On Primal and Dual Sparsity of Markov Networks

Jun Zhu[†*] JUN-ZHU@MAILS.TSINGHUA.EDU.CN
Eric P. Xing[†] EPXING@CS.CMU.EDU
[*]Dept. of Comp. Sci & Tech, TNList Lab, Tsinghua University, Beijing 100084 China
[†]School of Computer Science, Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh, PA 15213 USA

Abstract

Sparsity is a desirable property in high dimensional learning. The ℓ_1-norm regularization can lead to primal sparsity, while max-margin methods achieve dual sparsity. Combining these two methods, an ℓ_1-norm max-margin Markov network (ℓ_1-M^{3}N) can achieve both types of sparsity. This paper analyzes its connections to the Laplace max-margin Markov network (LapM^{3}N), which inherits the dual sparsity of max-margin models but is pseudo-primal sparse, and to a novel adaptive M^{3}N (AdapM^{3}N). We show that the ℓ_1-M^{3}N is an extreme case of the LapM^{3}N, and the ℓ_1-M^{3}N is equivalent to an AdapM^{3}N. Based on this equivalence we develop a robust EM-style algorithm for learning an ℓ_1-M^{3}N. We demonstrate the advantages of the simultaneously (pseudo-) primal and dual sparse models over the ones which enjoy either primal or dual sparsity on both synthetic and real data sets.

1. Introduction

Learning structured prediction models, which explicitly explore the structural dependencies among input features (e.g., text sequences, DNA strings) and structural interpretational outputs (e.g., parsing trees, gene annotations), has gained substantial popularity in data mining, machine intelligence, and scientific discovery. Based on different learning paradigms, major instances of such models include the conditional random fields (CRFs) (Lafferty et al., 2001) based on maximum conditional likelihood estimation, and max-margin Markov networks (M^{3}Ns) (Taskar et al., 2003) or structural SVMs (Altun et al., 2003; Tsochantaridis et al., 2004) based on max-margin learning.

Appearing in *Proceedings of the 26th International Conference on Machine Learning*, Montreal, Canada, 2009.

For domains with complex feature spaces, it is often desirable to pursue a sparse representation of the model that leaves out irrelevant features. We say a model enjoys *primal sparsity*, if only very few features in the original model have non-zero weights. The term "primal" stems from a convention in the optimization literature, which generally refers to (constrained) problems pertaining to the original model. Primal sparsity is important for selecting significant features and for reducing the risk of over-fitting. The *bet-on-sparsity principle* (Friedman et al., 2004) suggests that one should prefer the model that does well in sparse problems. In likelihood-based estimation, sparse model fitting has been extensively studied. A common strategy is to add an ℓ_1-penalty of feature weights to the log-likelihood function. Due to the singularity of ℓ_1-norm at the origin (Tibshirani, 1996), ℓ_1-norm regularization can lead to primal sparse estimates. Recent work on structure learning of graphical models (Lee et al., 2006; Wainwright et al., 2006) falls into this paradigm.

Another type of sparsity, as enjoyed by large margin models, like the unstructured SVM and the structured M^{3}N, is the *dual sparsity*, which refers to a phenomenon that only a few Lagrangian multipliers in the dual form of the original model turn out to be non-zero. When a model is dual sparse, its decision boundary depends only on a few number of support vectors, which in principle leads to a robust decision boundary. Moreover, the dual sparsity provides a theoretical motivation of the cutting-plane algorithms (Tsochantaridis et al., 2004) and the bundle methods (Smola et al., 2007), which generally explore the fact that in max-margin models only very few (e.g., polynomial) number of constraints are sufficient to achieve a good enough solution. Unfortunately, although both primal and dual sparsity can benefit structured prediction models, they usually do not co-exist. For example, the powerful M^{3}N is not primal sparse, because it employs an ℓ_2-norm penalty that cannot automatically select significant features.

One natural way to bring these two types of spar-

sity together is to build an ℓ_1-norm regularized large margin model. In the structured learning setting, ℓ_1-norm regularized max-margin Markov networks (ℓ_1-M^{3}N) can be formulated, following the same spirit of the unstructured 1-norm SVM (Zhu et al., 2004). Another approach that attempts to achieve both primal and dual sparsity is the recently proposed Laplace max-margin Markov networks (LapM^{3}N) (Zhu et al., 2008b), which inherit the dual sparseness of max-margin models. However, since the posterior shrinkage effect as shown in (Zhu et al., 2008b) is smooth, LapM^{3}N is *pseudo-primal sparse* (i.e., only very few input features have *large* weights) and does not explicitly select features by setting the weights of irrelevant features to zeros.

This paper presents the ℓ_1-M^{3}N and a novel *adaptive* M^{3}N, and analyzes their close connections to the LapM^{3}N from both theoretical and algorithmic perspectives. In the theoretical aspect, we show that ℓ_1-M^{3}N is an extreme case of LapM^{3}N when the regularization constant of the entropic regularizer goes to infinity and LapM^{3}N is a smooth relaxation of ℓ_1-M^{3}N. We also show that ℓ_1-M^{3}N is equivalent to an adaptive M^{3}N. In the algorithmic aspect, based on the equivalence between ℓ_1-M^{3}N and an adaptive M^{3}N, we develop a novel EM-style algorithm to learn an ℓ_1-M^{3}N. The robust algorithm has the same structure as the variational algorithm of LapM^{3}N (Zhu et al., 2008b) and helps uncover the difference between ℓ_1-M^{3}N and LapM^{3}N. Finally, we present empirical studies comparing ℓ_1-M^{3}N and LapM^{3}N with competing models, which enjoy either primal or dual sparsity but not both, on both synthetic and real data sets.

The rest of the paper is structured as follows. Sec. 2 presents problem formulations, including the ℓ_1-M^{3}N and an adaptive M^{3}N. Sec. 3 presents the theoretical connections, while Sec. 4 discusses the algorithmic connection. Sec. 5 presents our empirical studies, and Sec. 6 concludes this paper.

