierarchical regularizer on the exponential feature space and present the SVM formulation for binary classification as,

$$\min_{\mathbf{f}, b, \boldsymbol{\xi}} \frac{1}{2} \left(\sum_{v \in \mathcal{V}} d_v \|\mathbf{f}_{D(v)}\|_{\rho} \right)^2 + C \mathbf{1}^T \boldsymbol{\xi} \quad (5.1)$$

$$s.t. \forall i: y_i \left(\sum_{v \in \mathcal{V}} \langle f_v, \psi_v(\mathbf{x}_i) \rangle - b \right) \geq 1 - \xi_i, \boldsymbol{\xi} \geq 0$$

where $\mathbf{f}_{D(v)}$ is the vector of feature weights corresponding to the elements in the descendant nodes $D(v)$ of node v including the node v itself, $\rho \in (1, 2]$, d_v is the prior parameter that can be interpreted as the usefulness of node v , C is the regularization parameter, ξ_i is the slackness in the margin for i^{th} example, \mathbf{x}_i is the input vector of dimension N (where N is the number of basic features) corresponding to the i^{th} example, $y_i \in \{0, 1\}$ is the predicted output value of the i^{th} example, b is the bias term, f_v is the feature weight corresponding to v^{th} node and $\psi_v(\mathbf{x}_i)$ is the truth value of v^{th} node for the i^{th} training example. To discourage very large and potentially ineffective DisjunctProjs, we define d_v as $\beta^{|v|-k}$, where β and k are some parameterized constants and $|v|$ is the size of the node v . Many of $\|\mathbf{f}_{D(v)}\|_{\rho}$ are expected to be zero due to the 1-norm which will force $f_u, \forall u \in D(v)$, to reduce to zeros. This effectively discourages selection of large number DisjunctProjs. Additionally, as illustrated by Szafranski & Rakotomamonjy (2008), ρ -norm, where $\rho \in (1, 2]$, induces further sparsity among the nodes. Therefore, for $\|\mathbf{f}_{D(v)}\|_{\rho}$ that are not reduced to zero by 1-norm, the ρ -norm forces many of the descendants of node v to zero and thus ensures a sparse solution.

The kernel for a node v can be defined as, $\mathbf{K}_v(\mathbf{x}_i, \mathbf{x}_j) = \langle \psi_v(\mathbf{x}_i), \psi_v(\mathbf{x}_j) \rangle = (1 - \prod_{\hat{v} \in v} \overline{\psi_{\hat{v}}(\mathbf{x}_i)})(1 - \prod_{\hat{v} \in v} \overline{\psi_{\hat{v}}(\mathbf{x}_j)})$ This enables the sum of the kernels over a sub-lattice V

to be computed efficiently. For instance, the sum of kernels over the entire lattice is, $\sum_{v \in \mathcal{V}} \mathbf{K}_v(\mathbf{x}_i, \mathbf{x}_j) = 2^N + \prod_{l=1}^N (1 + \overline{\psi}_l(\mathbf{x}_i) \overline{\psi}_l(\mathbf{x}_j)) - \prod_{l=1}^N (1 + \overline{\psi}_l(\mathbf{x}_i)) - \prod_{l=1}^N (1 + \overline{\psi}_l(\mathbf{x}_j))$. This is consistent with the requirement of polynomial time summability of descendant kernels in HKL (Bach 2009) and thus the active set algorithm can be employed to iteratively select a sparse set of features, since the optimality condition check (which has been discussed afterwards) in it culminates into a more efficient computation with the exponential number of summations being reduced to polynomial number of products.

We now discuss the solution to the problem defined in equation (5.1). The solution to equation (5.1) is expected to yield a sparse set of features with non-zero weights. Therefore, as illustrated in (Bach 2009), the solution to equation (5.1) when solved with the entire set of features is the same when solved with the optimum set of features. As the latter has lesser computational complexity, an active set algorithm, which starts with a small subset of DisjunctProjs and iteratively adds nodes that violate a sufficiency condition, can be employed. The primal optimization problem with an active set of features \mathcal{W} (restricted primal) can be represented as

$$\min_{\mathbf{f}, b, \boldsymbol{\xi}} \frac{1}{2} \left(\sum_{v \in \mathcal{W}} d_v \parallel \mathbf{f}_{D(v) \cap \mathcal{W}} \parallel_{\rho} \right)^2 + C \mathbf{1}^{\top} \boldsymbol{\xi} \quad (5.2)$$

$$s.t. \forall i: y_i \left(\sum_{v \in \mathcal{W}} \langle f_v, \psi_v(\mathbf{x}_i) \rangle - b \right) \geq 1 - \xi_i, \quad \boldsymbol{\xi} \geq 0$$

To solve efficiently, we use variational characterization proposed in lemma 26 of (Micchelli & Pontil 2005) and reduce the regularizer term of equation (5.1) as,

$$\left(\sum_{v \in V} \parallel \mathbf{f}_{D(v)} \parallel_{\rho} \right)^2 = \min_{\boldsymbol{\gamma} \in \Delta_{|V|, 1}} \sum_{v \in V} \frac{d_v^2 \parallel \mathbf{f}_{D(v)} \parallel_{\rho}^2}{\gamma_v}.$$

where $\Delta_{d,r} = \{\boldsymbol{\lambda} \in \mathbb{R}^d, \sum_{j \in \mathbb{N}_d} \lambda_j^r = 1, \lambda_j \geq 0\}$.

Further by appying the same lemma again we have

$$\parallel \mathbf{f}_{D(v)} \parallel_{\rho}^2 = \min_{\boldsymbol{\lambda}_v \in \Delta_{|D(v)|, \hat{\rho}}} \sum_{u \in D(v)} \frac{\parallel \mathbf{f}_u \parallel_2^2}{\lambda_{vu}}$$

where $\hat{\rho} = \frac{\rho}{2-\rho}$.

By appylyng variational characterization, we can represent equation (1) as

$$\begin{aligned} \min_{\gamma \in \Delta_{|\mathcal{V}|,1}} \min_{\lambda_v \in \Delta_{|D(v)|,\bar{\rho}} \forall v \in \mathcal{V}} \min_{\mathbf{f}, b, \boldsymbol{\xi}} \sum_{u \in \mathcal{V}} \delta_u^{-1}(\gamma, \boldsymbol{\lambda}) \|f_u\|_2^2 + C \mathbf{1}^\top \boldsymbol{\xi} \\ \text{s.t. } \forall i : y_i \left(\sum_{v \in \mathcal{V}} \langle f_v, \psi_v(\mathbf{x}_i) \rangle - b \right) \geq 1 - \xi_i, \boldsymbol{\xi} \geq 0 \end{aligned}$$

where $\delta_u^{-1}(\gamma, \boldsymbol{\lambda}) = \sum_{v \in A(u)} \frac{\delta_v^2}{\gamma_v \lambda_{vu}}$, $A(u)$ denotes ancestors of u which includes the node u itself. By applying the representer theorem (Rakotomamonjy et al. 2008) on the variational characterization of the regularizer term, we can derive the partial dual of the above primal form with respect to $\mathbf{f}, b, \boldsymbol{\xi}$ alone as,

$$\min_{\gamma \in \Delta_{|\mathcal{V}|,1}} \min_{\lambda_v \in \Delta_{|D(v)|,\bar{\rho}} \forall v \in \mathcal{V}} \max_{\boldsymbol{\alpha} \in \boldsymbol{\tau}(y, C)} G(\gamma, \boldsymbol{\lambda}, \boldsymbol{\alpha})$$

where

$$G(\gamma, \boldsymbol{\lambda}, \boldsymbol{\alpha}) = \mathbf{1}^\top \boldsymbol{\alpha} - \frac{1}{2} \boldsymbol{\alpha}^\top \left(\sum_{u \in \mathcal{V}} \delta_u(\gamma, \boldsymbol{\lambda}) \mathbf{K}_u \right) \boldsymbol{\alpha}$$

and $\boldsymbol{\tau}(y, C) = \{\boldsymbol{\alpha} \in \mathbb{R}^m \mid 0 \leq \boldsymbol{\alpha} \leq C, \sum_{i=1}^m y_i \alpha_i = 0\}$

The final dual of equation (5.1) can be derived as,

$$\min_{\boldsymbol{\eta} \in \Delta_{|\mathcal{V}|,1}} g(\boldsymbol{\eta}) \tag{5.3}$$

where

$$g(\boldsymbol{\eta}) = \max_{\boldsymbol{\alpha} \in \boldsymbol{\tau}(y, C)} \mathbf{1}^\top \boldsymbol{\alpha} - \frac{1}{2} \left(\sum_{v \in \mathcal{V}} \zeta_v(\boldsymbol{\eta}) (\boldsymbol{\alpha}^\top K_v \boldsymbol{\alpha})^{\bar{\rho}} \right)^{\frac{1}{\bar{\rho}}}$$

where $\zeta_v(\boldsymbol{\eta}) = \left(\sum_{u \in A(v)} d_u^\rho \eta_u^{1-\rho} \right)^{\frac{1}{1-\rho}}$ and $\bar{\rho} = \frac{\rho}{2(\rho-1)}$.

The solution to the final dual, with \mathcal{V} restricted to the active set \mathcal{W} gives the solution to the restricted primal problem.

To solve the problem efficiently, we employ an active set algorithm. The active set algorithm (Bach 2009) starts with an initial set of features and at every iteration, solves the dual problem with the current active features, checks a sufficiency condition on the nodes that are sources of the complement set of current active set ($sources(\mathcal{W}^c) = \{w \in \mathcal{W}^c \mid A(w) \cap \mathcal{W}^c = \{w\}\}$, where \mathcal{W}^c is the complement of \mathcal{W} in \mathcal{V}), $A(w)$ is the set of ancestor nodes of w and adds the violating nodes to the active set. The process is continued until no new node violates the sufficiency condition. A mirror descent algorithm (Ben-Tal

& Nemirovskiaei 2001) is employed to solve the dual. The active set algorithm adapted from (Bach 2009) is outlined in figure (5.1). We now derive the sufficiency condition that determines whether a given active set of features yield an optimal model.

The sufficiency condition for the solution to the primal, is essentially obtained by restricting the duality gap by a threshold ϵ and is specified below. The duality gap is given by

$$\begin{aligned}
& \max_{\boldsymbol{\alpha} \in \boldsymbol{\tau}(y, C)} \min_{\boldsymbol{\gamma} \in \Delta_{|\mathcal{V}|, 1}} \min_{\boldsymbol{\lambda}_v \in \Delta_{|D(v)|, \hat{\rho}} \forall v \in \mathcal{V}} G(\boldsymbol{\gamma}, \boldsymbol{\lambda}, \boldsymbol{\alpha}) - \min_{\boldsymbol{\gamma} \in \Delta_{|\mathcal{V}|, 1}} \min_{\boldsymbol{\lambda}_v \in \Delta_{|D(v)|, \hat{\rho}} \forall v \in \mathcal{V}} \max_{\boldsymbol{\alpha} \in \boldsymbol{\tau}(y, C)} G(\boldsymbol{\gamma}, \boldsymbol{\lambda}, \boldsymbol{\alpha}) \\
& \leq \frac{1}{2} \left(\sum_{v \in V} \|\mathbf{f}_{D(v)}\|_{\rho}^2 + C \mathbf{1}^{\top} \boldsymbol{\xi} - \min_{\hat{\boldsymbol{\gamma}} \in \Delta_{|\mathcal{V}|, 1}} \min_{\hat{\boldsymbol{\lambda}}_v \in \Delta_{|D(v)|, \hat{\rho}} \forall v \in \mathcal{V}} G(\hat{\boldsymbol{\gamma}}, \hat{\boldsymbol{\lambda}}, \boldsymbol{\alpha}) \right) \\
& = \sum_{v \in V} \|\mathbf{f}_{D(v)}\|_{\rho}^2 + C \mathbf{1}^{\top} \boldsymbol{\xi} - \mathbf{1}^{\top} \boldsymbol{\alpha} + \frac{1}{2} \left(\sum_{v \in V} \|\mathbf{f}_{D(v)}\|_{\rho}^2 \right. \\
& \quad \left. - \max_{\hat{\boldsymbol{\gamma}} \in \Delta_{|\mathcal{V}|, 1}} \max_{\hat{\boldsymbol{\lambda}}_v \in \Delta_{|D(v)|, \hat{\rho}} \forall v \in \mathcal{V}} \sum_{u \in V} \delta_u(\boldsymbol{\gamma}, \boldsymbol{\lambda}) \boldsymbol{\alpha}^{\top} \mathbf{K}_u \boldsymbol{\alpha} \right)
\end{aligned}$$

Taking the Lagrange dual and applying the Lemma 26 of (Micchelli & Pontil 2005) we derive the final form of the sufficiency condition from the upper bound of the duality gap as,

$$\max_{u \in \text{sources}(\mathcal{W}^c)} \boldsymbol{\alpha}_{\mathcal{W}}^{\top} R(u) \boldsymbol{\alpha}_{\mathcal{W}} \leq \left(\sum_{v \in \mathcal{W}} \delta_v \|\mathbf{f}_{D(v)}\|_{\rho} \right)^2 + \epsilon$$

The $(i, j)^{th}$ component of $R(u)$ can be simplified to

$$\begin{aligned}
R(u)_{ij} = & \frac{\left(\beta^{\frac{2k}{N}} \left(1 + \frac{1}{(1+\beta)^2} \right) \right)^N}{\left(\beta^2 \left(1 + \frac{1}{(1+\beta)^2} \right) \right)^{|u|}} - \prod_{\hat{u} \in u} \frac{\frac{\overline{\psi}_{\hat{u}}(\mathbf{x}_i)}{\beta^2}}{\left(1 + \frac{\overline{\psi}_{\hat{u}}(\mathbf{x}_i)}{(1+\beta)^2} \right)} \prod_{l=1}^N \left(\beta^{\frac{2k}{N}} \left(1 + \frac{\overline{\psi}_l(\mathbf{x}_i)}{(1+\beta)^2} \right) \right) \\
& - \prod_{\hat{u} \in u} \frac{\frac{\overline{\psi}_{\hat{u}}(\mathbf{x}_j)}{\beta^2}}{\left(1 + \frac{\overline{\psi}_{\hat{u}}(\mathbf{x}_j)}{(1+\beta)^2} \right)} \prod_{l=1}^N \left(\beta^{\frac{2k}{N}} \left(1 + \frac{\overline{\psi}_l(\mathbf{x}_j)}{(1+\beta)^2} \right) \right) \\
& + \prod_{\hat{u} \in u} \frac{\frac{\overline{\psi}_{\hat{u}}(\mathbf{x}_i) \overline{\psi}_{\hat{u}}(\mathbf{x}_j)}{\beta^2}}{\left(1 + \frac{\overline{\psi}_{\hat{u}}(\mathbf{x}_i) \overline{\psi}_{\hat{u}}(\mathbf{x}_j)}{(1+\beta)^2} \right)} \prod_{l=1}^N \left(\beta^{\frac{2k}{N}} \left(1 + \frac{\overline{\psi}_l(\mathbf{x}_i) \overline{\psi}_l(\mathbf{x}_j)}{(1+\beta)^2} \right) \right)
\end{aligned}$$

At each iteration, the dual problem (5.3) is solved with the current set of active features, the sufficiency condition (5.4) is checked and the violating nodes are added to the active

<p>Input: Training data D, Maximum tolerance ϵ</p> <ol style="list-style-type: none"> 1. Initialize Active set $\mathcal{W} = \text{Top node}$ in the lattice 2. Compute $\boldsymbol{\eta}, \boldsymbol{\alpha}$ by solving (5.3) 3. while sufficiency condition is not satisfied, do 4. Add nodes from the set $\text{sources}(\mathcal{W})$ violating sufficiency condition to \mathcal{W} 5. Recompute $\boldsymbol{\eta}, \boldsymbol{\alpha}$ by solving (5.3) 6. end while 7. Output: active-set $\mathcal{W}, \boldsymbol{\eta}, \boldsymbol{\alpha}$
--

Figure 5.1: Active set algorithm

set. The mirror descent algorithm employed to solve (5.3) is as follows. For a given $\boldsymbol{\eta}$, let $\bar{\boldsymbol{\alpha}}$ be the solution to (5.4), then the v^{th} sub-gradient of $g(\boldsymbol{\eta})$ can be obtained from,

$$(\nabla g(\boldsymbol{\eta}))_v = -\frac{d_v^\rho \eta_v^{-\rho}}{2\bar{\rho}} \left(\sum_{u \in \mathcal{V}} \zeta_u(\boldsymbol{\eta}) (\bar{\boldsymbol{\alpha}}^\top \boldsymbol{\kappa}_u \bar{\boldsymbol{\alpha}})^{\bar{\rho}} \right)^{\frac{1}{\bar{\rho}}-1} \left(\sum_{u \in D(v)} \zeta_u(\boldsymbol{\eta})^\rho (\bar{\boldsymbol{\alpha}}^\top \boldsymbol{\kappa}_u \bar{\boldsymbol{\alpha}})^{\bar{\rho}} \right)$$

The updated $\boldsymbol{\eta}$ is then used to solve (5.4) by using the sequential minimal optimization (SMO) algorithm. Active set iterations are continued until no nodes violate the sufficiency condition.

The active set algorithm thus returns a sparse set of DisjunctProjs and their optimal weights. For settings where some background knowledge is available, the feature space can be explored more efficiently. We discuss this next.

Incorporating Background Knowledge: In domains such as sentiment classification, where some background knowledge about the features is available, it is possible to control the number of nodes to be considered for inclusion in each iteration of the active set algorithm and thus speed up the learning process. For instance, in sentiment classification, the background information is often derived from user-preferences on terms by modifying the Dirichlet prior or by adding some prior knowledge of word sentiments as available in SentiWordNet, HowNet and others (He 2011, Li et al. 2010, Lin & He 2009). This helps to explore lexical properties of words. For example, words in a synonymous wordset are likely to possess similar polarities.

Since words representing similar meanings are likely to have the same part-of-speech,

we can restrict our space of DisjunctProjs to disjunctions of words belonging to the same part-of-speech. This reduces the number of potential features to explore which in turn effectively speeds up the learning process. Prior information about the polarity of words can be embedded in the prior parameter d_v of node v , in our integrated dimensionality reduction approach. This effectively helps the system to select features that are strongly related to the classification task.