2. Problem Formulation

Structured output classification aims to learn a predictive function $h : \mathcal{X} \to \mathcal{Y}$, where $\mathcal{X}$ is the subspace of inputs and $\mathcal{Y} = \mathcal{Y}_1 \times \cdots \times \mathcal{Y}_l$ is the space of outputs, which are multivariate and structured. For part-of-speech (POS) tagging, $\mathcal{Y}_i$ consists of all the POS tags, and each input $\mathbf{x}$ is a word sequence and each label $\mathbf{y} = (y_1, \cdots, y_l)$ is a sequence of POS tags. We assume a finite number of feasible outputs for any input.

In supervised learning, where input-output pairs $(\mathbf{x}, \mathbf{y})$ are drawn i.i.d. from a distribution $P(\mathbf{X}, \mathbf{Y})$, the goal is to find an h from a hypothesis space that minimizes the risk: $\mathcal{R}(h) = E_{(\mathbf{x},\mathbf{y})\sim P}[\Delta\ell(h(\mathbf{x}), \mathbf{y})]$, where $\Delta\ell$ is a non-negative loss function, e.g., the *hamming loss*: $\Delta\ell(\hat{\mathbf{y}}, \mathbf{y}) = \sum_{j=1}^{l} \mathbb{I}(\hat{y}_j \neq y_j)$, where $\mathbb{I}(\cdot)$ is an indicator function that equals one if the argument holds and zero otherwise. $\Delta\ell(\hat{\mathbf{y}}, \mathbf{y})$ measures the loss of the prediction $\hat{\mathbf{y}}$ when the true prediction is $\mathbf{y}$. Since the true distribution P is unknown, empirical risk is used as an approximation of the risk. Given a set of training data $\mathcal{D} = \{(\mathbf{x}^i, \mathbf{y}^i)\}_{i=1}^N$, drawn i.i.d from P, the empirical risk is: $\mathcal{R}_{emp}(h) = \frac{1}{N}\sum_{i=1}^{N} \Delta\ell(h(\mathbf{x}^i), \mathbf{y}^i)$. To avoid over-fitting, one method is to minimize the regularized empirical risk: $\lambda\Omega(h) + \mathcal{R}_{emp}(h)$, where $\Omega(h)$ is a regularizer, and λ is a regularization parameter.

2.1. ℓ_2-norm Max-Margin Markov networks

Let $F(\ .\ ; \mathbf{w}) : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ be a parametric discriminant function over the input-output pairs. The max-margin Markov networks define a predictive rule as an optimization problem:

$$h_0(\mathbf{x}; \mathbf{w}) = \arg\max_{\mathbf{y}\in\mathcal{Y}} F(\mathbf{x}, \mathbf{y}; \mathbf{w}). \tag{1}$$

A common choice of F is a linear model, where F is defined by a set of K feature functions $f_k : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ and their weights w_k: $F(\mathbf{x}, \mathbf{y}; \mathbf{w}) = \mathbf{w}^\top \mathbf{f}(\mathbf{x}, \mathbf{y})$.

Consider the general case where errors are allowed in the training data. To learn such a prediction rule h_0, the "margin re-scaling" ℓ_2-norm M^{3}N (Taskar et al., 2003) minimizes a regularized structured hinge loss:

$$\min_{\mathbf{w}} \ \lambda\|\mathbf{w}\|_2^2 + \mathcal{R}_{hinge}(\mathbf{w}), \tag{2}$$

where $\mathcal{R}_{hinge}(\mathbf{w}) \triangleq \frac{1}{N}\sum_i \max_{\mathbf{y}\in\mathcal{Y}}[\Delta\ell_i(\mathbf{y}) - \mathbf{w}^\top\Delta\mathbf{f}_i(\mathbf{y})]$, of which $\Delta\ell_i(\mathbf{y}) = \Delta\ell(\mathbf{y}, \mathbf{y}^i)$ and $\Delta\mathbf{f}_i(\mathbf{y}) = \mathbf{f}(\mathbf{x}^i, \mathbf{y}^i) - \mathbf{f}(\mathbf{x}^i, \mathbf{y})$. $\mathbf{w}^\top\Delta\mathbf{f}_i(\mathbf{y})$ is the "margin" favored by the true label $\mathbf{y}^i$ over a prediction $\mathbf{y}$. Since $\max_{\mathbf{y}\in\mathcal{Y}}[\Delta\ell_i(\mathbf{y}) - \mathbf{w}^\top\Delta\mathbf{f}_i(\mathbf{y})] \geq \Delta\ell(\mathbf{y}, \mathbf{y}^i)$ for $\mathbf{w}^\top\Delta\mathbf{f}_i(\mathbf{y}) \leq 0$, $\mathcal{R}_{hinge}(\mathbf{w})$ is an upper bound of the empirical risk of the prediction rule (1).

The problem (2) can be equivalently formulated as a constrained optimization problem:

$$\begin{aligned} \text{P0 (M}^3\text{N)}: \quad & \min_{\mathbf{w},\xi} \ \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{N}\xi_i \\ \text{s.t. } \forall i, \forall \mathbf{y} \neq \mathbf{y}^i: \quad & \mathbf{w}^\top\Delta\mathbf{f}_i(\mathbf{y}) \geq \Delta\ell_i(\mathbf{y}) - \xi_i;\ \xi_i \geq 0, \end{aligned}$$

where ξ_i is a slack variable absorbing errors in training data and C is a positive constant. P0 is a convex program and satisfies the Slater's condition. Due to the KKT condition, ℓ_2-norm M^{3}N enjoys the *dual sparsity*, i.e., only a few lagrange multipliers are non-zero, which correspond to the active constraints whose equality holds, analogous to the support vectors in SVM. However, due to the differentiability of ℓ_2-norm, the ℓ_2-norm M^{3}N is not primal sparse.

Exploring sparse dependencies among individual labels in $\mathbf{y}$, efficient optimization algorithms based on cutting-plane (Tsochantaridis et al., 2004), message-passing (Taskar et al., 2003), or gradient descent (Bartlett et al., 2004; Ratliff et al., 2007) have been proposed to (approximately) solve P0.

2.2. Laplace Max-Margin Markov Networks

Unlike the M^{3}N, which performs point-estimate to predict based on a single rule $F(\ .\ ;\mathbf{w})$, the Laplace max-margin Markov networks (LapM^{3}N) (Zhu et al., 2008b) approach the structured prediction problem by performing Bayesian-style learning under the general *structured maximum entropy discrimination* formalism (Zhu et al., 2008b), which facilitates a Bayes-style prediction by averaging $F(\ .\ ;\mathbf{w})$ over a posterior distribution of rules $p(\mathbf{w})$:

$$h_1(\mathbf{x}) = \arg\max_{\mathbf{y}\in\mathcal{Y}} \int p(\mathbf{w}) F(\mathbf{x},\mathbf{y};\mathbf{w})\,\mathrm{d}\mathbf{w}\,, \qquad (3)$$

where $p(\mathbf{w})$ is estimated by learning a *maximum entropy discrimination Markov network* (MaxEnDNet, or MEDN) (Zhu & Xing, 2009).

In the same spirit of the structured hinge loss of M^{3}N, the empirical risk of the averaging model (3) is upper bounded by the *expected* structured hinge loss:

$$\mathcal{R}_{hinge}(p(\mathbf{w})) = \frac{1}{N}\sum_i \max_{\mathbf{y}\in\mathcal{Y}} \int p(\mathbf{w})[\Delta\ell_i(\mathbf{y}) - \Delta F_i(\mathbf{y};\mathbf{w})]\,\mathrm{d}\mathbf{w}\,.$$

MaxEnDNet minimizes a regularized structured hinge loss as in (2) and uses the KL-divergence with a prior to regularize $p(\mathbf{w})$, i.e., $\Omega(p(\mathbf{w})) = KL(p(\mathbf{w})\|p_0(\mathbf{w}))$. Similar to (2), the MaxEnDNet can be equivalently formulated as a constrained optimization problem:

$$\text{P1 (MEDN)}: \min_{p(\mathbf{w}),\xi}\ KL(p(\mathbf{w})||p_0(\mathbf{w})) + U(\xi)$$
$$\text{s.t. } \forall i,\mathbf{y}: \int p(\mathbf{w})[\Delta F_i(\mathbf{y};\mathbf{w}) - \Delta\ell_i(\mathbf{y})]\,\mathrm{d}\mathbf{w}\ \geq -\xi_i; \xi_i \geq 0,$$

where $U(\xi)$ is a closed proper convex function over slack variables ξ, e.g., $U(\xi) = C\sum_i \xi_i$. U is also known as a potential term in the maximum entropy principle.

The problem P1 is a convex program, and satisfies the Slater's condition. As showin in (Zhu et al., 2008b), the optimum solution of P1 is:

$$p(\mathbf{w}) = \frac{1}{Z(\alpha)} p_0(\mathbf{w}) \exp\{\sum_{i,\mathbf{y}} \alpha_i(\mathbf{y})[\Delta F_i(\mathbf{y};\mathbf{w}) - \Delta\ell_i(\mathbf{y})]\},$$

where $Z(\alpha)$ is a normalization factor and the lagrange multipliers $\alpha_i(\mathbf{y})$ (corresponding to constraints in P1) can be obtained by solving the following dual problem:

$$\text{D1}: \quad \max_{\alpha}\ -\log Z(\alpha) - U^\star(\alpha)$$
$$\text{s.t. }\ \alpha_i(\mathbf{y}) \geq 0,\ \forall i,\ \forall \mathbf{y},$$

where $U^\star(\cdot)$ is the conjugate of the slack function $U(\cdot)$, i.e., $U^\star(\alpha) = \sup_\xi \big(\sum_{i,\mathbf{y}} \alpha_i(\mathbf{y})\xi_i - U(\xi)\big)$.

Due to the KKT condition, the above solution enjoys the dual sparsity as in M^{3}N. Thus, MaxEnDNet enjoys a similar generalization property as the M^{3}N and SVM due to the small "effective size" of the margin constraints. In fact, as shown in (Zhu et al., 2008b), ℓ_2-norm M^{3}N is a Gaussian MaxEnDNet where the prior is standard normal and $U(\xi) = C\sum_i \xi_i$.

The Laplace max-margin Markov network (LapM^{3}N) is a Laplace MaxEnDNet by using a heavy tailed Laplace prior, which encodes the prior belief that the distribution of $\mathbf{w}$ is strongly peaked around zero. Since the KL-divergence is differentiable, the resulting posterior shrinkage effect in LapM^{3}N, as shown in (Zhu et al., 2008b), is smooth. Thus, in the input feature space, LapM^{3}N is *pseudo-primal sparse*, i.e., only a few elements in $\mathbf{w}$ have large values. This pseudo-primal sparsity makes LapM^{3}N enjoy nice robust properties and in many cases as we shall see LapM^{3}N can perform as well as a primal sparse M^{3}N, as presented below. The robustness of KL-regularization is also demonstrated in sparse coding (Bradley & Bagnell, 2008).

Below, we introduce two novel formulations of sparse M^{3}Ns and then analyze their connections.