To incorporate sentiment prior of node v , d_v has been heuristically computed as the product of the absolute sentiment score of the individual words in the disjunction corresponding to that node. i.e. $d_v = \prod_{\hat{v} \in v} SS(\hat{v})$ where $SS(\hat{v}) \in [0, 1]$ is the absolute sentiment score of the word \hat{v} , measured as $|\text{positive sentiment score} - \text{negative sentiment score}|$, where the sentiment scores are obtained from the SentiWordNet. The parameter d_v encourages selection of the more strongly polar features over the weaker ones. On this account, the $R(u)_{ij}$ term in the sufficiency condition for optimality is modified as

$$R(u)_{ij} = \frac{1}{\prod_{\hat{u} \in u} \left(\frac{SS(\hat{u})}{1 + SS(\hat{u})} \right)} \left(\frac{\prod_{l=1}^N \left(1 + \frac{1}{(1 + SS(l))^2} \right)}{\prod_{\hat{u} \in u} \left(1 + (1 + SS(\hat{u}))^2 \right)} - \frac{\prod_{l=1}^N \left(1 + \frac{\bar{\psi}_l(x_i)}{(1 + SS(l))^2} \right)}{\prod_{\hat{u} \in u} \left(1 + \frac{(1 + SS(\hat{u}))^2}{\bar{\psi}_{\hat{u}}(x_i)} \right)} \right. \\ \left. - \frac{\prod_{l=1}^N \left(1 + \frac{\bar{\psi}_l(x_j)}{(1 + SS(l))^2} \right)}{\prod_{\hat{u} \in u} \left(1 + \frac{(1 + SS(\hat{u}))^2}{\bar{\psi}_{\hat{u}}(x_j)} \right)} + \frac{\prod_{l=1}^N \left(1 + \frac{\bar{\psi}_l(x_i)\bar{\psi}_l(x_j)}{(1 + SS(l))^2} \right)}{\prod_{\hat{u} \in u} \left(1 + \frac{(1 + SS(\hat{u}))^2}{\bar{\psi}_{\hat{u}}(x_i)\bar{\psi}_{\hat{u}}(x_j)} \right)} \right)$$

We leave the complete theory and experiments on sentiment datasets as future work and move on to discuss our experiments and results in the next section.

Chapter 6

Experiments and Results

In Chapter 5, we discussed our main contribution in the form of efficient methods for feature induction for sequence labeling problems. In Chapter 4 and 5, we discussed our contribution in two related areas, that is, the fast and memory efficient satisfiability checking for inference in first order logic systems and the integrated non-parametric dimensionality reduction approach, respectively. We have implemented all our proposed approaches in java and have evaluated the approaches on standard datasets. In this chapter, we discuss our experiments and results for each of our contributions. We performed all our feature induction experiments and dimensionality reduction experiments on a 12-core (2.66 GHz) 64 bit AMD machine with 8 GB RAM and running Ubuntu 11.04. Our satisfiability experiments are performed on a dual-core (2.66 GHz) 64 bit AMD machine with 3 GB RAM and running Ubuntu 8.04. We start with the discussion for experiments with feature induction approaches.

6.1 Learning Discriminative Features for Sequence Labeling

In chapter 3, we categorized first order definite features and identified the class of Simple Conjuncts (\mathcal{SC}) and Composite Features (\mathcal{CF}) as important categories for sequence labeling. We have presented a greedy feature induction approach for learning \mathcal{SC} s. A learning approach using Hierarchical Kernel Learning for learning optimal \mathcal{SC} s is also presented. We then presented two strategies to learn optimal \mathcal{CF} s. Here we discuss the experimental results. We evaluate our feature induction approaches on activity recognition data. This is mainly because activity recognition problems have sparse, skewed and noisy data and thus learning is challenging. Our approach can be easily applied to other non sparse, non skewed and non noisy data as well. In all the sequence labeling experiments, we use

Viterbi algorithm for inference, unless stated otherwise. We first discuss experimental results of our greedy approach on activity recognition data in the next subsection.

6.1.1 Greedy Feature Induction for Sequence Labeling

For greedy approach, we have carried out experiments on the activity recognition data set made available by van Kasteren et al. (2008) of the University of Amsterdam. The dataset consists of binary values reported at each time interval by 14 sensors installed at various locations in a house. Activities are daily house hold activities like *sleeping*, *usingToilet*, *preparingDinner*, *preparingBreakfast*, *leavingOut*, and the like. There are eight activities annotated for 28 days. The data is marked for each one minute time slot and there are 40006 instances. Since the authors of the dataset are from the University of Amsterdam, we will refer to the dataset as the UA data. In the dataset, some activities occurred more frequently than others and some activities occurred for a longer duration, and hence there are distribution errors.

As discussed in the previous chapters, we proposed mapping labels and group of inputs/observations to improve the performance of sequence labeling approaches. Our initial experiments with a standard Inductive Logic Programming (ILP) system Aleph (Srinivasan 2007) did not learn useful rules. This is basically because these approaches tend to maximize logical coverage, whereas our objective is probabilistic coverage (probability by which examples are covered). The scoring function used in many ILP systems is a function of positive (*pos*) and negative (*neg*) examples covered. Moreover the examples covered in each step are removed. In typical activity recognition datasets that tend to be sparse, skewed and noisy, probabilistic coverage is the objective. Hence, we experiment with our greedy approach that employs HMM evaluation as scoring function. In a typical activity recognition setting, sensor observations are sparse and therefore, the systems that do not consider the temporal dependencies between activities fail to give comparable results, as observed in our experiments. For example, activities like sleeping may cause a sensor at the bedroom door to fire only at the start and at the end of the sleeping period. It is intuitive to think that a person most likely will be sleeping at a particular time step if he was sleeping at the previous time step. This justifies the use of HMM evaluation as score, since HMMs capture the transition dependencies along with the observation dependencies.

We assume the data is complete in our case. We constructed 28 sequences of equal length (1428 time instances) from the whole data and performed our experiments in a leave one day out manner in a 28 fold cross validation set-up. We report both micro-average and macro-average prediction accuracies. The micro-average accuracy is referred to as time-slice accuracy in (van Kasteren et al. 2008), and is the average of per-class accuracies, weighted by the number of instances of the class. Macro-average accuracy, referred to as class accuracy in (van Kasteren et al. 2008), is simply the average of the per-class accuracies. Micro-averaged accuracy is typically used as the performance evaluation measure. However in data that is biased towards some classes, macro-average also is an indicator of the quality of the model.

We ran four experiments on the data. The First experiment is the traditional/standard HMM¹ as suggested in (van Kasteren et al. 2008). The second experiment, B&BHMM, uses Aleph to learn emission rules in the form of definite clauses for each activity. These rules along with the data are passed to a customized implementation of HMM for probabilistic learning and inference. The third and fourth experiments are the proposed greedy Feature Induction assisted HMM (Greedy FIHMM) which inductively learns HMM emission model using HMM evaluation as the score. The emission model is combined with the n^2 inter state transitions and the probabilities are learned to obtain the complete HMM model. The third experiment optimizes macro-average accuracy (Greedy FIHMM (macro-average)) while the fourth experiment optimizes micro-average accuracy (Greedy FIHMM (micro-average)). The results are shown in tables 6.1. The performance comparison is also illustrated in Figure 6.1.

From the results, it can be noted that the B&BHMM gave a worse macro-average accuracy than traditional HMM while giving comparable micro-average accuracy. This is due to the inappropriate evaluation function used by the branch & bound structure learning systems while doing refinement of learned clauses. The proposed feature induction assisted HMM model construction with micro-average accuracy as the scoring function performed better than all other approaches both in micro-average and macro-average accuracies. Our statistical significance test for micro-average using Wilcoxon Signed Rank Test (Siegel & Castellan 1988) indicates a 0.00012 level of significance over traditional HMM and 0.00156 level of significance over B&BHMM. As our objective is

¹Use the assumption of conditional independence when there are multiple inputs at a sequence position.

Table 6.1: Micro average accuracy and macro average accuracy of classification in percentage using standard HMM, B&B learning assisted HMM and greedy feature induction assisted HMM (macro-average and micro-average accuracies as scoring function separately) on UA dataset with 28 fold cross validation.

	Micro avg.	Macro avg.
Std. HMM	55.41	45.62
B&B HMM	56.94	27.81
Greedy FIHMM (macro-average)	54.98	55.11
Greedy FIHMM (micro-average)	71.59	55.13

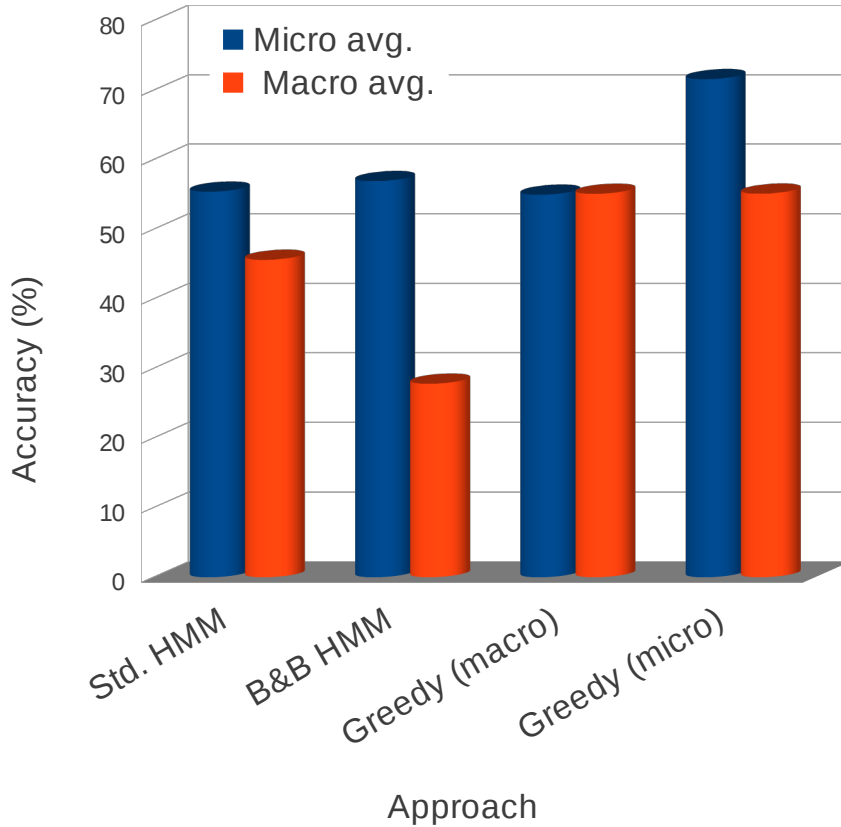


Figure 6.1: Performance comparison of standard HMM, B&B learning assisted HMM and greedy feature induction assisted HMM with scoring function defined by macro-average accuracy (Greedy (macro)) and micro-average accuracy (Greedy (micro)) on UA dataset.

to improve micro-average accuracy of labeling and as it is evident from the results that macro-average as score did not show improvement over the approach with micro-average

accuracy as score, we use micro-average accuracy as the score in further experiments.

In contrast to standard HMM, which has training and evaluation times in the order of seconds, the learning part of feature induction approaches take an average of three hours to train a model. The inference is faster and converges in fraction of a second. Since the learning is done once and inference, more often, this relatively long learning time is not considered as affecting system performance. Moreover, the relatively longer training time can be justified by the accuracy gain and fast inference. The B&BHMM takes one hour for training and a few seconds for evaluation. As explained in previous chapters, the greedy approaches cannot guarantee optimum models. We therefore proposed and developed an optimal feature learning approach using hierarchical kernel learning.

6.1.2 Optimal Feature Induction using Hierarchical Kernels for Sequence Labeling

In this subsection, we discuss the results of our experiments with Hierarchical Kernel Learning on Structured Output Spaces for learning simple conjuncts (SC) for sequence labeling. Our experiments are carried out on two publicly available activity recognition datasets. The first is the UA data provided by van Kasteren et al. (2008) of University of Amsterdam (UA data), which we explained in subsection 6.1.1. The second data is recorded at MIT Place-Lab by Tapia (2003, 2004) (we call the dataset PlaceLab data). The data is extracted from the apartments of two single-persons' (subject one and subject two). The apartments are fitted with 76 and 70 sensors for subject one and two, respectively; data is collected for two weeks (20160 instances). Annotated activities are categorized into nine high level activities such as *employmentRelated*, *personalNeeds*, *domesticWork*, *educational*, *entertainment*, and the like.

In this experiment, we split the whole UA data into 40 sequences of length 1000 each and each sequence is treated as an example. Similarly we split PlaceLab data, which has 20160 instances, into 14 sequences of length 1440. We used 25% of example sequences for training and the rest for testing. There are 10 sequences in the training set for UA data and 4 sequences in the training set for PlaceLab data. We report all accuracies by averaging across the four folds. We report both micro-average and macro-average prediction accuracies. In the following paragraphs, we compare our approach with other approaches that gave comparable results or better.

For UA data, we compare our results with eight other approaches: (a) standard HMM (Rabiner 1990), (b) Branch and Bound structure learning assisted HMM model construction (B&BHMM), where the rules learned by Aleph (Srinivasan 2007) (an ILP system which learns definite rules from examples) for each activity determine the HMM emission structure, (c) greedy feature induction assisted HMM approach (Greedy FIHMM) (Nair, Ramakrishnan & Krishnaswamy 2011), (d) StructSVM approach (Tsochantaridis et al. 2004), (e) Conditional Random Field (CRF) (Lafferty, McCallum & Pereira 2001), (f) Conditional Random Field with Feature Induction (FICRF) (McCallum 2003, 2002), (g) RELHKL (without considering transitions) (Jawanpuria et al. 2011) and (h) RELHKL + StructSVM. While standard approaches such as HMM, CRF and structSVM use basic features (binary sensor values) as emission features, feature induction approaches such as Greedy FIHMM and FICRF use conjunctions of basic features as emission features. In contrast to greedy feature induction approaches, RELHKL, and StructHKL find the feature conjunctions efficiently and optimally. While RELHKL without the transition features does not consider the structure in output space, RELHKL + StructSVM solves the problem in two steps. In the first step, RELHKL (without considering transitions) is employed to learn rules for each label. In the second step, the rules learned in the first step are fed as features into the StructSVM algorithm to get the final model. In contrast, StructHKL does the classification in structured output space (rules and parameters are learned simultaneously for structured output classification) and performs better². The results are summarized in Table 6.2 and illustrated in Figure 6.2³. We observed that the proposed StructHKL approach reports better micro-averaged accuracy than all other approaches. While our macro average accuracy is slightly less than FICRF, StructSVM, RELHKL + StructSVM and better than others, our standard deviation is much less. This suggests that the model is not skewed towards any particular activity. In general, our approach exhibits much lower standard deviation, reflecting its consistency.

In our experiments on PlaceLab dataset, we observed that the performance of standard HMM, B&B structure learning assisted HMM, and RELHKL without transition

²The basic HMM and CRF approaches were performed by considering features that are negations also. For feature learning, when we considered negations, due to the sparsity of inputs, the constructed features were dominated by a lot of meaning less rules. Therefore, we restricted our work to only positives.

³Since the cross validation sets are different, the Std. HMM, B&B HMM and Greedy FIHMM results are different from those that are given in table 6.1.

Table 6.2: Micro average accuracy and macro average accuracy of classification in percentage using standard HMM, B&B learning assisted HMM, greedy feature induction assisted HMM, StructSVM, CRF, CRF with feature induction, RELHKL without transitions, RELHKL + StructSVM and the proposed StructHKL approach on UA dataset.

	Micro avg.	Macro avg.
Std. HMM	25.40 (± 18.55)	21.75 (± 12.12)
B&B HMM	29.54 (± 20.70)	16.39 (± 02.74)
Greedy FIHMM	58.08 (± 10.14)	26.84 (± 04.41)
StructSVM	58.02 (± 11.87)	35.00 (± 05.24)
CRF	48.49 (± 05.02)	20.65 (± 04.82)
FICRF	59.52 (± 11.76)	33.60 (± 07.38)
RELHKL	46.28 (± 11.44)	23.11 (± 07.46)
RELHKL+StructSVM	55.74 (± 10.88)	38.56 (± 10.68)
StructHKL	63.96 (± 05.74)	32.01 (± 03.04)

features was poor and the greedy feature induction assisted HMM did not converge at all. Therefore, we compare our results with (a) StructSVM approach (Tsochantaridis et al. 2004), (b) Conditional Random Field (CRF) (Lafferty, McCallum & Pereira 2001), and (c) Conditional Random Field with Feature Induction (FICRF) (McCallum 2003, 2002). The results are summarized in Table 6.3 and illustrated in Figure 6.3 and Figure 6.4 respectively for subject one and subject two. Our results show that StructHKL returns better results than other approaches in micro-averaged accuracy for both subject one and two, while maintaining comparable macro-averaged class accuracies. Our approach shows less standard deviation in subject one data while giving slightly higher standard deviation than most of the other approaches in subject two data.

Our statistical significance tests for micro-average using Wilcoxon Signed Rank Test (Siegel & Castellan 1988) indicate a 0.01 level of significance over all other approaches we compared against for both the UA dataset and the placeLab dataset.

In a setting with n labels and N basic inputs, an exhaustive search for optimum features needs evaluation at $n \times 2^N$ nodes. This amounts to 131072 nodes in UA data and

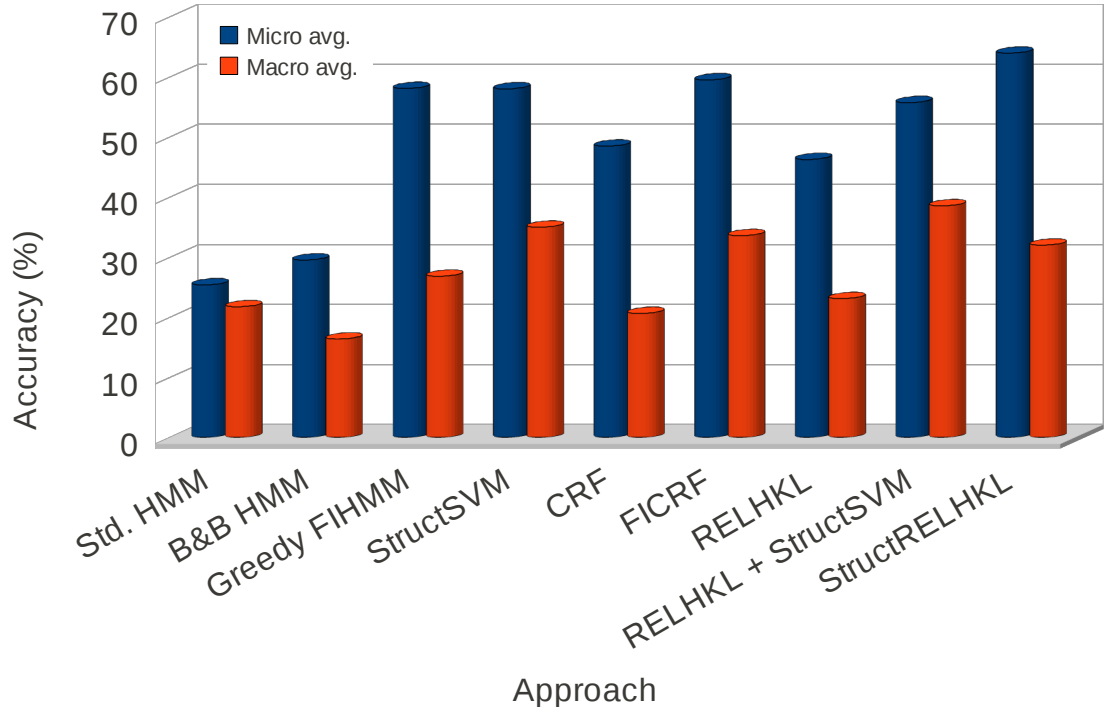


Figure 6.2: Performance comparison of different approaches on UA dataset.