2.3. ℓ_1-norm Max-Margin Markov Networks

To introduce the primal sparsity in max-margin Markov networks in a more direct way, we propose to use the ℓ_1-norm of the model parameters in the regularized hinge loss minimization framework (2). Therefore, the ℓ_1-M^{3}N is formulated as follows,

$$\text{P2 } (\ell_1\text{-M}^3\text{N}): \quad \min_{\mathbf{w},\xi}\ \frac{1}{2}\|\mathbf{w}\|_1 + C\sum_{i=1}^N \xi_i$$
$$\text{s.t. } \forall i,\ \forall \mathbf{y}\neq\mathbf{y}^i: \quad \mathbf{w}^\top \Delta\mathbf{f}_i(\mathbf{y}) \geq \Delta\ell_i(\mathbf{y}) - \xi_i;\ \xi_i \geq 0,$$

where $\|.\|_1$ is the ℓ_1-norm. Another equivalent formulation[1], which is useful in the subsequent analysis and algorithm development, is as follows:

$$\text{P2}': \quad \min_{\mathbf{w}}\ \mathcal{R}_{hinge}(\mathbf{w})$$
$$\text{s.t.} \quad \|\mathbf{w}\|_1 \leq \lambda$$

Unlike ℓ_2-norm, the ℓ_1-norm is not differentiable at the origin. This singularity property ensures that the ℓ_1-M^{3}N is able to remove noise features by estimating their weights to be exactly zero. When the feature space is high dimensional and has many noise features, the ℓ_2-norm M^{3}N will suffer a poor generalization ability caused by these noise features. Thus, the ℓ_1-norm M^{3}N, or the closely related pseudo-sparse LapM^{3}N, would be a better choice in this scenario. Moreover, the primal sparse ℓ_1-M^{3}N is of great interest itself like the 1-norm SVM because it can automatically select significant features in max-margin Markov networks.

[1] See (Taskar et al., 2006) for transformation techniques.

To learn an ℓ_1-M^{3}N, various methods can be applied, such as the sub-gradient method (Ratliff et al., 2007) with a projection to an ℓ_1-ball (Duchi et al., 2008) based on the P2$'$ formulation, and the cutting-plane method (Tsochantaridis et al., 2004) with an LP solver to solve the generated LP sub-problems based the formulation P2. Our empirical studies show that both of these algorithms are sensitive to their regularization constants. We will develop a novel EM-style algorithm, which is robust and helps uncover the connection and difference between ℓ_1-M^{3}N and LapM^{3}N.

2.4. Adaptive Max-Margin Markov Networks

Our EM-style algorithm for the ℓ_1-M^{3}N is developed based on an equivalence between the ℓ_1-M^{3}N and an *adaptive* M^{3}N, which is defined as follows:

$$\begin{aligned} \text{P3 (AdapM}^3\text{N)}: \quad & \min_{\mathbf{w},\tau,\xi} \ \mathbf{w}^\top \Sigma^{-1} \mathbf{w} + C \sum_{i=1}^{N} \xi_i, \\ \text{s.t. } \forall i, \ \forall \mathbf{y} \neq \mathbf{y}^i: \quad & \mathbf{w}^\top \Delta \mathbf{f}_i(\mathbf{y}) \geq \Delta \ell_i(\mathbf{y}) - \xi_i; \xi_i \geq 0 \\ \forall k: \quad & \frac{1}{K} \sum_{k=1}^{K} \tau_k = \frac{1}{\lambda}; \tau_k \geq 0. \end{aligned}$$

where $\Sigma = \text{diag}(\tau)$.

The rationale behind P3 is that: by adaptively penalizing different components, the coefficients of irrelevant features can be shrunk to zero, i.e., the corresponding τ go to zero. The same idea have been explored in Automatic Relevance Determination (Qi et al., 2004) and sparse Bayesian learning (Tipping, 2001). The mathematical intuition is from a two-layer interpretation of the Laplace prior (Figueiredo, 2003), namely, a univariate Laplace distribution $p(w) = \frac{\sqrt{\lambda}}{2} e^{-\sqrt{\lambda}|w|}$ is equivalent to a Gaussian-exponential model, where w is a zero-mean normal $p(w|\tau) = \mathcal{N}(w|0, \tau)$ and the variance τ has an exponential hyper-prior: $p(\tau|\lambda) = \frac{\lambda}{2} \exp\{-\frac{\lambda}{2}\tau\}$, for $\tau \geq 0$. Therefore, the quadratic term of $\mathbf{w}^\top \Sigma^{-1} \mathbf{w}$ in P3 is from the first layer Gaussian distribution, and the constraint $\frac{1}{K}\sum_k \tau_k = \frac{1}{\lambda}$ is from the second-layer hyper-prior, because the mean of the exponential hyper-prior is $E[\tau] = \frac{1}{\lambda}$ and $\frac{1}{K}\sum_{k=1}^{K} \tau_k$ is an empirical estimate of $E[\tau]$. Thus, the first-order constraint $\frac{1}{K}\sum_{k=1}^{K} \tau_k = \frac{1}{\lambda}$ can be seen as a relaxation of the exponential hyper-prior.

3. Theoretical Connections

In this section, we show the theoretical connections of three variants of sparse max-margin Markov networks. We show that LapM^{3}N is a smooth relaxation of ℓ_1-M^{3}N; ℓ_1-M^{3}N is an extreme case of LapM^{3}N; and ℓ_1-M^{3}N is equivalent to an adaptive M^{3}N. Due to space limitation, the proofs are deferred to a longer version.

We begin with the special case of Gaussian MaxEnDNet, for which we have the following corollary:

Corollary 1 *Assuming* $F(\mathbf{x}, \mathbf{y}; \mathbf{w}) = \mathbf{w}^\top \mathbf{f}(\mathbf{x}, \mathbf{y})$, $U(\xi) = C\sum_i \xi_i$, *and* $p_0(\mathbf{w}) = \mathcal{N}(0, I)$, *the mean* μ *of the posterior distribution* $p(\mathbf{w})$ *under the MaxEnDNet is achieved by solving the following problem:*

$$\begin{aligned} & \min_{\mu,\xi} \ \frac{1}{2}\mu^\top \mu + C\sum_{i=1}^{N} \xi_i \\ \text{s.t. } \forall i, \ \forall \mathbf{y} \neq \mathbf{y}^i: \quad & \mu^\top \Delta \mathbf{f}_i(\mathbf{y}) \geq \Delta \ell_i(\mathbf{y}) - \xi_i; \ \xi_i \geq 0. \end{aligned}$$

Since in linear models the averaging prediction rule h_1 is determined by the posterior mean, the above corollary shows a reduction of MaxEnDNet to M^{3}N. This result is complementary to the reduction theorem in (Zhu et al., 2008b), which considers dual problems.