Table 6.3: Micro average accuracy and macro average accuracy of classification in percentage using StructSVM, CRF, CRF with feature induction and the proposed StructHKL approach on PlaceLab dataset. (Std.HMM, B&B HMM, Greedy FIHMM, and RELHKL without transitions either failed to give comparable results or did not converge)

		Micro avg.	Macro avg.
Subject 1	StructSVM	75.03 (± 04.51)	26.99 (± 07.73)
	CRF	65.54 (± 06.80)	31.19 (± 07.39)
	FICRF	68.52 (± 07.19)	29.77 (± 03.59)
	StructHKL	82.88 (± 0.43)	28.92 (± 01.53)
Subject 2	StructSVM	63.49 (± 02.75)	25.33 (± 05.8)
	CRF	50.23 (± 06.80)	27.42 (± 07.65)
	FICRF	51.86 (± 07.35)	26.11 (± 05.89)
	StructHKL	67.16 (± 08.64)	24.32 (± 02.12)

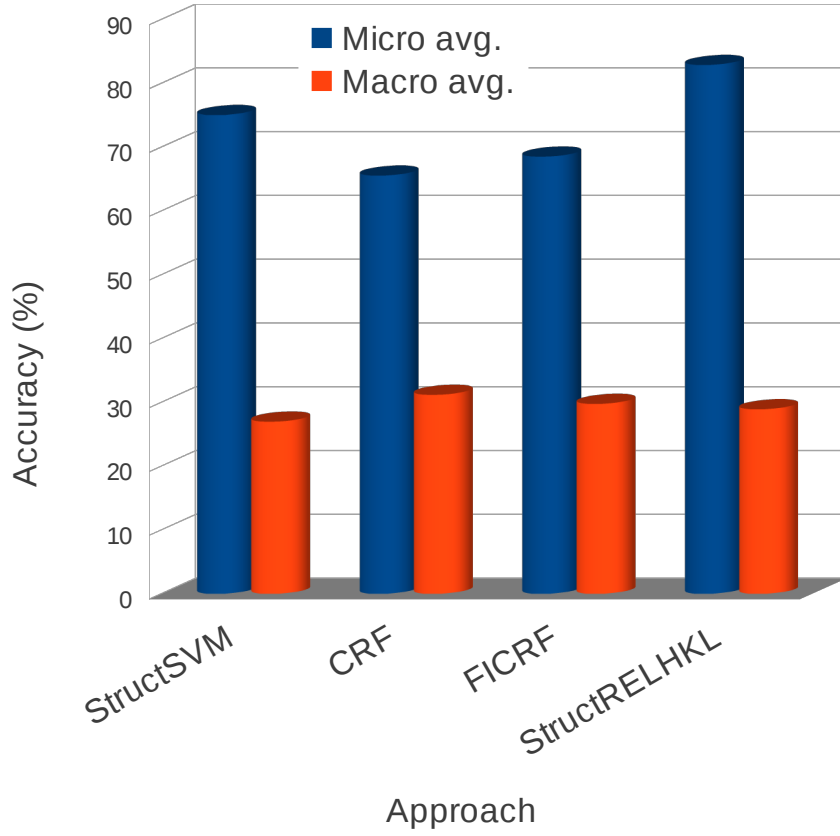


Figure 6.3: Performance comparison of different approaches on PlaceLab subject one data (Std.HMM, B&B HMM, Greedy FIHMM, and RELHKL without transitions either gave worse results or did not converge).

to the order of 10^{22} in PlaceLab data, which is computationally infeasible. In contrast, due to the active-set algorithm and sufficiency condition check, our approach explores only a few thousand nodes and converges in 24 hours approximately. In our experiments we have observed that traditional sequence labeling algorithms such as HMM and CRF take a few seconds for training. In contrast, greedy feature induction approaches such as FIHMM, FICRF take a few hours for training. RELHKL (without transitions) took 24 hours approximately for training. StructSVM's running time ranges between a few hours to a few days, depending on the regularization parameter used. Since all approaches use dynamic programming for prediction, time for inference depends only on the number of features used. While approaches such as StructSVM, HMM, and CRF take less than a second for inference, feature induction approaches such as FIHMM, FICRF, RELHKL, RELHKL + StructSVM and StructHKL take 1.2 seconds for inference.

The StructHKL approach discovered rules such as

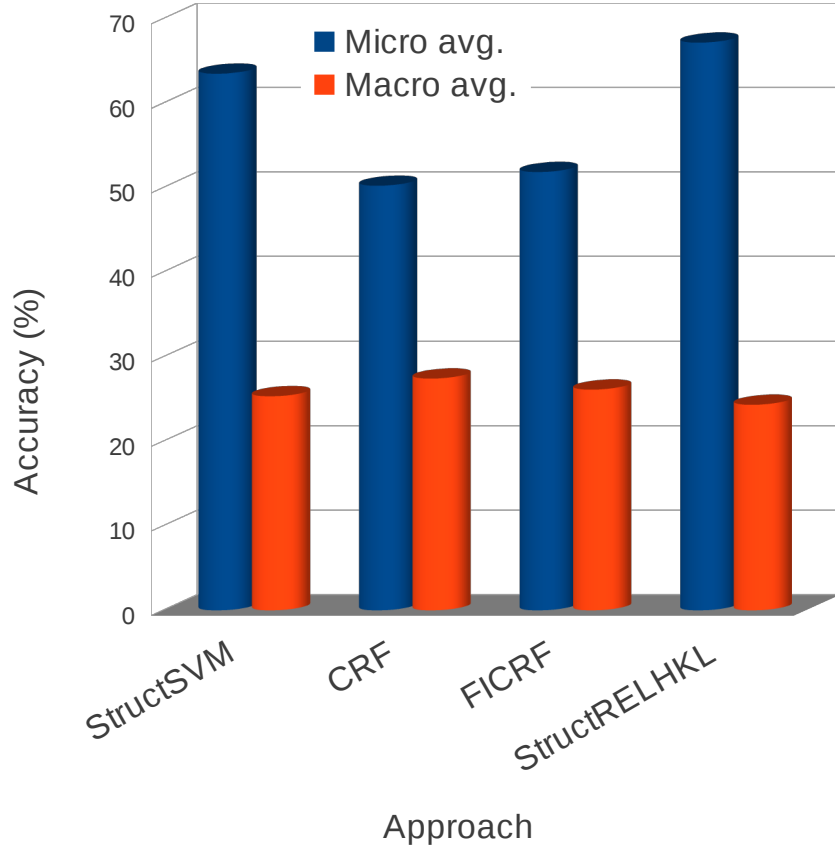


Figure 6.4: Performance comparison of different approaches on PlaceLab subject two data (Std.HMM, B&B HMM, Greedy FIHMM, and RELHKL without transitions either gave worse results or did not converge).

$$usingToilet(t) \leftarrow bathroomDoor(t) \wedge toiletFlush(t),$$

$$sleeping(t) \leftarrow bedroomDoor(t) \wedge toiletDoor(t) \wedge bathroomDoor(t),$$

$$preparingDinner(t) \leftarrow groceriesCupboard(t),$$

and the like.

The conjunction $bathroomDoor(t) \wedge toiletFlush(t)$ strongly indicates that the activity is *usingToilet* while *groceriesCupboard* indicates a higher chance of *preparingDinner*. Similarly $bedroomDoor(t) \wedge toiletDoor(t) \wedge bathroomDoor(t)$ increases the chance of predicting *sleeping* as the activity. This is reasonable as people access these doors during the night before going to sleep and the sensors at *bedroomDoor*, *toiletDoor*, and *bathroomDoor* fire once when the person accesses the door, and goes to off-mode while s/he is sleeping. However, since, the conjunction just before sleep gives a higher weight to the activity *sleeping*, the weight gets accrued and gets combined with transition weights

to accurately predict the activity as *sleeping*. We now discuss our experiments on learning complex relational features derived from relative sequence positions.

6.1.3 Learning Complex Relational Features for Sequence Labeling

In section 3.5, we identified composite features (\mathcal{CF}) as the relevant category of features for capturing sequence information among input variables. We then proved that \mathcal{CF} s can be constructed from a simpler class of features called absolute features (\mathcal{AF}). We have also shown that \mathcal{CF} s can be constructed efficiently from \mathcal{AF} s by leveraging the StructHKL framework. We seek to leverage optimal feature learning in all the steps of relational feature induction, which can be addressed either by i) enumerating \mathcal{AF} s and discovering their compositions (\mathcal{CF}) using StructHKL or by ii) developing methods to learn optimal \mathcal{AF} s (or \mathcal{CF} s directly).

As the space of \mathcal{AF} s is prohibitively large, enumerating all possible \mathcal{AF} s is infeasible in real world settings. We therefore selectively enumerate \mathcal{AF} s based on the support of the rule/feature in the data (weak relevance) and then leverage the StructHKL algorithm to learn composite features. We use Warmr (Dehaspe & Toivonen 1999, Dehaspe & Toivonen 2000), an ILP data mining algorithm that learns frequent patterns reflecting one to many and many to many relationships, to learn absolute features. Warmr uses an efficient level wise search through the pattern space and, with proper language bias, and generates absolute features. The absolute features learned by Warmr are then input to StructHKL code to learn the structure and parameters of the final model. We refer to this approach as \mathcal{CF} from enumerated \mathcal{AF} approach (enum \mathcal{AF}).

For the second option discussed, we leverage a relational kernel that computes the similarity between instances in an implicit feature space of \mathcal{CF} s. To this end, we employ the relational subsequence kernel (Bunescu & Mooney 2006) at each sequence position (over a time window of inputs around the pivot position) for the classification model. We refer to this approach as Relational Subsequence Kernels for StructSVM approach (SubseqSVM). We now discuss the datasets we used for these experiments and then discuss the results.

We use two publicly available activity recognition datasets, that is, i) the data provided by van Kasteren et al. (2008), which we discussed at the beginning of this

section, where we perform a 4 fold cross validation, as discussed in the previous section and ii) the relational activity recognition data provided by Landwehr et al. (2009) of Katholieke University, Leuven. We refer to the data as KU data. The data has been collected from a kitchen environment with 25 sensors/RFID attached to objects. There are 19 activities annotated. The data has been divided into 20 sequences. Each sequence has a length of 250 approximately. In this data, we perform our experiments in a leave one out cross-validation setup and report the average of the accuracies returned from each fold.

We have compared our approach with TildeCRF (Gutmann & Kersting 2006) and StructSVM (Tsochantaridis et al. 2004). TildeCRF is the state-of-the-art ILP approach for learning relational features for sequence labeling, and works in the same feature space that we are interested in, while we use StructSVM as a baseline for this experiment. In our experiments with StructSVM, individual basic features are inputs.

The comparison of results for UA data is outlined in Table 6.4 and Figure 6.5. Results show that our approaches for learning complex features for sequence labeling, that is, $\text{enum}\mathcal{AF}$ and SubseqSVM performed better than the base line approach (StructSVM) and the state-of-the-art approach (tildeCRF). Although $\text{enum}\mathcal{AF}$ optimally finds \mathcal{CF} s as conjunctions of (selectively enumerated) \mathcal{AF} s, the step for selectively enumerating \mathcal{AF} s is based on heuristics. In contrast, SubseqSVM works on a convex formulation and learns an optimal model, thus giving the best performance. This explains the difference in the performances of our two approaches.

The comparison of results for KU data is outlined in Table 6.5 and Figure 6.6. Since a single sequence step in this data has only one input feature, the feature space is not rich enough to evaluate the efficiency of our approaches. For this reason, the performance of our approaches is inferior to the baseline and the state-of-the-art. The baseline reported the best performance. While the performance of SubseqSVM approach is slightly inferior to the baseline and the state-of-the-art, $\text{enum}\mathcal{AF}$ performed badly in this data.

In UA data, both our approaches ($\text{enum}\mathcal{AF}$ and SubseqSVM) took 24 hours approximately to train the model. Whereas, StructSVM and TildeCRF took 20 hours and 0.5 hours, respectively, for training. In KU data, $\text{enum}\mathcal{AF}$ took around 24 hours and SubseqSVM took approximately 1.5 hours to train the model while StructSVM and TildeCRF took 15 hours and 10 minutes, respectively. Inference with SubseqSVM takes, on average,

Table 6.4: Micro average accuracy and macro average accuracy of classification in percentage using tildeCRF, \mathcal{CF} from enumerated \mathcal{AF} approach (enum \mathcal{AF}), StructSVM (with basic inputs as features) and relational subsequence Kernels for StructSVM approach (SubseqSVM) on UA data (exploring the space of \mathcal{CF} s).

	Micro avg.	Macro avg.
TildeCRF	56.22 (± 12.08)	35.36 (± 6.55)
StructSVM	58.02 (± 11.87)	35.00 (± 05.24)
enum \mathcal{AF}	60.36 (± 6.99)	30.39 (± 4.31)
SubseqSVM	65.25 (± 4.81)	29.34 (± 2.78)

six hours for UA data and eight minutes for KU data. Where as other approaches take a few seconds only for inference. The difference is due to the kernel computation. Our statistical significance test for micro-average using Wilcoxon Signed Rank Test (Siegel & Castellan 1988) indicates a 0.01 level of significance with SubseqSVM over other approaches on UA data. We now present an analysis of the progression of results on UA data, using different categories of features we have experimented.

The progression of results on UA data based on feature categories is shown in Table 6.6 and Figure 6.7. The baseline for sequence labeling can be one among the approaches that considers only basic inputs for model construction. HMM, CRF, and StructSVM falls into this category. Since StructSVM is the state-of-the-art in this category, we use StructSVM results for comparison. The next level of features is the set of simple conjuncts \mathcal{SC} , which are conjunctions of input features at a single sequence step. \mathcal{SC} s capture relationships among co-occurring features. We present our StructHKL results. Next is the category of \mathcal{CF} s, which are capable of capturing input relationships across time steps in sequence labeling. We presented two approaches for \mathcal{CF} s, that is, enum \mathcal{AF} approach and the SubseqSVM approach. Since the SubseqSVM approach performed better, we report that here.

We now present evaluation of our related contributions. The following section discusses our experiments on the approach for pruning the search space for satisfiability checking in weighted first order logical systems.

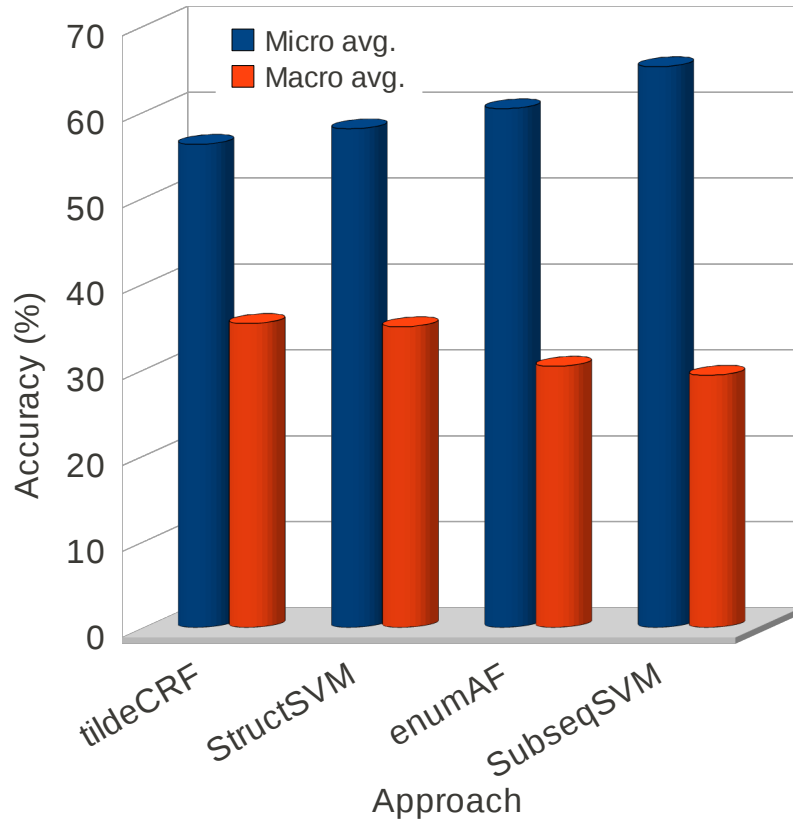


Figure 6.5: Performance comparison of different approaches on UA data (exploring the space of \mathcal{CF} s).

6.2 Pruning Search Space for Satisfiability in Weighted Horn Clauses

We now discuss the results of our contribution in the area of satisfiability based inference in first order weighted horn clause settings.

In this thesis, we have proposed and developed an approach that prunes a major part of the search space for satisfiability, which we refer to as *Modified- T_Σ* , and thereby speeds up satisfiability in first order weighted horn clauses. The overall approach is referred to as HornSAT.

We used the uwce knowledge base and dataset provided by alchemy (Richardson & Domingos 2006) for our experiments after making small modifications to make the clause set horn. The constants given as the evidence set is considered as the complete domain for each variable. We have performed three experiments on each dataset. The first experiment does the complete groundings and employs MaxWalkSAT on that. the second

Table 6.5: Micro average accuracy and macro average accuracy of classification in percentage using various approaches on KU data. As a single sequence step in this data has only one input feature, the feature space is not rich enough to evaluate the efficiency of our approaches. For this reason, the performance of our approaches is slightly inferior, as observable from the table.

	Micro avg.	Macro avg.
$\tilde{\text{tildCRF}}$	66.04 (± 13.50)	84.01 (± 8.76)
StructSVM	66.35 (± 17.16)	66.64 (± 16.04)
$\text{enum}\mathcal{AF}$	33.24 (± 15.72)	23.02 (± 11.13)
SubseqSVM	64.66 (± 8.42)	63.08 (± 7.05)

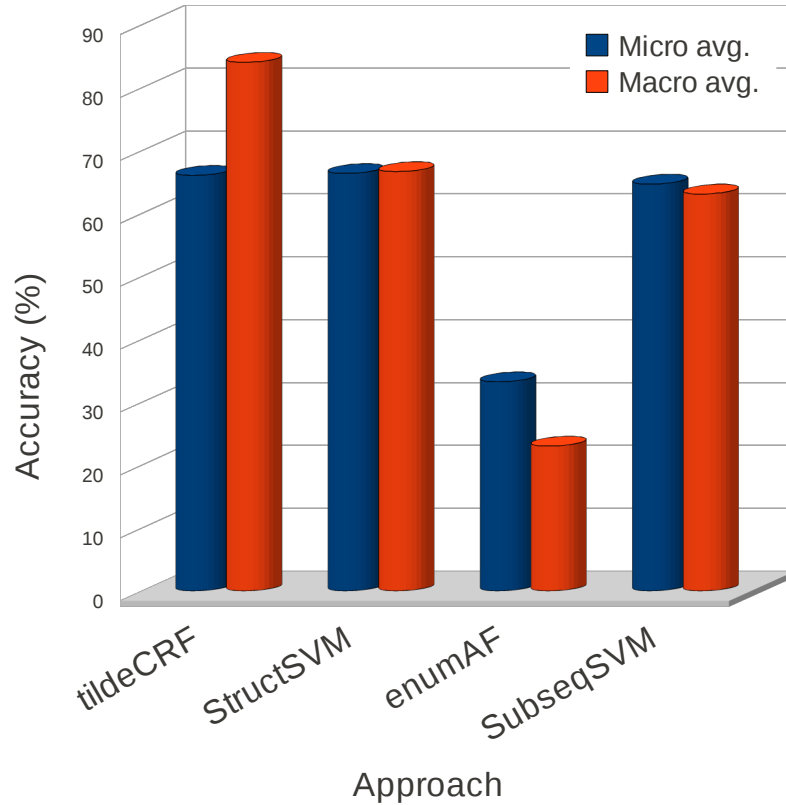


Figure 6.6: Performance comparison of different approaches on KU data (exploring the space of \mathcal{CF} s).

grounds the clauses with pruning and employs traditional MaxWalkSAT with random truth assignments. The third experiment employs MaxWalkSAT on the pruned clauses

Table 6.6: progression of sequence labeling results on UA data based on feature categories.

Feature	Approach	Micro avg.	Macro avg.
Basic	StructSVM	58.02 (± 11.87)	35.00 (± 05.24)
\mathcal{SC}	StructHKL	63.96 (± 05.74)	32.01 (± 03.04)
\mathcal{CF}	SubseqSVM	65.25 (± 4.81)	29.34 (± 2.78)

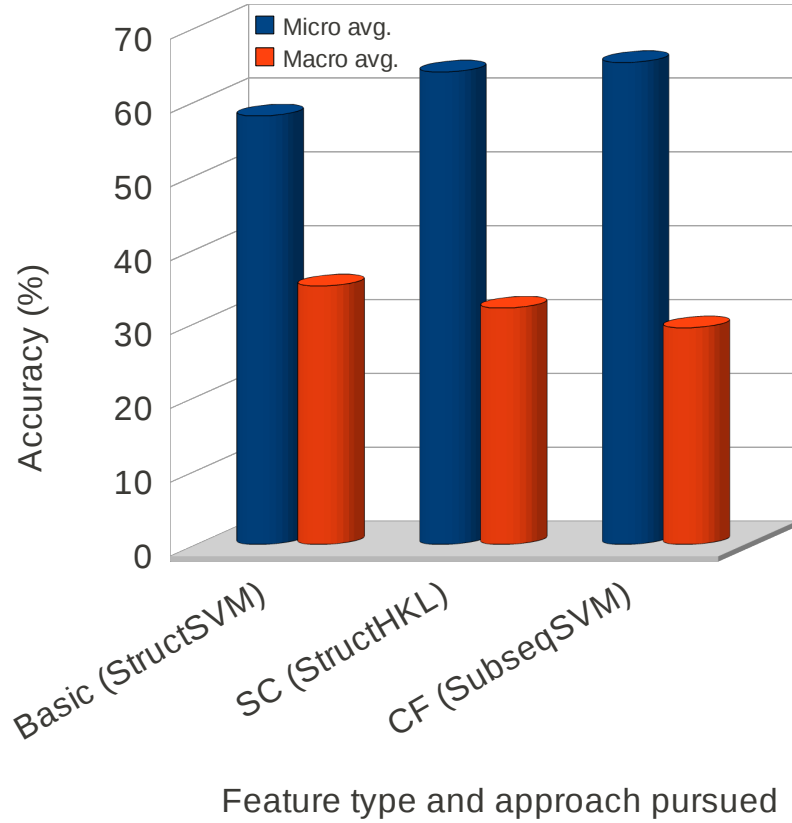


Figure 6.7: progression of sequence labeling results on UA data based on feature categories.

set with the initial truth assignment returned by *Modified-T_Σ* (HornSAT). Evidence set of different sizes are used. The number of grounding for uwce language and AI KBs with different evidence set using the complete grounding and pruned approach are compared in Table 6.7. The time versus cost analysis of different approaches discussed here is given in Table 6.8. Figures 6.8, 6.8, 6.8 portrays the results when 181 atoms of uwce language dataset, 87 atoms of uwce language dataset and 766 atoms of uwce AI dataset are used, respectively, as the evidence set. Experimental results show that the proposed method

Table 6.7: Comparison of number of groundings after i) Complete grounding and ii) Pruning on uwce knowledge base.

Evidence set	Complete grounding	After pruning
language 181 atoms	508788	6908
language 87 atoms	177738	3205
AI 766 atoms	Memory error	182690

Table 6.8: Performance comparison of satisfiability approaches on uwce knowledge base and different evidence sets.

Evidence set	Complete grounding		Pruned		HornSAT	
	+ MaxWalkSAT		+ MaxWalkSAT			
	Converged	Time	Converged	Time	Converged	Time
	cost	(ms)	cost	(ms)	cost	(ms)
lang. 181 atm.	90.452	2475736	70.265	1823778	70.265	6896
lang. 87 atm.	81.463	2329459	37.892	1098285	37.892	2351
AI 766 atm.	Memory error		344.584	7507578	344.584	7462967

outperforms the traditional approach in terms of memory and speed.

We used similar maxWalkSAT settings for our experiments (with or without pruning). Since our goal was to evaluate the efficiency of pruning under similar maxWalkSAT settings, we did not try large maxWalkSAT tries or flips. Trying large maxWalkSAT tries or flips does, of course, result in a longer running time⁴.

As described earlier, inference in sequence labeling is typically performed using the dynamic programming approach, the Viterbi algorithm (Forney 1973). We hypothesize that there could be complex first order settings where it is possible to have complex models with a huge set of groundings. In such sequence labeling settings, if it is feasible to ground all the variables, the Viterbi algorithm (Forney 1973) is the best choice. In all other cases

⁴Experimental comparison with LazySAT under similar conditions is not trivial. However, the LazySAT paper reports a comparable running time for LazySAT with maxWalkSAT. Whereas, our approach converges much quicker than maxWalkSAT.

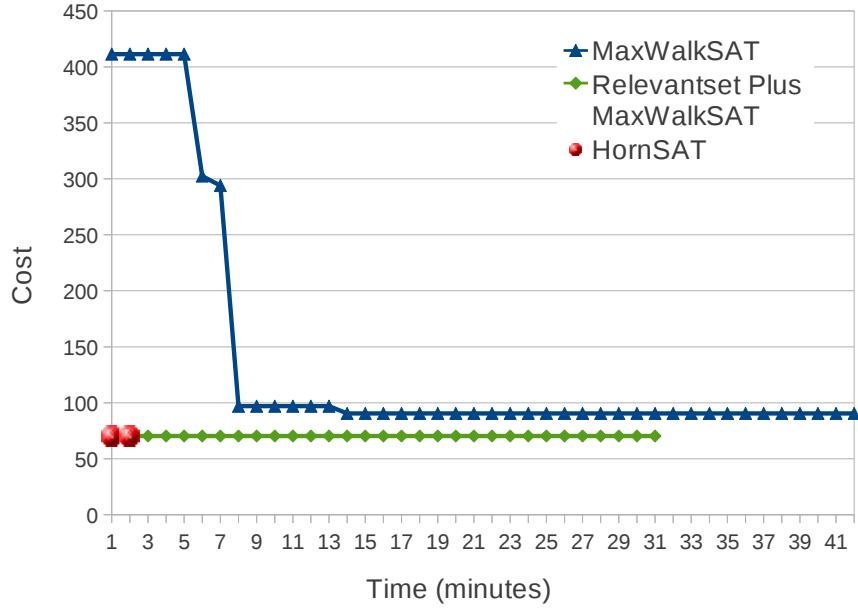


Figure 6.8: Performance comparison of different satisfiability approaches on uwcse language KB. All 181 atoms are given as evidence.

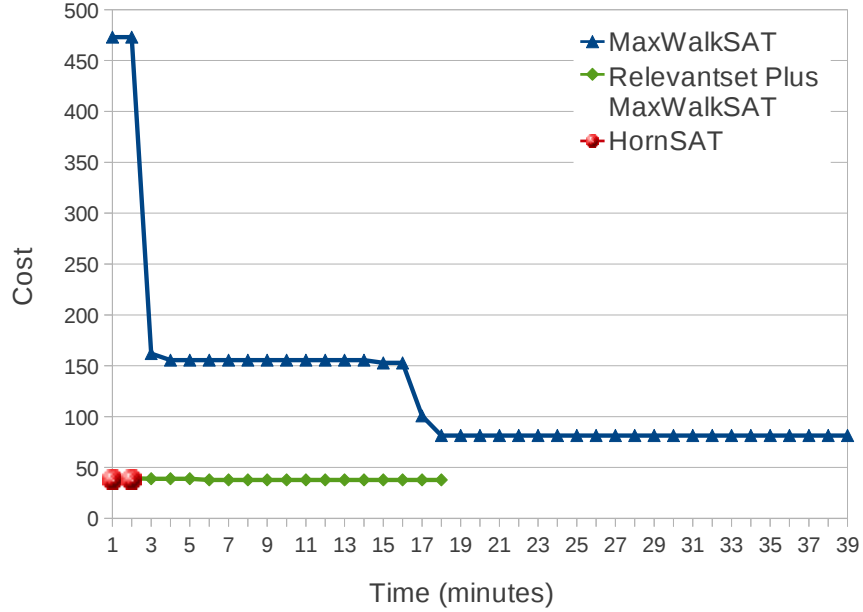


Figure 6.9: Performance comparison of different satisfiability approaches on uwcse language KB. 87 atoms are given as evidence.

(sequence labeling or general), we suggest using our satisfiability based inference⁵. Here we compare the performance of both Viterbi algorithm and our satisfiability based approach in a subset of UA data. While the Viterbi algorithm globally assigns labels to

⁵Since we do not have access to such a sequence labeling data to demonstrate the validity of our approach, we presented our results in general settings in the previous paragraphs

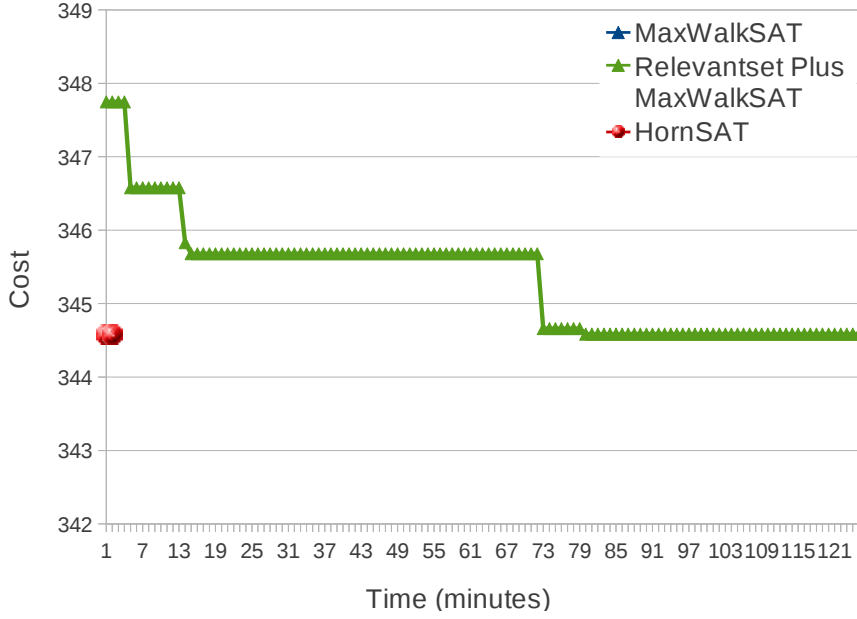


Figure 6.10: Performance comparison of different satisfiability approaches on uwce AI KB. All 766 atoms are given as evidence. In this experiment, complete grounding resulted in an out-of-memory error

each position of a sequence optimally and in a fast manner, the satisfiability approach (HornSAT) we proposed uses a modified maxSAT approach to find the final truth assignments after pruning the search space. The maxSAT algorithm used in HornSAT is greedy and thus the final interpretation may not be optimal. Moreover, the maxSAT step in the approach might do a large number of flips and tries, thus making the inference slower than the Viterbi algorithm⁶. We compare the proposed HornSAT results with that of the Viterbi algorithm in solving inference in activity recognition problems (sequence labeling) in Table 6.9 and illustrated in Figure 6.11⁷. Here, we train a model using StructHKL on a subset of the UA data (van Kasteren et al. 2008); two experiments are then performed for inference using the learned model on a subset of test data. The two inference methods performed are the Viterbi algorithm and the HornSAT. The aim is to predict labels for a sequence of length 1000, with 8 activity labels and 14 sensors. As discussed above, the Viterbi algorithm performed much better than the HornSAT. Moreover, the dynamic programming approach is much faster. HornSAT took seven hours to converge while the

⁶In other first order problems where dynamic programming is infeasible, the proposed HornSAT performs better than the traditional satisfiability solvers.

⁷Since the training and test sets are different from that used in table 6.1 and 6.2, the Viterbi algorithm results are different here.

Table 6.9: Micro average accuracy and macro average accuracy of classification in percentage using two methods of inference, that is, the Viterbi algorithm and the HornSAT. A model is first trained using StructHKL and the two experiments for inference are performed. The time taken for inference for each of the approaches are also reported.

	Micro avg.	Macro avg.	Time
Viterbi	94.5	33.31	3 sec.
HornSAT	45.8	26.57	7 hrs.

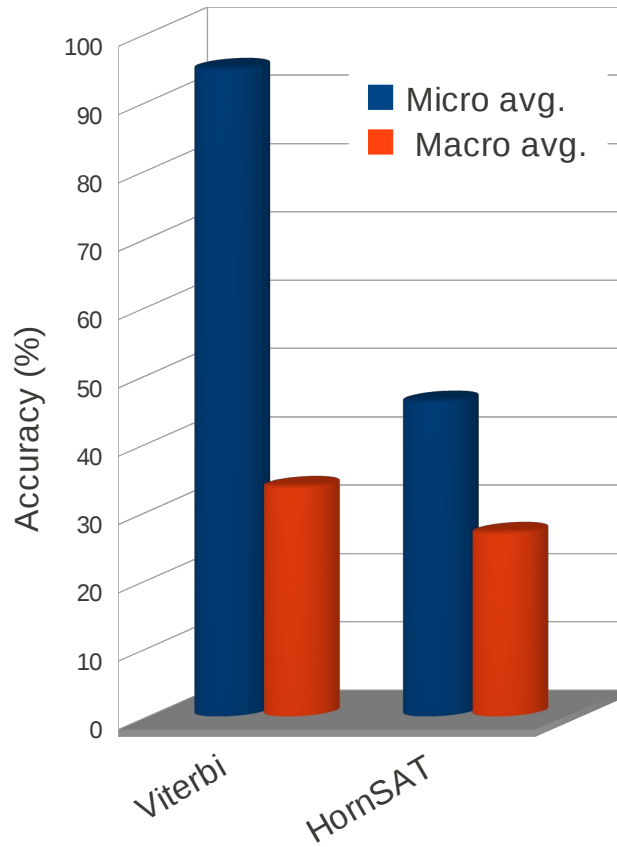


Figure 6.11: Performance comparison of the two inference approaches on activity recognition data (sequence labeling).

Viterbi algorithm took a few seconds.

We now discuss the evaluation of our integrated non-parametric dimensionality reduction approach in the next section.

6.3 Optimally Extracting Discriminative Disjunctive Features for Dimensionality Reduction

We now discuss the results of our contribution in a related area, that is, integrated non-parametric dimensionality reduction using hierarchical kernel learning. We discuss our experimental setup and compare our results with the state-of-the-art methods for dimensionality reduction. We report our results on two publicly available datasets, the first set belonging to UCI data repository (Frank & Asuncion 2010) and the second is the 20 *Newsgroups* data (Lang n.d.).

6.3.1 UCI data

We performed our first set of experiments on the following data-sets: *Breast-cancer*(286 instances, 9 attributes), *Wisconsin breast-cancer*(699 instances, 10 attributes), *Hepatitis*(185 instances, 19 attributes), *Monk-1*, *Monk-2*, *Monk-3*(432 instances, 7 attributes), *Transfusion*(748 instances, 5 attributes), *Tic-Tac-Toe*(958 instances, 9 attributes) and *Vote*(435 instances, 16 attributes), from the UCI repository. Each of these datasets corresponds either to a binary or a multi-class classification problem. We performed experiments on each of the above datasets with all the wrapper-based dimensionality reduction approaches available in Weka (Hall et al. 2009), a machine learning toolbox. Out of all the wrapper methods provided by weka, only those which give comparable results are reported. For each dataset, and for each choice of the wrapper, we considered two choices for the classifier: 1) a 2-norm SVM (LibSVM implementation Chang & Lin (2011)) and 2) a 1-norm SVM (LibLinear implementation Fan et al. (2008)). The comparison of the methods mentioned earlier with our dimension reduction approach is provided in Table 6.10. Among the approaches that we compare against, we report the accuracies of only those (namely, Correlated Subset Evaluator, Consistency Subset Evaluator and Filtered Subset Evaluator) which perform comparably or better. For each of the subset selection approaches, various search strategies such as, BestFirst, GreedyStep, LinearFwd, Rank and SubsetSizeFwd have been employed and the results reported.