3.1. LapM^{3}N v.s. ℓ_1-M^{3}N

By using the Laplace prior in MaxEnDNet, we can get the following theorem for LapM^{3}N.

Theorem 2 *Assuming* $F(\mathbf{x}, \mathbf{y}; \mathbf{w}) = \mathbf{w}^\top \mathbf{f}(\mathbf{x}, \mathbf{y})$, $U(\xi) = C\sum_i \xi_i$, *and* $p_0(\mathbf{w}) = \left(\frac{\sqrt{\lambda}}{2}\right)^K e^{-\sqrt{\lambda}\|\mathbf{w}\|}$, *the mean* μ *of the posterior distribution* $p(\mathbf{w})$ *under the MaxEnDNet is obtained by solving the following primal problem:*

$$\begin{aligned} & \min_{\mu,\xi} \sqrt{\lambda}\sum_{k=1}^{K}\left(\sqrt{\mu_k^2 + \frac{1}{\lambda}} - \frac{1}{\sqrt{\lambda}} \log \frac{\sqrt{\lambda\mu_k^2 + 1} + 1}{2}\right) + C\sum_{i=1}^{N} \xi_i \\ \text{s.t. } & \mu^\top \Delta \mathbf{f}_i(\mathbf{y}) \geq \Delta \ell_i(\mathbf{y}) - \xi_i; \ \xi_i \geq 0, \quad \forall i, \ \forall \mathbf{y} \neq \mathbf{y}^i. \end{aligned}$$

Since the term $\sum_{k=1}^{K}\left(\sqrt{\mu_k^2 + \frac{1}{\lambda}} - \frac{1}{\sqrt{\lambda}} \log \frac{\sqrt{\lambda\mu_k^2+1}+1}{2}\right)$ corresponds to the KL-divergence between $p(\mathbf{w})$ and $p_0(\mathbf{w})$ under a Laplace MaxEnDNet, we will refer to it as a *KL-norm*[2] and denote it by $\|\mu\|_{KL}$ in the sequel. This KL-norm is different from the ℓ_2-norm, but is closely related to the ℓ_1-norm, which encourages a sparse estimator due to its singularity at the origin. Specifically, we have the following corollary:

Corollary 3 *The LapM^{3}N yields the same estimate as the ℓ_1-M^{3}N when $\lambda \to \infty$.*

To prove Corollary 3, we note that as λ goes to infinity, the logarithm terms in $\|\mu\|_{KL}$ disappear because of the fact that $\frac{\log x}{x} \to 0$ when $x \to \infty$. Thus, the KL-norm $\|\mu\|_{KL}$ approaches $\|\mu\|_1$, i.e., the ℓ_1-norm, as $\lambda \to \infty$. This means that the LapM^{3}N will be (nearly) the same as the ℓ_1-M^{3}N if the regularization constant λ is large enough. In (Zhu et al., 2008b), a posterior shrinkage effect is shown based on the exact computation of the

[2]This is not exactly a norm because the positive scalability does not hold. However, by using the inequality $e^x \geq 1 + x$, we can show: $\forall k$, $\left(\sqrt{\mu_k^2 + \frac{1}{\lambda}} - \frac{1}{\sqrt{\lambda}} \log \frac{\sqrt{\lambda\mu_k^2+1}+1}{2}\right)$ is monotonically increasing with respect to μ_k^2 and $\|\mu\|_{KL} \geq K/\sqrt{\lambda}$, where the equality holds only when $\mu = 0$. Thus, $\|\mu\|_{KL}$ penalizes large weights. For convenient comparison with the popular ℓ_2 and ℓ_1 norms, we call it a KL-norm.

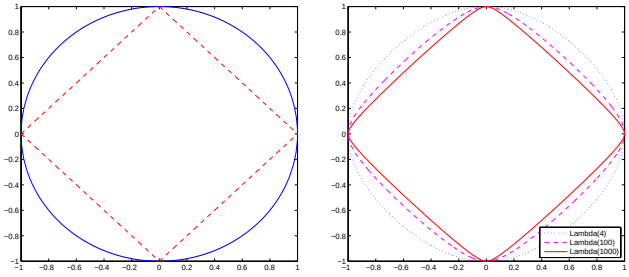


Figure 1. (Left) ℓ_2-norm (solid line) and ℓ_1-norm (dashed line); (Right) KL-norm with different Laplace priors.

normalization factor. Theorem 2 and Corollary 3 offer another perspective of how the pseudo-primal sparse LapM^{3}N relates to the primal sparse ℓ_1-M^{3}N.

A more explicit illustration of the entropic regularization under a LapM^{3}N, comparing to the ℓ_1 and ℓ_2 regularization over an M^{3}N, can be seen in Figure 1, where the feasible regions due to the three different norms used in the regularizer are plotted in a two dimensional space. Specifically, it shows (1) ℓ_2-norm: $w_1^2 + w_2^2 \leq 1$; (2) ℓ_1-norm: $|w_1| + |w_2| \leq 1$; and (2) KL-norm[3]: $\sqrt{w_1^2 + 1/\lambda} + \sqrt{w_2^2 + 1/\lambda} - (1/\sqrt{\lambda}) \log(\sqrt{\lambda w_1^2 + 1}/2 + 1/2) - (1/\sqrt{\lambda}) \log(\sqrt{\lambda w_1^2 + 1}/2 + 1/2) \leq b$, where b is a parameter to make the boundary pass the $(0, 1)$ point for easy comparison with the ℓ_2 and ℓ_1 curves. It is easy to show that b equals to $\sqrt{1/\lambda} + \sqrt{1 + 1/\lambda} - (1/\sqrt{\lambda}) \log(\sqrt{\lambda + 1}/2 + 1/2)$. It can be seen that the ℓ_1-norm boundary has sharp turning points when it passes the axes, whereas the ℓ_2 and KL-norm boundaries turn smoothly at those points. This is the intuitive explanation of why the ℓ_1-norm directly gives sparse estimators, whereas the ℓ_2-norm and KL-norm due to a Laplace prior do not. But as shown in Figure 1, when the λ gets larger and larger, the KL-norm boundary moves closer and closer to the ℓ_1-norm boundary. When $\lambda \to \infty$, $\sqrt{w_1^2 + 1/\lambda} + \sqrt{w_2^2 + 1/\lambda} - (1/\sqrt{\lambda}) \log(\sqrt{\lambda w_1^2 + 1}/2 + 1/2) - (1/\sqrt{\lambda}) \log(\sqrt{\lambda w_1^2 + 1}/2 + 1/2) \to |w_1| + |w_2|$ and $b \to 1$, which yields exactly the ℓ_1-norm in the two dimensional space. Thus, under the linear model assumption of the discriminant functions $F(\,\cdot\,; \mathbf{w})$, the MaxEnDNet with a Laplace prior (i.e., the LapM^{3}N) can be seen as a smooth relaxation of the ℓ_1-M^{3}N.

3.2. ℓ_1-M^{3}N is an Adaptive M^{3}N

For the ℓ_1-M^{3}N and adaptive M^{3}N, we have the following equivalence theorem:

Theorem 4 *The AdapM^{3}N yields the same estimate as the ℓ_1-M^{3}N.*

[3]The curves are drawn with a symbolic computational package to solve an equation of the form: $2x - \log x = a$, where x is the variable to be solved and a is a constant.

Basically, our proof follows a similar technique as in (Grandvalet, 1998), where an equivalence between adaptive regression and the ℓ_1-regularized least square regression (LASSO) (Tibshirani, 1996) is proved.

4. EM-Style Learning of ℓ_1-M^{3}N

Based on the theorem 4, we develop a novel algorithm to approximately solve the ℓ_1-M^{3}N. As we shall see, this algorithm provides another perspective on the connection and difference between the ℓ_1-M^{3}N and LapM^{3}N. The algorithm iteratively solves the following two steps until a local optimum is arrived:

Step 1: keep τ fixed, optimize P3 over $(\mathbf{w}, \xi)$. This is an ℓ_2-norm M^{3}N problem:

$$\min_{\mathbf{w},\xi} \ \mathbf{w}^\top \Sigma^{-1} \mathbf{w} + C \sum_{i=1}^{N} \xi_i$$
$$\text{s.t. } \forall i, \ \forall \mathbf{y} \neq \mathbf{y}^i: \quad \mathbf{w}^\top \Delta \mathbf{f}_i(\mathbf{y}) \geq \Delta \ell_i(\mathbf{y}) - \xi_i; \ \xi_i \geq 0,$$

Step 2: keep $(\mathbf{w}, \xi)$ fixed, optimize P3 over τ. The problem reduces to:

$$\min_{\tau} \ \mathbf{w}^\top \Sigma^{-1} \mathbf{w},$$
$$\text{s.t. } \forall k: \quad \frac{1}{K} \sum_{k=1}^{K} \tau_k = \frac{1}{\lambda}; \tau_k \geq 0.$$

By forming a Lagrangian and doing some algebra, it is easy to show that the solution is:

$$\forall k: \quad \tau_k = \frac{K|w_k|}{\lambda \sum_k |w_k|} \tag{4}$$

Note that when $w_k = 0$, $\tau_k = 0$ and the corresponding feature will be discarded in the final estimate.

4.1. Connection to the Variational LapM^{3}N

As shown in (Zhu et al., 2008b), exact calculation leads to a normalization factor of LapM^{3}N that couples all the dual variables. Thus, the problem of LapM^{3}N is hard to be directly optimized. In (Zhu et al., 2008b), an efficient variational algorithm was developed to learn the LapM^{3}N, as recaped below.

Let $p(\mathbf{w}|\tau) = \prod_{k=1}^{K} p(w_k|\tau_k)$, $p(\tau|\lambda) = \prod_{k=1}^{K} p(\tau_k|\lambda)$ and $\mathrm{d}\tau \triangleq \mathrm{d}\tau_1 \cdots \mathrm{d}\tau_K$, then the multivariate independent Laplace prior is $p_0(\mathbf{w}) = \int p(\mathbf{w}|\tau) p(\tau|\lambda) \, \mathrm{d}\tau$ by the two-layer interpretation of a Laplace distribution. By applying the Jensen's inequality, an upper bound of the KL-divergence in LapM^{3}N is achieved,

$$\mathcal{L}(p(\mathbf{w}), q(\tau)) = -H(p) - \langle \int q(\tau) \log \frac{p(\mathbf{w}|\tau) p(\tau|\lambda)}{q(\tau)} \, \mathrm{d}\tau \rangle_p,$$

where $q(\tau)$ is a variational distribution to approximate $p(\tau|\lambda)$. Substituting this upper bound for the KL in LapM^{3}N, the variational method alternatively optimizes over $(p(\mathbf{w}), \xi)$ and $q(\tau)$ in two steps:

Step 1: solve the following problem to get the posterior mean μ of $\mathbf{w}$:

$$\min_{\mu,\xi} \ \frac{1}{2}\mu^\top \Sigma^{-1}\mu + C\sum_{i=1}^{N}\xi_i$$

$$\text{s.t. } \forall i,\ \forall \mathbf{y} \neq \mathbf{y}^i: \ \ \mu^\top \Delta\mathbf{f}_i(\mathbf{y}) \geq \Delta\ell_i(\mathbf{y}) - \xi_i;\ \xi_i \geq 0,$$

where $\Sigma = \text{diag}(\langle\tau_k^{-1}\rangle_q^{-1}) = \langle\mathbf{w}\mathbf{w}^\top\rangle_p - \mu\mu^\top$ is a diagonal matrix.