Subset Evaluator	Search Method	Breast-cancer		Wisconsin		Hepatitis		Transfusion		Vote		Monk-1		Monk-2		Monk-3		Tic-Tac-Toe		20NewsGroups	
		\mathcal{L}_2	\mathcal{L}_1	\mathcal{L}_2	\mathcal{L}_1	\mathcal{L}_2	\mathcal{L}_1	\mathcal{L}_2	\mathcal{L}_1	\mathcal{L}_2	\mathcal{L}_1	\mathcal{L}_2	\mathcal{L}_1	\mathcal{L}_2	\mathcal{L}_1	\mathcal{L}_2	\mathcal{L}_1	\mathcal{L}_2	\mathcal{L}_1	\mathcal{L}_2	\mathcal{L}_1
Correlation	BestFirst	74.64	70.65	93.84	94.72	95.0	93.75	91.04	91.04	96.28	96.28	69.37	74.94	63.34	59.63	97.22	97.22	80.88	73.88	93.67	89.10
	GreedyStep	74.64	67.75	93.84	94.72	95.0	93.75	91.04	91.04	96.28	96.28	69.37	74.94	63.34	59.63	97.22	97.22	80.88	73.88	93.76	89.10
	LinearFwd	74.64	67.75	93.84	94.72	95.0	93.75	91.04	91.04	96.28	96.28	69.37	74.94	63.34	59.63	97.22	97.22	80.88	73.88	92.09	89.98
	Rank	74.64	70.65	93.84	94.72	94.15	93.75	91.04	91.04	96.28	96.28	69.37	74.94	63.34	59.63	97.22	97.22	80.88	73.88	92.26	91.38
	SubsetSize	74.64	70.65	93.84	94.72	94.15	93.75	91.04	91.04	96.28	96.28	69.37	74.94	63.34	59.63	97.22	97.22	80.88	73.88	92.09	89.98
Consistent	BestFirst	67.39	72.46	95.31	95.89	88.75	86.25	92.37	90.78	95.26	96.28	100.0	83.29	93.27	59.63	93.03	97.22	100.0	76.49	89.98	92.97
	GreedyStep	70.29	67.75	95.75	95.01	87.0	86.25	91.18	71.92	94.40	93.10	100.0	74.94	87.93	59.63	93.5	97.22	99.58	98.33	89.98	92.97
	LinearFwd	70.65	71.01	94.72	94.43	88.75	87.5	91.44	90.78	97.41	95.69	100.0	83.3	93.27	59.63	97.22	97.22	99.68	75.44	87.34	89.28
	Rank	68.48	68.48	94.57	92.08	92.5	91.25	91.57	89.84	94.6	93.1	100.0	74.94	91.87	59.63	93.27	97.22	99.68	80.45	93.67	91.91
	SubsetSize	70.65	71.01	94.72	94.43	88.75	88.75	91.04	91.04	96.28	96.28	83.3	66.59	93.27	59.63	97.22	97.22	99.68	75.44	87.34	89.28
Filtered	BestFirst	77.54	70.29	94.43	94.14	91.25	91.25	90.1	90.1	96.28	96.28	74.94	74.94	62.41	59.63	97.22	97.22	70.01	70.01	93.14	91.56
	GreedyStep	77.54	70.29	94.43	94.14	91.25	91.25	90.1	90.1	96.28	96.28	74.94	74.94	62.41	59.63	97.22	97.22	70.01	70.01	93.14	91.56
	LinearFwd	77.54	70.29	94.43	94.14	91.25	91.25	90.1	90.1	96.28	96.28	74.94	74.94	62.41	59.63	97.22	97.22	70.01	70.01	90.51	87.34
	Rank	77.14	70.29	94.43	94.14	88.75	92.50	91.04	91.04	96.28	96.28	74.94	74.94	62.41	59.63	97.22	97.22	70.01	70.01	84.88	85.86
	SubsetSize	77.89	70.29	94.43	94.14	91.25	91.25	90.1	90.1	96.28	96.28	74.94	74.94	62.41	59.63	97.22	97.22	70.01	70.01	90.51	87.34
Integrated Dim. Red.		75.36 \pm 0.49	96.34 \pm 0.19	91.25 \pm 0.29	91.04 \pm 0.30	96.28 \pm 0.17	100.0 \pm 0.0	85.15 \pm 0.38	97.22 \pm 0.16	100.0 \pm 0.0	94.55 \pm 0.23										
Baseline \mathcal{L}_2		71.01	95.89	87.50	91.17	94.40	100.0	87.93	93.50	99.58	91.38										
Baseline \mathcal{L}_1		70.65	95.01	86.25	89.57	93.10	74.94	61.25	97.21	98.32	93.14										

Table 6.10: Comparison of accuracies (in percentage) of different dimensionality reduction approaches on the UCI dataset and the 20NewsGroups *alt.atheism* vs. *talk.religion.misc* problem. Here (\mathcal{L}_2) and (\mathcal{L}_1) refer to 2-norm and 1-norm SVM employed over the set of features selected by the Weka Feature Subset Selection Wrappers. For *IntegratedDim.Red.* the *Accuracy \pm RMS Error over Cross Validation Folds* has been reported. Baseline \mathcal{L}_2 and \mathcal{L}_1 refer to 2-norm and 1-norm SVM respectively, without applying any feature subset selection

Table 6.11: Comparison of accuracies of different approaches on 20 *Newsgroups* dataset (MedLDA accuracy is not exactly reported in (Zhu et al. 2010) and therefore, it has been calculated from the relative improvement ratios reported in (Xu 2010). The results of the competitor approaches are the best ones obtained by the authors by cross validating on the parameter, number of topics.).

Approach	Accuracy
$\mathcal{L}_1 SVM$	93.14%
$\mathcal{L}_2 SVM$	91.38%
MMpLSA	84.7%
DiscLDA	83.0%
MedLDA	73.12%
Integ.Dim.Red.	94.55%

Further all accuracies (except for *Monk-1*, *Monk-2* and *Monk-3*), are presented as averages over a 4-fold cross validation. For *Monk-1*, *Monk-2* and *Monk-3*, the train and test splits provided in the UCI repository have been used.

We observe that our approach (Integrated Dim. Red.) performs consistently better than most of the other approaches we compared against. On each dataset, our results are comparable with the best results among all the approaches and better in many cases. Wilcoxon Signed Rank Test was performed in order to compare each of the Feature Selection wrappers provided by Weka, when LibSVM (2-norm regularized SVM (\mathcal{L}_2)) and LibLinear (1-norm regularized SVM (\mathcal{L}_1)) are used for model building. The results show that our approach is significantly better, at 0.01 level of significance⁸ than each of the 15 Feature Subset Selection methods provided by Weka, when LibLinear is used as a model builder. Correspondingly when LibSVM is used for model building, our model is found to be significantly better, at 0.05 level of significance (that is, error probability), than only three out of the 15 Feature Subset Selection methods, namely those in which *Consistent Subset Evaluator* is used with the Search Methods *Greedy-Step* and *Linear-Forward* and when *Filtered Subset evaluator* is used with *Rank* as the Search Method,

⁸Significance tests are done with all the 10 datasets taken together and therefore the numbers are not pertaining to any one dataset.

while for the remaining methods, our approach is found to have comparable performance.

For the baseline 1-norm and 2-norm regularized SVM, where no feature selection has been applied, the Wilcoxon signed rank test indicates that our method, *Integrated Dimension Reduction*, is significantly better than 2-norm SVM at the 0.05 level of significance and also significantly better than 1-norm SVM at the 0.005 level of significance. This indicates that the sparse variant of SVM with 1-norm regularization cannot leverage feature selection with model learning, whereas our approach can simultaneously learn dimension reduction and optimal model building better and suffers less from over-fitting.

Some wrappers that use ChiSquare, GainRatio, InfoGain, LSA-based, PCA-based or Relief-based attribute evaluator showed significantly worse performance and are not included in the table. The training time for our approach is comparable to the approaches it is being compared against.

6.3.2 20 Newsgroups data

In order to compare against the current state-of-the-art supervised dimensionality reduction-cum-classification techniques, we evaluated our approach on the 20 *Newsgroups* dataset that contains postings to Usenet newsgroups. We apply our approach on the binary classification problem of distinguishing postings from two newsgroups *alt.atheism* and *talk.religion.misc*, which is considered to be a hard task, owing to the content similarity between them. This benchmark dataset has the train and test splits provided, thus making it convenient for us to compare against existing approaches. This dataset has been used as a benchmark data by most of the recent approaches. In Table 6.11, we present a comparison of the accuracy achieved by our approach with the best values reported by the existing approaches such as DiscLDA, MedLDA and MMpLSA on this dataset and in Table 6.10, the comparison has been reported on the feature selection wrappers provided by Weka.

We note that the proposed integrated dimensionality reduction approach outperforms other approaches. The number of disjunctions discovered (automatically) by our approach is 170. Some of the disjunctions reported are as follows:

$\{\textit{religion}, \textit{sandvik}, \textit{benedikt}\}$

⁸MedLDA accuracy is not exactly reported in Zhu et al. (2010) and therefore, it has been calculated from the relative improvement ratios reported in Xu (2010). The results of the competitor approaches are the best ones obtained by the authors by crossvalidating on the parameter, number of topics.

{*religion, kent, benedikt*}
 {*biblical, islam*}
 {*atheism, historical*}
 {*reading, writes*}{*fax, run, mode*}
 {*data, mode, graphics*}
 {*version, order, directory*}
 {*version, works, help, mail*}
 {*use, interested, need*}{*book, bill*}
 {*images, mode*}
 {*mode, algorithm*}
 {*works, run*}
 {*god, beliefs*}
 {*book, edu*}

MedLDA (Zhu et al. 2010) is reported to have a best improvement ratio of 0.2 at 20 topics, over its baseline which is a two-step LDA + SVM approach as well as the baseline used in Xu (2010) which is a two-step pLSA + SVM approach. Whereas MMpLSA Xu (2010), which gives best accuracy of 84.7% at 3 topics, shows a 0.39 relative improvement ratio over its baseline, that is, pLSA + SVM and a 2% relative improvement over DiscLDA and claims to perform better than MedLDA consistently. DiscLDA itself has the best accuracy of 83.0% which is achieved at 60 topics.

In addition to the improvement in performance, unlike other approaches that require the number of topics to be learned as an input, our approach automatically learns the number of disjunctive features. Since other methods do not discover the number of topics, they often have to resort to enumerating the classifier model’s performance for different values of this parameter and have to report the number of topics that leads to the best performance in classification. Moreover, parametric approaches to determine the number of topics may not yield an optimum result, especially if there is no integrated learning of the topic detection parameters and the classification parameters. As a result, an inappropriate number of topics may be used by such systems and thus, can result in over-fitting, as hypothesized by the authors of Xu (2010). We overcome this limitation by our non-parametric approach and learn an optimum number of disjunctive projections. Since our

model assimilates topic selection within the classifier and handles over-fitting by regularization in a unified manner, our approach guarantees an optimum model that performs efficient dimension reduction without compromising on the classifier performance.

Our approach takes around 26 hours on 20 newsgroups data for training. For other datasets, our approach takes a few minutes to train.

Chapter 7

Conclusion

Conventional approaches in sequence labeling capture the state-observation relationships (observation/emission dependency) at each step and the transition relationships between states during successive steps. These approaches, unless the input structure is provided upfront, ignore the input structure and assume independence among the individual input features. In this report, we have discussed the drawbacks of this assumption. On the other hand, considering the joint state of input features as a single variable results in exponential feature space and is infeasible in real world settings. Since strong independence or dependence assumptions have their own drawbacks, we proposed an intermediate solution, wherein, we seek to discover the input structure in the form of relational features that map compositions of input features to labels. These relational features can be represented in the form of definite clause rules/features. To get more insight into the feature spaces, we categorized definite features based on their complexity and identified feature classes that are relevant to sequence labeling. Among the feature categories, we identified Simple Conjuncts (\mathcal{SC}) and Composite Features (\mathcal{CF}) as useful categories for sequence labeling tasks. While \mathcal{SC} s are derived from inputs/observations at a single sequence position, \mathcal{CF} s are derived from inputs from multiple sequence positions relative to a pivot position.

We first developed a greedy feature induction approach for discovering \mathcal{SC} s for sequence labeling, which searches the lattice of possible features using a heuristic score. In each iteration of the search, our system drops or adds a conjunctive feature to the current emission model, combines it with the transition features, learns the parameters for the current HMM model and based on the performance of the current model on training data, decides on whether to include the new feature in the model or not. Although this approach yields a better performance than conventional approaches, since the approach

is greedy, an optimum model is not guaranteed. Searching exhaustively in the lattice of features for optimum models is not feasible in real world settings. To learn optimum \mathcal{SC} s, we proposed and developed a Hierarchical Kernel Learning based feature learning approach for Structured Output Spaces (StructHKL) such as sequence labeling.

The StructHKL approach optimally discovers features from a feature space that follows a partial order and that follows a property that the summation over descendant kernels of any node can be computed in polynomial time. This approach builds on the StructSVM framework and considers all possible features in the emission model. A hierarchical regularizer is employed to select a sparse set of useful features. The exponential search space is explored using an active set algorithm and the exponential constraint space is searched by a cutting plane algorithm.

Although StructHKL efficiently learns \mathcal{SC} s, it has limitations in discovering complex features that are derived from basic inputs at relative positions (\mathcal{CF}). We have shown that \mathcal{CF} s are conjunctions of features belonging to a simpler category called Absolute Features (\mathcal{AF}). We proposed and developed two strategies to learn optimal \mathcal{CF} s. One is to selectively enumerate absolute features and employ structHKL to learn their conjunctions. The second strategy is to incorporate a type of relational kernels called relational subsequence kernels, that implicitly capture the information about all possible relational features, without explicitly enumerating them.

We have demonstrated the efficiency of our approaches by evaluating on publicly available activity recognition datasets. From our experiments, it is evident that the accuracy of labeling increases when higher order features that capture input relations are used. Our model with features constructed from individual inputs at a single sequence position performed better than the conventional approaches that assumed independence among them. We also observed in our experiments that the models constructed from relational features derived from inputs at relative sequence positions performed better than the models with features derived from inputs at single sequence position. It is also observed that leveraging relational subsequence kernels for sequence labeling tasks captures relational information implicitly and performs better than other approaches.

Although feature learning for sequence labeling is the main contribution of this thesis, we have contributed in two other problem domains. One is to perform fast and memory efficient inference in general first order problem settings which cannot be solved

using dynamic programming and the other is learning optimal disjunctive projections for dimensionality reduction in a maximum margin framework. We present our concluding remarks on these contributions in the following paragraphs.

Several ground clauses formed as a result of propositionalization of first order horn formulae are satisfied by default and it is a wastage of resources to consider them for optimization. We presented an algorithm that prunes the search space and proved that the optimal solution must lie in the pruned space. Experiments indicate the scope for efficient inference using MaxSAT for the set of horn clauses. This in turn helps in fast and memory efficient inference in weighted first order horn clause settings.

Most existing approaches to dimensionality reduction for classification decouple the dimensionality reduction and classification phases. Some approaches are greedy while some others are parameterized, imposing a restriction on the learning system. In this paper we pose the requirement of optimal dimensionality reduction as an integrated non-parametric supervised max-margin optimization problem. We project the original features into the space of disjunctions and present algorithms inspired by the hierarchical kernel learning approach to select a sparse set of important disjunctions. We have shown analytically and empirically that our integrated approach learns optimal features in the form of interpretable disjunctions of features capturing similar discriminative information for classification and leads to accurate models. We discuss a few potential future directions that can be explored based on the theory developed in our research work.

Future research directions:

The main focus of this research has been improving the efficiency of sequence prediction problems. Our contributions can be extended to other structured output spaces such as trees, graphs, lattices and the like. The formulations that we derived, serve as templates for deriving formulations for other structured output spaces.

Our work can be extended to regular first order problem settings. It would be interesting to investigate the applicability of Hierarchical Kernel Learning to optimize all the steps of structure learning in Markov Logic Networks, without compromising the interpretability of resultant clauses.

Leveraging the learning approaches we developed to learn efficient models for other problem domains such as Natural Language Processing, Bio-informatics *etc.* is another

interesting future work.

Leveraging Hierarchical Kernel Learning for learning disjunctions for dimensionality reduction in domains with background knowledge (which we briefly discussed in chapter 5) is also a potential future work.

Appendix A

Derivations and Proofs

A.1 Sufficiency Condition

We first derive a variational characterization of the regularizer $\Omega_E(f_E)^2$

Lemma 26 of (Micchelli & Pontil 2005) says,

if $r \geq 0$ and $p = 1 + \frac{1}{r}$ then $(\sum_{j \in \mathbb{N}_d} |a_j|^{\frac{2}{p}})^{\frac{p}{2}} = \min_{\lambda_v \in \Delta_{d,r}} \sqrt{\sum_{j \in \mathbb{N}_d} \frac{a_j^2}{\lambda_j}}$

where $\Delta_{d,r} = \{\boldsymbol{\lambda} \in \mathbb{R}^d, \sum_{j \in \mathbb{N}_d} \lambda_j^r = 1, \lambda_j \geq 0\}$

By applying the above lemma on objective function $\Omega_E(f_E)^2$

$$\begin{aligned}
 \Omega_E(f_E)^2 &= ((\sum_{v \in V_e} (d_v \|\mathbf{f}_{ED(v)}\|_\rho)^1)^1)^2 \\
 &= \min_{\boldsymbol{\gamma} \in \Delta_{d,r}} \left(\sqrt{(\sum_{v \in V_e} \frac{d_v^2 \|\mathbf{f}_{ED(v)}\|_\rho^2}{\gamma_v})} \right)^2 \\
 &= \min_{\boldsymbol{\gamma} \in \Delta_{d,r}} \left(\sum_{v \in V_e} \frac{d_v^2 \|\mathbf{f}_{ED(v)}\|_\rho^2}{\gamma_v} \right) \\
 &\text{(where } p = 2 \text{ i.e. } r = 1 \text{ and } d = |v| \text{ i.e. } \Delta_{|v|,1} = \{\boldsymbol{\eta} \in R^{|v|}, \sum_{v \in V} \eta_v = 1, \eta_v \geq 0\}) \\
 &= \min_{\boldsymbol{\gamma} \in \Delta_{d,r}} \sum_{v \in V} \frac{d_v^2}{\gamma_v} \left((\sum_{w \in D(v)} f_{Ew}^\rho)^\frac{1}{\rho} \right)^2
 \end{aligned}$$

By applying the lemma again

$$\begin{aligned}
 &= \min_{\boldsymbol{\gamma} \in \Delta_{d,r}} \sum_{v \in V} \frac{d_v^2}{\gamma_v} \min_{\lambda_v \in \Delta_{|D(v)|,r}} \left(\sqrt{\sum_{w \in D(v)} \frac{\|f_{Ew}\|_2^2}{\lambda_{wv}}} \right)^2 \\
 &\text{where } \frac{2}{p} = \rho \text{ i.e. } \rho = \frac{2}{p} \text{ and } r = \frac{1}{p-1} = \frac{\rho}{2-\rho} = \hat{\rho} \text{ and } \Delta_{|D(v)|,r} = \{\boldsymbol{\lambda} \in \mathbb{R}^{|D(v)|}, \sum_{w \in D(v)} \lambda_{wv}^r = 1\} \\
 &= \min_{\boldsymbol{\gamma} \in \Delta_{|v|,1}} \min_{\lambda_v \in \Delta_{|D(v)|,\hat{\rho}}} \sum_{v \in V_e} \sum_{w \in D(v)} \frac{\|f_{Ew}\|_2^2}{\lambda_{wv} \frac{\gamma_v}{d_v^2}} \\
 &= \min_{\boldsymbol{\gamma} \in \Delta_{|v|,1}} \min_{\lambda_v \in \Delta_{|D(v)|,\hat{\rho}}} \sum_{v \in V_e} \sum_{w \in A(w)} \frac{\|f_{Ew}\|_2^2}{\lambda_{wv} \frac{\gamma_v}{d_v^2}}
 \end{aligned}$$

$$\begin{aligned}
&= \min_{\gamma \in \Delta_{|v|,1}} \min_{\lambda_v \in \Delta_{|D(v)|,\hat{\rho}} \forall v \in V_e} \sum_{w \in V_e} \|f_{Ew}\|_2^2 \sum_{v \in A(w)} \frac{1}{\lambda_{wv} \frac{\gamma_v}{d_v^2}} \\
&= \min_{\gamma \in \Delta_{|v|,1}} \min_{\lambda_v \in \Delta_{|D(v)|,\hat{\rho}} \forall v \in V_e} \sum_{w \in V_e} \delta_w(\gamma, \lambda)^{-1} \|f_{Ew}\|_2^2 \\
&\text{where } \delta_w(\gamma, \lambda)^{-1} = \sum_{v \in A(w)} \frac{d_v^2}{\lambda_{wv} \gamma_v}
\end{aligned}$$

Now we derive the dual of equation 3.10 before giving a sufficiency condition for the reduced solution to be the final solution.

By applying the variational characterization of the regularizer, the lagrangian for equation 3.10 can be written as

$$\begin{aligned}
&\frac{1}{2} \sum_{w \in \mathcal{V}_{\mathbf{E}}} \delta_w^{-1}(\gamma, \lambda) \|f_{Ew}\|_2^2 + \frac{1}{2} \|\mathbf{f}_{\mathbf{T}}\|_2^2 + \frac{C}{n} \sum_{i=1}^m \xi_i - \sum_i \theta_i \xi_i \\
&- \sum_{i, Y \neq Y_i} \alpha_{iY} \left[\sum_{v \in \mathcal{V}_{\mathbf{E}}} \langle f_{Ev}, \psi_{Evi}^\delta(Y) \rangle + \langle f_T, \psi_{Ti}^\delta(Y) \rangle + \frac{\xi_i}{\Delta(Y_i, Y)} - 1 \right]
\end{aligned}$$

where α, θ are lagrange multipliers. The partial derivative of Lagrangian with respect to \mathbf{f} alone is

$$\bar{\mathbf{f}}_{\mathbf{E}} + \mathbf{f}_{\mathbf{T}} - \sum_{i, Y \neq Y_i} \alpha_{iY} \psi_{\mathbf{E}i}^\delta(Y) - \sum_{i, Y \neq Y_i} \alpha_{iY} \psi_{\mathbf{T}i}^\delta(Y)$$

where $\bar{\mathbf{f}}_{\mathbf{E}}$ has elements $f_{Ew}^- = \delta_w^{-1}(\gamma, \lambda) f_{Ew}$. By KKT conditions, equating the partial derivative to zero yeilds,

$$\bar{\mathbf{f}}_{\mathbf{E}} + \mathbf{f}_{\mathbf{T}} = \sum_{i, Y \neq Y_i} \alpha_{iY} \psi_{\mathbf{E}i}^\delta(Y) + \sum_{i, Y \neq Y_i} \alpha_{iY} \psi_{\mathbf{T}i}^\delta(Y)$$

From our definition, it is easy to observe that

$$\bar{\mathbf{f}}_{\mathbf{E}} = \sum_{i, Y \neq Y_i} \alpha_{iY} \psi_{\mathbf{E}i}^\delta(Y)$$

and

$$\mathbf{f}_{\mathbf{T}} = \sum_{i, Y \neq Y_i} \alpha_{iY} \psi_{\mathbf{T}i}^\delta(Y)$$

There fore, each element of $\mathbf{f}_{\mathbf{E}}$ is

$$f_{Ew} = \delta_w(\gamma, \lambda) f_{Ew}^- = \delta_w(\gamma, \lambda) \sum_{i, Y \neq Y_i} \alpha_{iY} \psi_{Ewi}^\delta(Y)$$

Partial derivative with respect to ξ gives additional constraint,

$$\forall i : m \sum_{Y \neq Y_i} \frac{\alpha_{iY}}{\Delta(Y, Y_i)} \leq C$$

Putting these back to the objective function gives the following partial dual problem,

$$\min_{\gamma \in \Delta_{|\mathcal{V}_{\mathbf{E}}|, 1}} \min_{\lambda_v \in \Delta_{|D(v)|, \hat{\rho}} \forall v \in \mathcal{V}_{\mathbf{E}}} \max_{\alpha \in \tau(\mathcal{Y}, C)} G(\gamma, \lambda, \alpha) \quad (\text{A.1})$$

where

$$G(\gamma, \lambda, \alpha) = \sum_{i, Y \neq Y_i} \alpha_{iY} - \frac{1}{2} \alpha^\top \left(\sum_{w \in \mathcal{V}_{\mathbf{E}}} \delta_w(\gamma, \lambda) \kappa_{\mathbf{E}w} \right) \alpha - \frac{1}{2} \alpha^\top \kappa_{\mathbf{T}} \alpha$$

and $\tau(\mathcal{Y}, C) = \{\alpha \in \mathbb{R}^{m(n^l-1)} \mid \alpha_{i,Y} \geq 0, m \sum_{Y \neq Y_i} \frac{\alpha_{iY}}{\Delta(Y, Y_i)} \leq C, \forall i, Y\}$, which is same as that given in equation (3.12).

We consider this partial dual problem as a new primal. In the new formulation, let the primal solution, $\min_{\gamma, \lambda_v} \max_{\alpha} G(\gamma, \lambda, \alpha)$, be p^* and the dual solution, $\max_{\alpha} \min_{\gamma, \lambda_v} G(\gamma, \lambda, \alpha)$, be d^* . The duality gap $p^* - d^*$ can be written as

$$\min_{\hat{\gamma} \in \Delta_{|\mathcal{V}_{\mathbf{E}}|, 1} \hat{\lambda}_v \in \Delta_{|D(v)|, \hat{\rho}} \forall v \in \mathcal{V}_{\mathbf{E}}} \min_{\hat{\alpha}} G(\hat{\gamma}, \hat{\lambda}, \alpha) - \max_{\hat{\alpha}} G(\gamma, \lambda, \hat{\alpha}) \quad (\text{A.2})$$

where $\alpha = \argmax_{\hat{\alpha}} G(\gamma, \lambda, \hat{\alpha})$ and $(\gamma, \lambda) = G(\hat{\gamma}, \hat{\lambda}, \alpha)$
since dual \leq primal,

$$\max_{\hat{\alpha}} \min_{\hat{\gamma}, \hat{\lambda}_v} G(\hat{\gamma}, \hat{\lambda}, \hat{\alpha}) \leq \frac{1}{2} \Omega_E(\mathbf{f}_{\mathbf{E}})^2 + \frac{1}{2} \Omega_T(\mathbf{f}_{\mathbf{T}})^2 + \frac{C}{m} \sum_i \xi_i \quad (\text{A.3})$$

Combining equation (A.2) and equation (A.3), we get,

$$\begin{aligned} & \min_{\hat{\gamma} \in \Delta_{|\mathcal{V}_{\mathbf{E}}|, 1} \hat{\lambda}_v \in \Delta_{|D(v)|, \hat{\rho}} \forall v \in \mathcal{V}_{\mathbf{E}}} \min_{\hat{\alpha}} G(\hat{\gamma}, \hat{\lambda}, \alpha) - \max_{\hat{\alpha}} G(\gamma, \lambda, \hat{\alpha}) \\ & \geq \min_{\hat{\gamma} \in \Delta_{|\mathcal{V}_{\mathbf{E}}|, 1} \hat{\lambda}_v \in \Delta_{|D(v)|, \hat{\rho}} \forall v \in \mathcal{V}_{\mathbf{E}}} \min_{\hat{\alpha}} \left(\sum_{i, Y \neq Y_i} \alpha_{iY} - \frac{1}{2} \sum_{w \in \mathcal{V}_{\mathbf{E}}} \delta_w(\hat{\gamma}, \hat{\lambda}) \alpha^\top \kappa_{\mathbf{E}w} \alpha - \frac{1}{2} \alpha^\top \kappa_{\mathbf{T}} \alpha \right) \\ & \quad - \left(\frac{1}{2} \Omega_E(\mathbf{f}_{\mathbf{E}})^2 + \frac{1}{2} \Omega_T(\mathbf{f}_{\mathbf{T}})^2 + \frac{C}{m} \sum_i \xi_i \right) \end{aligned}$$

which is equivalent to,

$$\begin{aligned} & \min_{\hat{\gamma} \in \Delta_{|\mathcal{V}_{\mathbf{E}}|, 1} \hat{\lambda}_v \in \Delta_{|D(v)|, \hat{\rho}} \forall v \in \mathcal{V}_{\mathbf{E}}} \min_{\hat{\alpha}} G(\hat{\gamma}, \hat{\lambda}, \alpha) - \max_{\hat{\alpha}} G(\gamma, \lambda, \hat{\alpha}) \\ & \geq \min_{\hat{\gamma} \in \Delta_{|\mathcal{V}_{\mathbf{E}}|, 1} \hat{\lambda}_v \in \Delta_{|D(v)|, \hat{\rho}} \forall v \in \mathcal{V}_{\mathbf{E}}} \min_{\hat{\alpha}} - \left(\frac{1}{2} \sum_{w \in \mathcal{V}_{\mathbf{E}}} \delta_w(\hat{\gamma}, \hat{\lambda}) \alpha^\top \kappa_{\mathbf{E}w} \alpha - \frac{1}{2} (\Omega_E(\mathbf{f}_{\mathbf{E}})^2 + \Omega_T(\mathbf{f}_{\mathbf{T}})^2) \right) - e \end{aligned}$$

where $e = \Omega_E(\mathbf{f}_E)^2 + \Omega_T(\mathbf{f}_T)^2 + \frac{C}{m} \sum_i \xi_i + \frac{1}{2} \boldsymbol{\alpha}^\top \boldsymbol{\kappa}_T \boldsymbol{\alpha} - \sum_{i, Y \neq Y_i} \alpha_{iY}$

Therefore, the sufficiency condition for the active set to get a duality gap less than ϵ where $\epsilon \geq 0$ is

$$\max_{\hat{\gamma} \in \Delta_{|\mathcal{V}_E|, 1}} \max_{\hat{\lambda}_v \in \Delta_{|D(v)|, \hat{\rho}} \forall v \in \mathcal{V}_E} \left(\sum_{w \in \mathcal{V}_E} \delta_w(\hat{\gamma}, \hat{\lambda}) \boldsymbol{\alpha}_{\mathcal{W}}^\top \boldsymbol{\kappa}_{Ew} \boldsymbol{\alpha}_{\mathcal{W}} \right) \leq \Omega_E(\mathbf{f}_{E\mathcal{W}})^2 + \Omega_T(\mathbf{f}_{T\mathcal{W}})^2 + 2(\epsilon - e_{\mathcal{W}}) \quad (\text{A.4})$$

where $e_{\mathcal{W}}$ is the gap associated with the computation of $\alpha_{\mathcal{W}}$. The lagrange dual of

$\max_{\hat{\gamma} \in \Delta_{|\mathcal{V}_E|, 1}} \max_{\hat{\lambda}_v \in \Delta_{|D(v)|, \hat{\rho}} \forall v \in \mathcal{V}_E} \left(\sum_{w \in \mathcal{V}_E} \delta_w(\hat{\gamma}, \hat{\lambda}) \boldsymbol{\alpha}_{\mathcal{W}}^\top \boldsymbol{\kappa}_{Ew} \boldsymbol{\alpha}_{\mathcal{W}} \right)$ with respect to γ (Jawanpuria et al. 2011) is given by

$$\max_{\hat{\lambda}_v \in \Delta_{|D(v)|, \hat{\rho}} \forall v \in \mathcal{V}_E} \min_{k \in L} \max_{v \in \mathcal{V}} \sum_{w \in D(v)} \frac{k_{vw}^2 \lambda_{vw} \boldsymbol{\alpha}_{\mathcal{W}}^\top \boldsymbol{\kappa}_{Ew} \boldsymbol{\alpha}_{\mathcal{W}}}{d_v^2},$$

where $L = \{k \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|} | k \geq 0, \sum_{v \in A(w)} k_{vw} = 1, \sum_{v \in A(w)^c} k_{vw} = 0 \forall w \in \mathcal{V}\}$. By minmax theorem (Sion 1958), this is less than $\min_{k \in L} \max_{v \in \mathcal{V}} \max_{\hat{\lambda}_v \in \Delta_{|D(v)|, \hat{\rho}}} \sum_{w \in D(v)} \frac{k_{vw}^2 \lambda_{vw} \boldsymbol{\alpha}_{\mathcal{W}}^\top \boldsymbol{\kappa}_{Ew} \boldsymbol{\alpha}_{\mathcal{W}}}{d_v^2}$.

Applying Lemma 26 in (Micchelli & Pontil 2005), we get,

$$\min_{k \in L} \max_{v \in \mathcal{V}} d_v^{-2} \left(\sum_{w \in D(v)} (k_{vw}^2 \boldsymbol{\alpha}_{\mathcal{W}}^\top \boldsymbol{\kappa}_{Ew} \boldsymbol{\alpha}_{\mathcal{W}})^{\hat{\rho}} \right)^{\frac{1}{\hat{\rho}}} \quad (\text{A.5})$$

Since $\mathcal{W} = \text{hull}(\mathcal{W})$, for all $w \in \mathcal{W}$, k is taken to be the optimal k obtained by solving the small problem in equation (3.11) and for all $w \in \mathcal{W}^c$, $k_{vw} = d_v \left(\sum_{u \in A(v) \cap \mathcal{W}^c} d_u \right)^{-1}$.

With this choice, the upper bound can be written as

$$\max \left\{ \Omega_E(\mathbf{f}_{E\mathcal{W}})^2, \max_{u \in \mathcal{W}^c} \left(\sum_{w \in D(u)} \left(\frac{\boldsymbol{\alpha}_{\mathcal{W}}^\top \boldsymbol{\kappa}_{Ew} \boldsymbol{\alpha}_{\mathcal{W}}}{(\sum_{v \in A(w) \cap \mathcal{W}^c} d_v)^2} d_v \right)^{\hat{\rho}} \right)^{\frac{1}{\hat{\rho}}} \right\}$$

Since $\mathcal{W} = \text{hull}(\mathcal{W})$ and using the condition $\sum_{v \in A(w) \cap \mathcal{W}^c} d_v \geq \sum_{v \in A(w) \cap D(u)} d_v$, we get the upper bound of LHS as,

$$\max \left\{ \Omega_E(\mathbf{f}_{E\mathcal{W}})^2, \max_{u \in \text{sources}(\mathcal{W}^c)} \left(\sum_{w \in D(u)} \left(\frac{\boldsymbol{\alpha}_{\mathcal{W}}^\top \boldsymbol{\kappa}_{Ew} \boldsymbol{\alpha}_{\mathcal{W}}}{(\sum_{v \in A(w) \cap D(u)} d_v)^2} d_v \right)^{\hat{\rho}} \right)^{\frac{1}{\hat{\rho}}} \right\}$$

which is

$$\leq \max_{u \in \text{sources}(\mathcal{W}^c)} \left(\sum_{w \in D(u)} \left(\frac{\boldsymbol{\alpha}_{\mathcal{W}}^\top \boldsymbol{\kappa}_{Ew} \boldsymbol{\alpha}_{\mathcal{W}}}{(\sum_{v \in A(w) \cap D(u)} d_v)^2} d_v \right)^{\hat{\rho}} \right)^{\frac{1}{\hat{\rho}}}$$

Since for any $\hat{\beta}$, $\|\hat{\beta}\|_{\hat{\rho}} \leq \|\hat{\beta}\|_1$, the upper bound of LHS can be written as

$$\max_{u \in \text{sources}(\mathcal{W}^c)} \sum_{w \in D(u)} \frac{\boldsymbol{\alpha}_{\mathcal{W}}^\top \boldsymbol{\kappa}_{Ew} \boldsymbol{\alpha}_{\mathcal{W}}}{(\sum_{v \in A(w) \cap D(u)} d_v)^2}$$

In a sequence labeling setup, this is equivalent to

$$\begin{aligned}
& \max_{u \in \text{sources}(\mathcal{W}^c)} \sum_{i, Y \neq Y_i} \sum_{j, Y' \neq Y_j} \alpha_{\mathcal{W}iY}^\top \sum_{p=1}^{l_i} \sum_{q=1}^{l_j} \\
& \left(\sum_{w \in D(u)} \frac{\kappa_{Ew}(\mathbf{x}_i^p, \mathbf{x}_j^q) (\Lambda(y_i^p, y_j^q) + \Lambda(y^p, y'^q) - \Lambda(y_i^p, y'^q) - \Lambda(y^p, y_j^q))}{(\sum_{v \in A(w) \cap D(u)} d_v)^2} \right) \alpha_{\mathcal{W}jY'} \\
& \leq \max_{u \in \text{sources}(\mathcal{W}^c)} \sum_{i, Y \neq Y_i} \sum_{j, Y' \neq Y_j} \alpha_{\mathcal{W}iY}^\top \sum_{p=1}^{l_i} \sum_{q=1}^{l_j} 2 \sum_{w \in D(u)} \frac{\kappa_{Ew}(\mathbf{x}_i^p, \mathbf{x}_j^q)}{(\sum_{v \in A(w) \cap D(u)} d_v)^2} \alpha_{\mathcal{W}jY'}
\end{aligned}$$

d_v is given values 1 for the top node and β^i for nodes at level i , where β is a constant (Jawanpuria et al. 2011). There fore,

$$\sum_{v \in A(w) \cap D(u)} d_v = \beta^u + (w - u)\beta^{u+1} + \binom{w-u}{2}\beta^{u+2} + \binom{w-u}{3}\beta^{u+3} + \dots + \binom{w-u}{w-u}\beta^w = \beta^u(1 + \beta)^{w-u}.$$

The upper bound can now be written as

$$\leq \max_{u \in \text{sources}(\mathcal{W}^c)} \sum_{i, Y \neq Y_i} \sum_{j, Y' \neq Y_j} \alpha_{\mathcal{W}iY}^\top \sum_{p=1}^{l_i} \sum_{q=1}^{l_j} 2 \sum_{w \in D(u)} \frac{\kappa_{Ew}(\mathbf{x}_i^p, \mathbf{x}_j^q)}{\beta^{2u}((1 + \beta)^{w-u})^2} \alpha_{\mathcal{W}jY'} \quad (\text{A.6})$$

Now,

$$\begin{aligned}
& \sum_{w \in D(u)} \frac{\kappa_{Ew}(\mathbf{x}_i^p, \mathbf{x}_j^q)}{\beta^{2u}((1 + \beta)^{w-u})^2} = \sum_{w \in D(u)} \frac{\langle \psi_{Ew}(\mathbf{x}_i^p), \psi_{Ew}(\mathbf{x}_j^q) \rangle}{\beta^{2u}((1 + \beta)^{w-u})^2} \\
& = \sum_{w \in D(u)} \frac{\langle \prod_{k \in w} \psi_{Ek}(\mathbf{x}_i^p), \prod_{k \in w} \psi_{Ek}(\mathbf{x}_j^q) \rangle}{\beta^{2u}((1 + \beta)^{w-u})^2} \\
& = \sum_{w \in D(u)} \langle \prod_{k \in u} \frac{\psi_{Ek}(\mathbf{x}_i^p)}{\beta} \prod_{k \notin u, k \in w} \frac{\psi_{Ek}(\mathbf{x}_i^p)}{1 + \beta}, \prod_{k \in u} \frac{\psi_{Ek}(\mathbf{x}_j^q)}{\beta} \prod_{k \notin u, k \in w} \frac{\psi_{Ek}(\mathbf{x}_j^q)}{1 + \beta} \rangle \\
& \leq \sum_{w \in D(u)} \langle \prod_{k \in u} \frac{\psi_{Ek}(\mathbf{x}_i^p)}{\beta}, \prod_{k \in u} \frac{\psi_{Ek}(\mathbf{x}_j^q)}{\beta} \rangle \langle \prod_{k \notin u, k \in w} \frac{\psi_{Ek}(\mathbf{x}_i^p)}{1 + \beta}, \prod_{k \notin u, k \in w} \frac{\psi_{Ek}(\mathbf{x}_j^q)}{1 + \beta} \rangle \\
& = \prod_{k \in u} \frac{\psi_{Ek}(\mathbf{x}_i^p) \psi_{Ek}(\mathbf{x}_j^q)}{\beta^2} \sum_{w \in D(u)} \langle \prod_{k \notin u, k \in w} \frac{\psi_{Ek}(\mathbf{x}_i^p)}{1 + \beta}, \prod_{k \notin u, k \in w} \frac{\psi_{Ek}(\mathbf{x}_j^q)}{1 + \beta} \rangle
\end{aligned}$$

Applying the kernel trick (Jawanpuria et al. 2011), it can be written as

$$\prod_{k \in u} \frac{\psi_{Ek}(\mathbf{x}_i^p) \psi_{Ek}(\mathbf{x}_j^q)}{\beta^2} \prod_{k \notin u} \left(1 + \frac{\psi_{Ek}(\mathbf{x}_i^p) \psi_{Ek}(\mathbf{x}_j^q)}{(1 + \beta)^2} \right)$$