Step 2: update the expectations $\langle\tau_k^{-1}\rangle_q$ as follows,

$$\forall k: \ \ \langle\frac{1}{\tau_k}\rangle_q^{new} = \sqrt{\frac{\lambda}{\langle w_k^2\rangle_p}} = \sqrt{\frac{\lambda}{\mu_k^2 + 1/\langle\tau_k^{-1}\rangle_q^{(old)}}}. \quad (5)$$

It is obvious that the difference between the EM-style learning of the ℓ_1-M^{3}N and the variational learning of the LapM^{3}N is the second step in updating the adaptive parameters, i.e., τ in ℓ_1-M^{3}N and $\langle\tau^{-1}\rangle_q^{-1}$ in LapM^{3}N. The different update rules reflet the essential difference between the ℓ_1-M^{3}N and the LapM^{3}N. In Eq. (4), if $w_k = 0$, then $\tau_k = 0$. That means the corresponding feature will be discarded in the final estimate. However, in the LapM^{3}N, the update rule (5) ensures that $\langle\tau_k^{-1}\rangle_q^{-1}$ are always positive. Therefore, LapM^{3}N does not explicitly discard features even though the variances can be very small. This observation (approximately) explains why the ℓ_1-M^{3}N is primal sparse, while LapM^{3}N is pseudo-primal sparse.

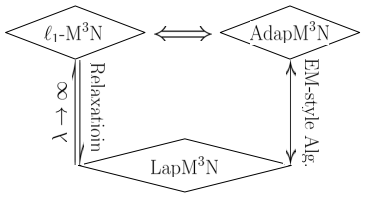


Figure 2. Relationships.

Figure 2 summarizes the relationships among the three variants of sparse M^{3}N. Basically, (1) the ℓ_1-M^{3}N is equivalent to the adaptive M^{3}N; (2) ℓ_1-M^{3}N is an extreme case of LapM^{3}N when $\lambda \to \infty$; (3) LapM^{3}N is a smooth relaxation of ℓ_1-M^{3}N; and (4) LapM^{3}N and adaptive M^{3}N share a similar EM-style algorithm.

5. Experiments

This section presents some empirical studies of the ℓ_1-M^{3}N and LapM^{3}N, compared with competing methods, including the primal sparse ℓ_1-norm regularized CRFs and the dual sparse ℓ_2-norm M^{3}N.

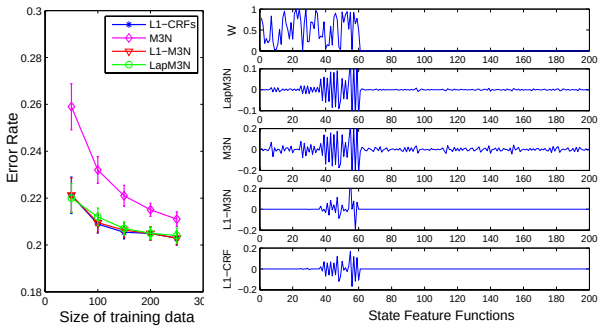
5.1. Evaluation on Synthetic Data

We follow the method as described in (Zhu et al., 2008b) to do the experiments. We generate sequence data sets, i.e., each input $\mathbf{x}$ is a sequence $(x_1, \cdots, x_L)$, and each component x_l is a d-dimensional vector of input features. The synthetic data are generated from pre-specified CRF models with either i.i.d. instantiations of the input features or correlated instantiations of the input features, from which samples of the structured output $\mathbf{y}$, i.e., a sequence $(y_1, \cdots, y_L)$, can be drawn from the conditional distribution $p(\mathbf{y}|\mathbf{x})$ defined by the CRF based on a Gibbs sampler.

Due to space limitation, we only report the results on the data sets with correlated input features. Conclusions in the i.i.d case are the same. Specifically, we set $d = 100$ and 30 input features are relevant to the output. The 30 relevant features are partitioned into 10 groups. For the features in each group, we first draw a real-value from a standard normal distribution and then corrupt the feature with a random Gaussian noise (zero mean and standard variance 0.05) to get 3 correlated features. Then, we generate 10 linear-chain CRFs with 8 binary states (i.e., $L = 8$ and $\mathcal{Y}_l = \{0, 1\}$). The feature functions include: 200 real valued state-feature functions, of which each is over a one-dimensional input feature and a class label; and 4 (2×2) transition feature functions capturing pairwise label dependencies. Each CRF is used to generate a data set that contains 1000 instances.

We do K-fold cross-validation on each data set and take the average over the 10 data sets as the final results. In each run we choose one part to do training and test on the rest $K-1$ parts. K is changed from 20, 10, 7, 5, to 4. In other words, we use 50, 100, about 150, 200, and 250 samples during the training. Figure 3(a) shows the performance. We can see that the primal sparse models (i.e., ℓ_1-M^{3}N and ℓ_1-CRFs) outperform the M^{3}N, which is only dual sparse, when the underlying model is primal sparse. As we have shown, the pseudo-sparse LapM^{3}N is a smooth relaxation of the ℓ_1-M^{3}N. If we choose a large regularization constant, LapM^{3}N will shrink the weights of irrelevant features to be extremely small. Thus, the LapM^{3}N performs similarly to the primal-sparse models.

Figure 3(b) shows the average weights of different models doing 10-fold CV on the first data set and the weights of the CRF model (first plot) that generates this data set. For LapM^{3}N and M^{3}N, all the weights are non-zero, although the weights of LapM^{3}N are generally much smaller than those of M^{3}N because of a shrinkage effect (Zhu et al., 2008b). For ℓ_1-M^{3}N and ℓ_1-CRFs, the estimates are sparse. Both of them can discard all the noise features when choosing an appropriate regularization constant. As shown in (Zhu et al., 2008b), ℓ_1-CRFs are very sensitive to the regularization constant. As we shall see the ℓ_1-M^{3}N with the EM-style algorithm is very robust. Note that all the models have quite different average weights from the model that generates the data. This is because we use a stochastic procedure (i.e., Gibbs sampler) to assign labels to the generated data samples. In fact, if we use the model that generates the data to pre-



(a) (b)

Figure 3. (a) Error rates and (b) average weights of different models.