Therefore, the upper bound for the LHS of sufficiency condition becomes,

$$\max_{u \in \text{sources}(\mathcal{W}^c)} \sum_{i, Y \neq Y_i} \sum_{j, Y' \neq Y_j} \alpha_{\mathcal{W}iY}^\top \sum_{p=1}^{l_i} \sum_{q=1}^{l_j} 2 \left(\prod_{k \in u} \frac{\psi_{Ek}(\mathbf{x}_i^p) \psi_{Ek}(\mathbf{x}_j^q)}{\beta^2} \right) \left(\prod_{k \notin u} \left(1 + \frac{\psi_{Ek}(\mathbf{x}_i^p) \psi_{Ek}(\mathbf{x}_j^q)}{(1 + \beta)^2} \right) \right) \alpha_{\mathcal{W}jY'} \quad (\text{A.7})$$

The sufficiency condition can thus be written as,

$$\begin{aligned} \max_{u \in \text{sources}(\mathcal{W}^c)} \sum_{i, Y \neq Y_i} \sum_{j, Y' \neq Y_j} \alpha_{\mathcal{W}iY}^\top \sum_{p=1}^{l_i} \sum_{q=1}^{l_j} 2 \cdot \left(\prod_{k \in u} \frac{\psi_{Ek}(\mathbf{x}_i^p) \psi_{Ek}(\mathbf{x}_j^q)}{\beta^2} \right) \left(\prod_{k \notin u} \left(1 + \frac{\psi_{Ek}(\mathbf{x}_i^p) \psi_{Ek}(\mathbf{x}_j^q)}{(1+\beta)^2} \right) \right) \alpha_{\mathcal{W}jY'} \\ \leq \Omega_E(\mathbf{f}_{\mathbf{E}\mathcal{W}})^2 + \Omega_T(\mathbf{f}_{\mathbf{T}\mathcal{W}})^2 + 2(\epsilon - e_{\mathcal{W}}) \end{aligned} \quad (\text{A.8})$$

which is same as that given in equation (3.13)

A.2 Solution to the Reduced Problem

We consider the partial dual of equation 3.10, as derived in equation (A.1), as new primal and derive the dual of the new primal problem here.

$$\max_{\alpha \in \tau(\mathcal{Y}, C)} \min_{\gamma \in \Delta_{|\mathcal{V}_{\mathbf{E}}|, 1}} \min_{\lambda_v \in \Delta_{|D(v)|, \hat{\rho}} \forall v \in \mathcal{V}_{\mathbf{E}}} - \frac{1}{2} \alpha^\top \left(\sum_{w \in \mathcal{V}_{\mathbf{E}}} \delta_w(\gamma, \lambda) \kappa_{\mathbf{E}w} \right) \alpha - \frac{1}{2} \alpha^\top \kappa_{\mathbf{T}} \alpha + \sum_{i, Y \neq Y_i} \alpha_{iY}$$

which is equivalent to

$$\max_{\alpha \in \tau(\mathcal{Y}, C)} \sum_{i, Y \neq Y_i} \alpha_{iY} - \frac{1}{2} \alpha^\top \kappa_{\mathbf{T}} \alpha - \max_{\gamma \in \Delta_{|\mathcal{V}_{\mathbf{E}}|, 1}} \max_{\lambda_v \in \Delta_{|D(v)|, \hat{\rho}} \forall v \in \mathcal{V}_{\mathbf{E}}} \frac{1}{2} \alpha^\top \left(\sum_{w \in \mathcal{V}_{\mathbf{E}}} \delta_w(\gamma, \lambda) \kappa_{\mathbf{E}w} \right) \alpha \quad (\text{A.9})$$

In equation (A.5), we got,

$$\max_{\gamma \in \Delta_{|\mathcal{V}_{\mathbf{E}}|, 1}} \max_{\lambda_v \in \Delta_{|D(v)|, \hat{\rho}} \forall v \in \mathcal{V}_{\mathbf{E}}} \alpha^\top \left(\sum_{w \in \mathcal{V}_{\mathbf{E}}} \delta_w(\gamma, \lambda) \kappa_{\mathbf{E}w} \right) \alpha = \min_{k \in L} \max_{v \in \mathcal{V}} d_v^{-2} \left(\sum_{w \in D(v)} (k_{vw}^2 \alpha^\top \kappa_{\mathbf{E}w} \alpha)^{\hat{\rho}} \right)^{\frac{1}{\hat{\rho}}}$$

Dual of this term (Jawanpuria et al. 2011) is given as,

$$\max_{\eta \in \Delta_{|D(v)|, 1}} \left(\sum_{w \in \mathcal{V}} \zeta_w(\eta) (\alpha^\top \kappa_{\mathbf{E}w} \alpha)^{\hat{\rho}} \right)^{\frac{1}{\hat{\rho}}}$$

where $\zeta_w(\eta) = \left(\sum_{v \in A(w)} d_v^\rho \eta_v^{1-\rho} \right)^{\frac{1}{1-\rho}}$. Substituting in equation (A.9), we get the final dual which is,

$$\max_{\alpha \in \tau(\mathcal{Y}, C)} \sum_{i, Y \neq Y_i} \alpha_{iY} - \frac{1}{2} \alpha^\top \kappa_{\mathbf{T}} \alpha - \frac{1}{2} \max_{\eta \in \Delta_{|D(v)|, 1}} \left(\sum_{w \in \mathcal{V}} \zeta_w(\eta) (\alpha^\top \kappa_{\mathbf{E}w} \alpha)^{\hat{\rho}} \right)^{\frac{1}{\hat{\rho}}}$$

where $\zeta_w(\eta) = \left(\sum_{v \in A(w)} d_v^\rho \eta_v^{1-\rho} \right)^{\frac{1}{1-\rho}}$. Which is same as that given in equation (3.14) and equation (3.15). The solution to the dual problem in equation (3.14) with \mathcal{V} restricted to \mathcal{W} gives the solution to the restricted primal problem given in equation (3.11).

If $\bar{\alpha}$ is the optimal solution to equation (3.15) with some $\boldsymbol{\eta}$, then the i^{th} sub-gradient for the objective function $g(\boldsymbol{\eta})$ is given by

$$(\nabla g(\boldsymbol{\eta}))_i = -\frac{d_i^\rho \eta_i^{-\rho}}{2\hat{\rho}} \left(\sum_{w \in \mathcal{V}_{\mathbf{E}}} \zeta_w(\boldsymbol{\eta}) (\bar{\alpha}^\top \boldsymbol{\kappa}_{\mathbf{E}w} \bar{\alpha})^{\hat{\rho}} \right)^{\frac{1}{\hat{\rho}}-1} \left(\sum_{w \in D(i)} \zeta_w(\boldsymbol{\eta})^\rho (\bar{\alpha}^\top \boldsymbol{\kappa}_{\mathbf{E}w} \bar{\alpha})^{\hat{\rho}} \right)$$

where

$$\bar{\alpha}^\top \boldsymbol{\kappa}_{\mathbf{E}w} \bar{\alpha} = \sum_{i, Y \neq Y_i} \sum_{j, Y' \neq Y_j} \alpha_{iY} \alpha_{jY'} \sum_{p=1}^{l_i} \sum_{q=1}^{l_j} \kappa_{Ew}(\mathbf{x}_i^p, \mathbf{x}_j^q) (\Lambda(y_i^p, y_j^q) + \Lambda(y^p, y'^q) - \Lambda(y_i^p, y'^q) - \Lambda(y^p, y_j^q)).$$

In each iteration of the mirror descend algorithm, $\bar{\alpha}$ is obtained by using a cutting plane algorithm to solve equation (3.15), the parameters of equation (3.14) are updated with the sub-gradient. The algorithm terminates when the sufficiency condition given in equation (3.13) is satisfied.

A.3 Kernels in StructHKL

The kernel functions $\boldsymbol{\kappa}_{\mathbf{E}w}$ and $\boldsymbol{\kappa}_{\mathbf{T}}$ are those corresponding to emission kernel at node w and the transition respectively. We briefly discuss this in the following paragraph.

The kernel $\boldsymbol{\kappa}_{\mathbf{E}w}$ stands for the inner product of the feature node values corresponding to two different input-output pairs of sequences (Tsochantaridis 2006). The inner product with respect to node w of examples i, j and their corresponding sample sequences Y and Y' is given by,

$$\begin{aligned} \langle \psi_{Ewi}^\delta(Y), \psi_{Ewj}^\delta(Y') \rangle &= \langle (\psi_{Ew}(X_i, Y_i) - \psi_{Ew}(X_i, Y)), (\psi_{Ew}(X_j, Y_j) - \psi_{Ew}(X_j, Y')) \rangle \\ &= \langle \psi_{Ew}(X_i, Y_i), \psi_{Ew}(X_j, Y_j) \rangle + \langle \psi_{Ew}(X_i, Y), \psi_{Ew}(X_j, Y') \rangle \\ &\quad - \langle \psi_{Ew}(X_i, Y_i), \psi_{Ew}(X_j, Y') \rangle - \langle \psi_{Ew}(X_i, Y), \psi_{Ew}(X_j, Y_j) \rangle. \end{aligned}$$

We define a kernel for each of these inner products and since sum of these kernels is a kernel, $\boldsymbol{\kappa}_{\mathbf{E}w}$ for two input output pairs can be split into $\kappa_{Ew}((X_i, Y_i), (X_j, Y_j)) + \kappa_{Ew}((X_i, Y), (X_j, Y')) - \kappa_{Ew}((X_i, Y_i), (X_j, Y')) - \kappa_{Ew}((X_i, Y), (X_j, Y_j))$.

Since, the input and output are sequences, each of these kernels can be defined such as

$$\kappa_{Ew}((X_i, Y_i), (X_j, Y_j)) = \sum_{p=1}^{l_i} \sum_{q=1}^{l_j} \kappa_{Ew}((\mathbf{x}_i^p, y_i^p), (\mathbf{x}_j^q, y_j^q)),$$

where l_i is the length of the i^{th} sequence, \mathbf{x}_i^p is the p^{th} input vector of X_i and y_i^p is the p^{th} output label of Y_i . The kernel corresponding to a time step in a sequence can be defined as

$$\kappa_{Ew}((\mathbf{x}_i^p, y_i^p), (\mathbf{x}_j^q, y_j^q)) = \kappa_{Ew}(\mathbf{x}_i^p, \mathbf{x}_j^q) \Lambda(y_i^p, y_j^q),$$

where $\Lambda(y_i^p, y_j^q) = 1$ if $y_i^p = y_j^q$; 0 otherwise. Therefore, $\kappa_{\mathbf{E}w}$ for i, j, Y and Y' is

$$\sum_{p=1}^{l_i} \sum_{q=1}^{l_j} \kappa_{\mathbf{E}w}(\mathbf{x}_i^p, \mathbf{x}_j^q) (\Lambda(y_i^p, y_j^q) + \Lambda(y_i^p, y_j'^q) - \Lambda(y_i^p, y_j'^q) - \Lambda(y_i^p, y_j^q)) ,$$

where $\kappa_{\mathbf{E}w}(\mathbf{x}_i^p, \mathbf{x}_j^q) = \langle \psi_w(\mathbf{x}_i^p), \psi_w(\mathbf{x}_j^q) \rangle$. Further as described in (Jawanpuria et al. 2011), for any sublattice \mathcal{U} formed by the descendants of a node, $\sum_{v \in \mathcal{U}} \kappa_v(\mathbf{x}_i^p, \mathbf{x}_j^q) = \prod_{k \in B \cap \mathcal{U}} (1 + \psi_k(\mathbf{x}_i^p) \psi_k(\mathbf{x}_j^q))$, where B is the set of basic features. Similarly, $\kappa_{\mathbf{T}}$ over two sequences Y_i and Y_j is defined as $\sum_{p=1}^{l_i-1} \sum_{q=1}^{l_j-1} \Lambda(y_i^p, y_j^q) \Lambda(y_i^{p+1}, y_j^{q+1})$.

A.4 Cutting Plane Algorithm

Our objective is to incrementally find a solution to equation (3.14) using mirror descent. In each iteration of the mirror descent algorithm, we find an optimal value of α using cutting plane algorithm, then find the sub-gradient to $g(\eta)$ using the value of α and update the η value. The objective to the cutting plane algorithm is equation (3.15). ie,

$$\max_{\alpha \in \tau(\mathcal{Y}, \mathcal{C})} \sum_{i, Y \neq Y_i} \alpha_{iY} - \frac{1}{2} \alpha^\top \kappa_{\mathbf{T}} \alpha - \frac{1}{2} \left(\sum_{w \in \mathcal{V}} \left(\sum_{v \in A(w)} d_v^\rho \eta_v^{1-\rho} \right)^{\frac{1}{1-\rho}} (\alpha^\top \kappa_{\mathbf{E}w} \alpha)^{\hat{\rho}} \right)^{\frac{1}{\hat{\rho}}} \quad (\text{A.10})$$

Merging the value of $\zeta_w(\eta) = \left(\sum_{v \in A(w)} d_v^\rho \eta_v^{1-\rho} \right)^{\frac{1}{1-\rho}}$ into kernel, we can write the above equation as,

$$\max_{\alpha \in \tau(\mathcal{Y}, \mathcal{C})} \sum_{i, Y \neq Y_i} \alpha_{iY} - \frac{1}{2} \sum_{i, Y} \sum_{j, Y'} \alpha_{iY} \alpha_{jY'} \kappa_{T i Y j Y'} - \frac{1}{2} \left(\sum_{w \in \mathcal{V}} \left(\sum_{i, Y} \sum_{j, Y'} \alpha_{iY} \alpha_{jY'} \kappa'_{E w i Y j Y'} \right)^{\hat{\rho}} \right)^{\frac{1}{\hat{\rho}}} \quad (\text{A.11})$$

To implement cutting plane algorithm, we define a cost function (the amount by which the margin is violated by a constraint). This is derived from the constraints of equation (3.10). It is defined as for each i ,

$$H(Y) \equiv \left[1 - \sum_{j, Y' \in S_j \setminus Y_j} \alpha_{j, Y'} \left(\kappa_{T i Y j Y'} + \sum_{w \in \mathcal{W}} \delta_w(\gamma, \lambda) \kappa_{E i Y j Y'} \right) \right] \Delta(Y_i, Y) \quad (\text{A.12})$$

The cutting plane algorithm is outlined in Fig. 3.4. (Tsochantaridis et al. 2004) suggests to use dynamic programming to compute line 5 in the algorithm. We use Viterbi algorithm to solve this. Line 9 needs to solve the dual objective given in equation equation (A.11) with the constraint set restricted to S . Cutting plane algorithm ends with a polynomial number of constraints from an exponential number of constraints.

Bibliography

- Agrawal, R. & Srikant, R. (1994), Fast algorithms for mining association rules, *in* ‘Proc. of 20th Intl. Conf. on VLDB’, pp. 487–499.
- Almuallim, H. & Dietterich, T. (1991), ‘Learning with many irrelevant features’, pp. 547–552.
- Amit, Y., Fink, M., Srebro, N. & Ullman, S. (2007), Uncovering shared structures in multiclass classification, *in* ‘Proceedings of the 24th international conference on Machine learning’, ICML ’07, ACM, New York, NY, USA, pp. 17–24.
URL: <http://doi.acm.org/10.1145/1273496.1273499>
- Argyriou, A., Evgeniou, T. & Pontil, M. (2007), Multi-task feature learning, *in* ‘Advances in Neural Information Processing Systems 19’, MIT Press.
- Arun, R., Suresh, V., Veni Madhavan, C. & Narasimha Murthy, M. (2010), On finding the natural number of topics with latent dirichlet allocation: Some observations, *in* ‘Advances in Knowledge Discovery and Data Mining’, Vol. 6118 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 391–402.
- Bach, F. (2009), ‘High-dimensional non-linear variable selection through hierarchical kernel learning’, *Technical report, INRIA, France*.
- Belkin, M. & Niyogi, P. (2002), ‘Laplacian eigenmaps for dimensionality reduction and data representation’, *Neural Computation* **15**, 1373–1396.
- Ben-Tal, A. & Nemirovskiaei, A. S. (2001), *Lectures on modern convex optimization: analysis, algorithms, and engineering applications*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA.
- Binsztok, H., Artieres, T. & Gallinari, P. (2004), ‘A model-based approach to sequence

clustering’, *European Conference on Artificial Intelligence* .

Blaschko, M. B. & Lampert, C. H. (2012), ‘Guest editorial: Special issue on structured prediction and inference’, *Int. J. Comput. Vision* **99**(3), 257–258.

URL: <http://dx.doi.org/10.1007/s11263-012-0530-y>

Blei, D. M., Griffiths, T., Jordan, M. & Tenenbaum, J. (2003), Hierarchical topic models and the nested chinese restaurant process, *in* ‘NIPS’.

Blei, D. M. & Mcauliffe, J. D. (2007), ‘Supervised topic models’.

Blei, D. M., Ng, A. Y. & Jordan, M. I. (2003), ‘Latent dirichlet allocation’, *J. Mach. Learn. Res.* **3**, 993–1022.

URL: <http://dl.acm.org/citation.cfm?id=944919.944937>

Bo, L. & Sminchisescu, C. (2010), ‘Twin gaussian processes for structured prediction’, *Int. J. Comput. Vision* **87**(1-2), 28–52.

URL: <http://dx.doi.org/10.1007/s11263-008-0204-y>

Bunescu, R. & Mooney, R. J. (2006), Subsequence kernels for relation extraction, *in* ‘Submitted to the Ninth Conference on Natural Language Learning (CoNLL-2005)’, Ann Arbor, MI. Available at url<http://www.cs.utexas.edu/users/ml/publication/ie.html>.

URL: <http://www.cs.utexas.edu/users/ai-lab/pub-view.php?PubID=51413>

Chang, C.-C. & Lin, C.-J. (2011), ‘LIBSVM: A library for support vector machines’, *ACM Transactions on Intelligent Systems and Technology* **2**, 27:1–27:27.

Chechik, G. (2008), Max margin dimensionality reduction, Technical report.

Cox, T. & Cox, M. (2001), *Multidimensional Scaling*, Chapman and Hall, second edition.

Davis, M., Putnam, H., Logemann, G. & Loveland, D. W. (1962), ‘A machine program for theorem proving’, *Communications of the ACM* **5** (7) pp. 394–397.

Dehaspe, L. & Toironen, H. (2000), Relational data mining, Springer-Verlag New York, Inc., New York, NY, USA, chapter Discovery of relational association rules, pp. 189–208.

URL: <http://dl.acm.org/citation.cfm?id=567222.567232>

- Dehaspe, L. & Toivonen, H. (1999), ‘Discovery of frequent datalog patterns’, *Data Min. Knowl. Discov.* **3**(1), 7–36.
URL: <http://dx.doi.org/10.1023/A:1009863704807>
- Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R. & Lin, C.-J. (2008), ‘LIBLINEAR: A library for large linear classification’, *Journal of Machine Learning Research* **9**, 1871–1874.
- Fern, A. (2009), ‘A penalty-logic simple-transition model for structured sequences’, *Computational Intelligence* .
- Flach, P. & Lachiche, N. (1999), 1bc: A first-order bayesian classifier, *in* S. Dzeroski & P. Flach, eds, ‘Ninth International Workshop on Inductive Logic Programming (ILP’99)’, Springer-Verlag, pp. 92–103.
- Forney, G. J. (1973), ‘The viterbi algorithm’, *Proceedings of IEEE* **61**(3), 268–278.
- Frank, A. & Asuncion, A. (2010), ‘UCI machine learning repository’.
URL: <http://archive.ics.uci.edu/ml>
- Fukumizu, K., Bach, F. R. & Jordan, M. I. (2003), Kernel dimensionality reduction for supervised learning, MIT Press.
- Gibson, C., van Kasteren, T. & Krose, B. (2008), ‘Monitoring homes with wireless sensor networks’, *Proceedings of the International Med-e-Tel Conference* .
- Gutmann, B. & Kersting, K. (2006), Tildecrf: conditional random fields for logical sequences, *in* ‘Proceedings of the 17th European conference on Machine Learning’, ECML’06, Springer-Verlag, Berlin, Heidelberg, pp. 174–185.
- Hall, M. A. (1998), Correlation-based feature selection for machine learning, Technical report.
- Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P. & Witten, I. H. (2009), ‘The weka data mining software: an update’, *SIGKDD Explor. Newsl.* **11**(1), 10–18.
URL: <http://doi.acm.org/10.1145/1656274.1656278>
- He, Y. (2011), ‘Incorporating sentiment prior knowledge for weakly-supervised sentiment

- analysis', *ACM Transactions on Asian Language Information Processing* . to appear.
- Heras, F., Larrosa, J. & Oliveras, A. (2008), 'Minimaxsat: an efficient weighted max-sat solver', *Journal of Artificial Intelligence Research* **31**(1), 1–32.
- Hofmann, T. (1999), Probabilistic latent semantic indexing, *in* 'Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval', SIGIR '99, ACM, New York, NY, USA, pp. 50–57.
URL: <http://doi.acm.org/10.1145/312624.312649>
- Hogger, C. J. (1990), 'Essentials of logic programming', *Oxford University Press, New York, USA* .
- Jawanpuria, P., Nath, J. S. & Ramakrishnan, G. (2011), Efficient rule ensemble learning using hierarchical kernels., *in* L. Getoor & T. Scheffer, eds, 'ICML', Omnipress, pp. 161–168.
- Jin, X., Xu, A., Bie, R. & Guo, P. (2006), Machine learning techniques and chi-square feature selection for cancer classification using sage gene expression profiles, *in* 'Proceedings of the 2006 international conference on Data Mining for Biomedical Applications', BioDM'06, Springer-Verlag, Berlin, Heidelberg, pp. 106–115.
- Joachims, T., Finley, T. & Yu, C.-N. J. (2009), 'Cutting-plane training of structural svms', *Mach. Learn.* **77**(1), 27–59.
URL: <http://dx.doi.org/10.1007/s10994-009-5108-8>
- Jolliffe, I. (1986a), *Principal Component Analysis*, Springer Verlag.
- Jolliffe, I. T. (1986b), *Principal Component Analysis*, Springer-Verlag.
- Kersting, K. (2005), Say em for selecting probabilistic models for logical sequences, *in* 'In Proceedings of the twenty first conference on uncertainty in artificial intelligence', Morgan Kaufmann, pp. 300–307.
- Kersting, K., Raedt, L. D. & Raiko, T. (2006), 'Logical hidden markov models', *Journal of Artificial Intelligence Research* **25**, 2006.
- Kira, K. & Rendell, L. A. (1992), The feature selection problem: traditional methods

and a new algorithm, in ‘Proceedings of the tenth national conference on Artificial intelligence’, AAAI’92, AAAI Press, pp. 129–134.

URL: <http://dl.acm.org/citation.cfm?id=1867135.1867155>

Kloft, M., Brefeld, U., Sonnenburg, S., Laskov, P., Muller, K. R. & Zien, A. (2009), ‘Efficient and accurate p-norm multiple kernel learning’, *NIPS*.

Kohavi, R. & John, G. H. (1997), ‘Wrappers for feature subset selection’, *ARTIFICIAL INTELLIGENCE* **97**(1), 273–324.

Lacoste-Julien, S., Sha, F. & Jordan, M. I. (2008), Disclda: Discriminative learning for dimensionality reduction and classification, in ‘NIPS’, pp. 897–904.

Lafferty, J. D., McCallum, A. & Pereira, F. C. N. (2001), Conditional random fields: Probabilistic models for segmenting and labeling sequence data, in ‘Proceedings of the Eighteenth International Conference on Machine Learning’, ICML ’01, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 282–289.

URL: <http://dl.acm.org/citation.cfm?id=645530.655813>

Lafferty, J., McCallum, A. & Pereir, F. (2001), ‘Conditional random fields: Probabilistic models for segmenting and labeling sequence data’, *International Conference on Machine Learning*.

Landwehr, N., Gutmann, B., Thon, I., Raedt, L. D. & Philipose, M. (2009), ‘Relational transformation-based tagging for activity recognition’, *Progress on Multi-Relational Data Mining* **89**(1), 111–129.

Landwehr, N., Passerini, A., Raedt, L. D. & Frasconi, P. (2006), ‘Kfoil: Learning simple relational kernels’, *21st National Conference on Artificial Intelligence*.

Lang, K. (n.d.), ‘20 newsgroups data set’.

URL: <http://www.ai.mit.edu/people/jrennie/20Newsgroups/>

Li, F., Huang, M. & Zhu, X. (2010), Sentiment analysis with global topics and local dependency., in M. Fox & D. Poole, eds, ‘AAAI’, AAAI Press.

URL: <http://dblp.uni-trier.de/db/conf/aaai/aaai2010.htmlLiHZ10>

- Li, H., Jiang, T. & Zhang, K. (2003), Efficient and robust feature extraction by maximum margin criterion, *in* ‘In Advances in Neural Information Processing Systems 16’, MIT Press, pp. 157–165.
- Lin, C. & He, Y. (2009), Joint sentiment/topic model for sentiment analysis, *in* ‘Proceedings of the 18th ACM conference on Information and knowledge management’, CIKM ’09, ACM, New York, NY, USA, pp. 375–384.
URL: <http://doi.acm.org/10.1145/1645953.1646003>
- Liu, Y.-Y., Ishikawa, H., Chen, M., Wollstein, G., Schuman, J. S. & Rehg, J. M. (2012), Automated foveola localization in retinal 3d-oct images using structural support vector machine prediction, *in* ‘Proceedings of the 15th international conference on Medical Image Computing and Computer-Assisted Intervention - Volume Part I’, MICCAI’12, Springer-Verlag, Berlin, Heidelberg, pp. 307–314.
- Lodhi, H., Saunders, C., Shawe-Taylor, J., Cristianini, N. & Watkins, C. (2002), ‘Text classification using string kernels’, *J. Mach. Learn. Res.* **2**, 419–444.
URL: <http://dx.doi.org/10.1162/153244302760200687>
- Mauro, N. D., Basile, T. M. A., Ferilli, S. & Esposito, F. (2010), ‘Feature construction for relational sequence learning: Technical report’.
- McCallum, A. K. (2002), Mallet: A machine learning for language toolkit.
<http://mallet.cs.umass.edu>.
- McCallum, A. K. (2003), Efficiently inducing features of conditional random fields. Proceedings of the Nineteenth Conference Annual Conference on Uncertainty in Artificial Intelligence.
- McCreath, E. & Sharma, A. (1998), Lime: A system for learning relations, *in* ‘Proceedings of the 9th International Conference on Algorithmic Learning Theory’, ALT ’98, Springer-Verlag, London, UK, UK, pp. 336–374.
URL: <http://dl.acm.org/citation.cfm?id=647716.735752>
- Miao, X. & Rao, R. P. (2012), ‘Fast structured prediction using large margin sigmoid belief networks’, *Int. J. Comput. Vision* **99**(3), 302–318.

URL: <http://dx.doi.org/10.1007/s11263-011-0423-5>

Micchelli, C. & Pontil, M. (2005), ‘Learning the kernel function via regularization’, *Journal of Machine Learning Research* .

Mihalkova, L. & Richardson, M. (2009), ‘Speeding up inference in statistical relational learning by clustering similar query literals’, *International Conference on Inductive Logic Programming* .

Mika, S., Ratsch, G., Weston, J., Scholkopf, B. & Mullers, K. (1999), Fisher discriminant analysis with kernels, *in* ‘Neural Networks for Signal Processing IX, 1999. Proceedings of the 1999 IEEE Signal Processing Society Workshop’, pp. 41–48.

Nagesh, A., Nair, N. & Ramakrishnan, G. (2013), Comparison between explicit feature learning and implicit modeling of relational features in structured output spaces, *in* ‘23rd International Conference on Inductive Logic Programming’.

Nair, N., Govindan, A., Jayaraman, C., TVS, K. & Ramakrishnan, G. (2011), Pruning search space for weighted first order horn clause satisfiability, *in* ‘Proceedings of the 20th international conference on Inductive logic programming’, ILP’10, Springer-Verlag, Berlin, Heidelberg, pp. 171–180.

URL: <http://dl.acm.org/citation.cfm?id=2022735.2022757>

Nair, N., Nagesh, A. & Ramakrishnan, G. (2012), Probing the space of optimal markov logic networks for sequence labeling, *in* ‘22nd International Conference on Inductive Logic Programming’.

Nair, N., Ramakrishnan, G. & Krishnaswamy, S. (2011), Enhancing activity recognition in smart homes using feature induction, *in* ‘Proceedings of the 13th international conference on Data warehousing and knowledge discovery’, DaWaK’11, Springer-Verlag, Berlin, Heidelberg, pp. 406–418.

URL: <http://dl.acm.org/citation.cfm?id=2033616.2033657>

Nair, N., Saha, A., Ramakrishnan, G. & Krishnaswamy, S. (2012), Rule ensemble learning using hierarchical kernels in structured output spaces, *in* ‘Twenty-Sixth AAAI Conference on Artificial Intelligence’.

- Perotte, A. J., Wood, F., Elhadad, N. & Bartlett, N. (2011), Hierarchically supervised latent dirichlet allocation, *in* ‘NIPS’, pp. 2609–2617.
- Rabiner, L. R. (1990), Readings in speech recognition, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, chapter A tutorial on hidden Markov models and selected applications in speech recognition, pp. 267–296.
URL: <http://dl.acm.org/citation.cfm?id=108235.108253>
- Rakotomamonjy, A., Bach, F., Canu, S. & Grandvalet, Y. (2008), ‘Simplemkl’, *JMLR* **9**, 2491–2521.
- Ramage, D., Hall, D., Nallapati, R. & Manning, C. D. (2009), Labeled lda: a supervised topic model for credit attribution in multi-labeled corpora, *in* ‘Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1 - Volume 1’, EMNLP ’09, Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 248–256.
URL: <http://dl.acm.org/citation.cfm?id=1699510.1699543>
- Richardson, M. & Domingos, P. (2006), ‘Markov logic networks’, *Mach. Learn.* **62**(1-2), 107–136.
URL: <http://dx.doi.org/10.1007/s10994-006-5833-1>
- Rivest, R. L. (1987), ‘Learning decision lists’, *Machine Learning* **2**(3), 229–246.
- Roth, D. & tau Yih, W. (2005), Integer linear programming inference for conditional random fields, *in* ‘Proceedings of the 22nd international conference on Machine learning’, ICML ’05, ACM, New York, NY, USA, pp. 736–743.
URL: <http://doi.acm.org/10.1145/1102351.1102444>
- Roth, D. & tau Yih, W. (2007), Global Inference for Entity and Relation Identification via a Linear Programming Formulation, *in* L. Getoor & B. Taskar, eds, ‘Introduction to Statistical Relational Learning’, The MIT press.
- Sarawagi, S. & Gupta, R. (2008), Accurate max-margin training for structured output spaces, *in* ‘Proceedings of the 25th international conference on Machine learning’, ICML ’08, ACM, New York, NY, USA, pp. 888–895.

URL: <http://doi.acm.org/10.1145/1390156.1390268>

Schölkopf, B., Smola, A. & Müller, K.-R. (1997), Kernel principal component analysis, *in* W. Gerstner, A. Germond, M. Hasler & J.-D. Nicoud, eds, ‘Artificial Neural Networks ICANN’97’, Vol. 1327 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 583–588. 10.1007/BFb0020217.

URL: <http://dx.doi.org/10.1007/BFb0020217>

Schulte, O., Khosravi, H., Kirkpatrick, A., Man, T., Gao, T. & Zhu, Y. (2012), Modelling relational statistics with bayes nets, *in* ‘proceedings of 22nd International Conference on Inductive Logic Programming (ILP-2012)’, Springer.

Selman, B., Kautz, H. & Cohen, B. (1993), ‘Local search strategies for satisfiability testing’, *Second DIMACS Implementation Challenge on Cliques, Coloring and Satisfiability*.

Selman, B., Levesque, H. & Mitchell, D. (1992), ‘A new method for solving hard satisfiability problems’, *AAAI-92, San Jose, CA* pp. 440–446.

Shavlik, J. & Natarajan, S. (2009), ‘Speeding up inference in markov logic networks by preprocessing to reduce the size of the resulting grounded network’, *Proceedings of the 21st international JCAI*.

Siegel, S. & Castellan, N. (1988), *Nonparametric statistics for the behavioral sciences*, second edn, McGraw-Hill, Inc.

Singla, P. & Domingos, P. (2006), ‘Memory-efficient inference in relational domains’, *Proceedings of the Twenty-First National Conference on Artificial Intelligence* pp. 488–493.

Sion, M. (1958), ‘On general minimax theorem’, *Pacific Journal of Mathematics*.

Srinivasan, A. (2007), ‘The aleph manual’, *Technical Report, University of Oxford*.

Stewart, L., He, X. & Zemel, R. S. (2008), ‘Learning flexible features for conditional random fields’, *IEEE Trans. Pattern Anal. Mach. Intell.* **30**(8), 1415–1426.

URL: <http://dx.doi.org/10.1109/TPAMI.2007.70790>

- Szafranski, M. & Rakotomamonjy, A. (2008), ‘Composite kernel learning’, *ICML* .
- Tan, M., Wang, L. & Tsang, I. W. (2010), Learning sparse svm for feature selection on very high dimensional datasets, *in* ‘Proceedings of the 27th International Conference on Machine Learning (ICML-10)’, pp. 1047–1054.
- Tapia, E. M. (2003), ‘Activity recognition in the home setting using simple and ubiquitous sensors’, *S.M. Thesis, Massachusetts Institute of Technology* .
- Tapia, E. M. (2004), ‘Activity recognition in the home using simple and ubiquitous sensors’, *International Conference on Pervasive Computing* .
- Taskar, B., Chatalbashev, V. & Koller, D. (2004), Learning associative markov networks, *in* ‘Proceedings of the twenty-first international conference on Machine learning’, ICML ’04, ACM, New York, NY, USA, pp. 102–.
- URL:** <http://doi.acm.org/10.1145/1015330.1015444>
- Taskar, B., Guestrin, C. & Koller, D. (2004), Max-margin markov networks, *in* S. Thrun, L. Saul & B. Schölkopf, eds, ‘Advances in Neural Information Processing Systems 16’, MIT Press, Cambridge, MA.
- Taskar, B., Lacoste-Julien, S. & Jordan, M. I. (2006), ‘Structured prediction, dual extragradient and bregman projections’, *J. Mach. Learn. Res.* **7**, 1627–1653.
- URL:** <http://dl.acm.org/citation.cfm?id=1248547.1248607>
- Teh, Y. W., Jordan, M. I., Beal, M. J. & Blei, D. M. (2004), ‘Hierarchical dirichlet processes’, *Journal of the American Statistical Association* **101**.
- Thon, I. (2010), Don’t fear optimality: sampling for probabilistic-logic sequence models, *in* ‘Proceedings of the 19th international conference on Inductive logic programming’, ILP’09, Springer-Verlag, Berlin, Heidelberg, pp. 226–233.
- URL:** <http://dl.acm.org/citation.cfm?id=1893538.1893560>
- Thon, I., Landwehr, N. & Raedt, L. (2011), ‘Stochastic relational processes: Efficient inference and applications’, *Mach. Learn.* **82**(2), 239–272.
- URL:** <http://dx.doi.org/10.1007/s10994-010-5213-8>

- Tsochantaridis, I. (2006), ‘Support vector machine learning for interdependent and structured output spaces’.
- Tsochantaridis, I., Hofmann, T., Joachims, T. & Altun, Y. (2004), Support vector machine learning for interdependent and structured output spaces, *in* ‘Proceedings of the twenty-first international conference on Machine learning’, ICML ’04, ACM, New York, NY, USA, pp. 104–.
- URL:** <http://doi.acm.org/10.1145/1015330.1015341>
- van Kasteren, T., Noulas, A., Englebienne, G. & Kröse, B. (2008), Accurate activity recognition in a home setting, *in* ‘Proceedings of the 10th international conference on Ubiquitous computing’, UbiComp ’08, ACM, New York, NY, USA, pp. 1–9.
- URL:** <http://doi.acm.org/10.1145/1409635.1409637>
- Wang, S., Pentney, W., Popescu, A.-M., Choudhury, T. & Philipose, M. (2007), ‘Common sense based joint training of human activity recognizers’, *20th International Joint Conference on Artificial Intelligence* .
- Wilson, D. H. (2005), ‘Assistive intelligent environments for automatic health monitoring’, *PhD Thesis, Carnegie Mellon University* .
- Xie, B. & Passonneau, R. J. (n.d.), Supervised hdp using prior knowledge, Technical report.
- URL:** <http://www1.ccls.columbia.edu/~beck/pubs/shdp.pdf>
- Xu, W. (2010), Supervising latent topic model for maximum-margin text classification and regression, *in* M. Zaki, J. Yu, B. Ravindran & V. Pudi, eds, ‘Advances in Knowledge Discovery and Data Mining’, Vol. 6118 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 403–414. 10.1007/978-3-642-13657-3_44.
- Ye, J. & Ji, S. (n.d.), ‘Discriminant analysis for dimensionality reduction: An overview of recent developments’.
- Yu, J., Tian, Q., Rui, T. & Huang, T. (2007), ‘Integrating discriminant and descriptive information for dimension reduction and classification’, *Circuits and Systems for Video Technology, IEEE Transactions on* **17**(3), 372–377.

Zaki, M. J. & Aggarwal, C. C. (2006), ‘Xrules: An effective algorithm for structural classification of xml data’, *Mach. Learn.* **62**(1-2), 137–170.

URL: <http://dx.doi.org/10.1007/s10994-006-5832-2>

Zhai, Y., Tan, M., Ong, Y. S. & Tsang, I. W. (2012), Discovering support and affiliated features from very high dimensions, *in* J. Langford & J. Pineau, eds, ‘Proceedings of the 29th International Conference on Machine Learning (ICML-12)’, ACM, New York, NY, USA, pp. 1455–1462.

URL: <http://icml.cc/2012/papers/726.pdf>

Zhu, J., Ahmed, A. & Xing, E. (2010), ‘MedLDA: Maximum Margin Supervised Topic Models’, *Journal of Machine Learning Research* **1**, 1–48.