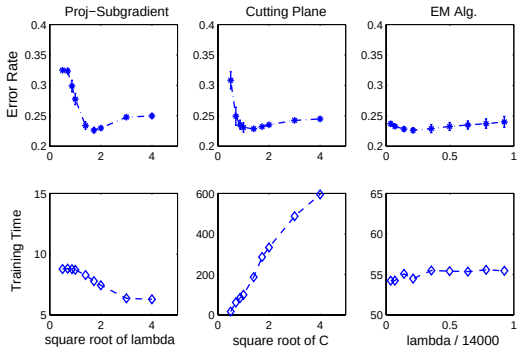


Figure 4. Error rates and training time of different algorithms for ℓ_1-M^{3}N on the first generated data set.

dict on its generated data, the error rate is about 0.5. Thus, the learned models, which get higher accuracy, are different from the model that generates the data.

Figure 4 shows the error rate and training time of three algorithms for ℓ_1-M^{3}N: projected sub-gradient (Ratliff et al., 2007) (based on P2′), cutting plane (Tsochantaridis et al., 2004) (based on P2), and our EM algorithm (based on P3, where C is kept fixed), doing 10-fold CV on the first data set. Since the regularization constants for different algorithms are generally incomparable, we select a set of values around the best one we have tried for each method (exact values shown in Table 1). We can see that both the sub-gradient and cutting plane methods are sensitive to their regularization constants. For the sub-gradient method, a projection to ℓ_1-ball (Duchi et al., 2008) is performed in each iteration, and the time depends largely on the ℓ_1-ball's radius. For larger balls, the projection is easier. So, the training time decreases as λ increases. For the cutting plane method, the time is mainly dependent on the LP solver (e.g., MOSEK as we use) and increases very fast as C gets larger. For the EM-algorithm, both the error rate and training time are stable as λ changes. We use 15 EM-iterations in these experiments and each iteration takes about 3.7 cpu-seconds, less than the time of the sub-gradient method.

Table 1 shows the number of non-zero average weights

Table 1. The number of non-zero average weights by different algorithms doing 10-fold CV on the first data set.

Proj-Subgradient	$\sqrt{\lambda}$	0.5	0.7	0.87	1	1.41	1.73	2	3	4
	Irrelevant	138	136	140	140	140	140	140	140	140
	Total	198	196	200	200	200	200	200	200	200
Cutting Plane	$\sqrt{C}$	0.5	0.7	0.87	1	1.41	1.73	2	3	4
	Irrelevant	0	0	2	16	103	122	134	139	140
	Total	13	21	26	44	145	169	184	186	189
EM Alg.	$\frac{\lambda}{14000}$	0.036	0.069	0.14	0.21	0.35	0.5	0.64	0.78	0.93
	Irrelevant	140	140	128	108	48	18	2	0	0
	Total	200	198	182	158	90	54	34	32	28

of (Total) all the state-feature functions and (Irrelevant) the state-feature functions based on irrelevant input features. In EM, we set $\tau = 0$ if it is less than 10^{-4}. We can see the EM algorithm has similar numbers of non-zero weights as the cutting-plane method. However, the projected sub-gradient method keeps many features, whose weights are small but not exactly zero, and truncating the feature weights with the same threshold as in EM doesn't change the sparse pattern much. Maybe tuning the learning rate could make this tail of very small features disappear.

5.2. Web Data Extraction

Web data extraction is a task to identify interested information from web pages. Each sample is a data record or an entire web page which is represented as a set of HTML elements. One striking characteristic of web data extraction is that various types of structural dependencies between HTML elements exist, e.g. the HTML tag tree is itself hierarchical. In (Zhu et al., 2008a), hierarchical CRFs are shown to achieve better performance than flat models like linear-chain CRFs (Lafferty et al., 2001). One method to construct a hierarchical model is to first use a parser to construct a so called vision tree. Then, based on the vision tree, a hierarchical model can be constructed accordingly to extract the interested attributes. See (Zhu et al., 2008a) for an example of the vision tree and the corresponding hierarchical model.

We use the data set that is built with web pages generated by 37 different templates (Zhu et al., 2008a) and extract the *Name*, *Image*, *Price*, and *Description* for each product. For each template, there are 5 pages for training and 10 for testing. Here, we assume that data records are given, and compare different hierarchical models on extracting attributes in the given records. There are 1585 and 3391 data records in the training and testing pages, respectively. We use the two comprehensive evaluation measures, i.e. average F1 and block instance accuracy (Zhu et al., 2008a). Average F1 is the average value of the F1 scores of the four attributes, and block instance accuracy is the percent of data records whose *Name*, *Image*, and *Price* are all correctly identified. On this data set, the cutting-plane method is too slow, and both the sub-gradient and EM

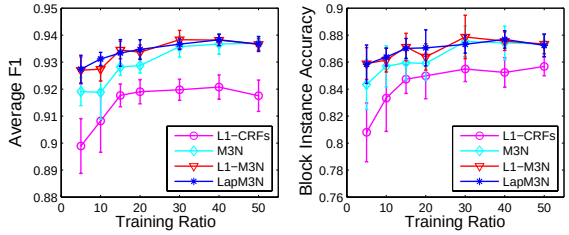


Figure 5. Average F1 and block instance accuracy on web data extraction with different number of training data.

algorithms are efficient and have similar performance.

We randomly select m (5, 10, 15, 20, 30, 40, or, 50) percent of the training records for training and test on all the testing records. For each m, 10 independent experiments are conducted and the average performance is summarized in Figure 5. We can see that: LapM^{3}N performs comparably with ℓ_1-M^{3}N, which enjoys both dual and primal sparsity, and outperforms the other models which enjoy either dual sparsity (i.e., M^{3}N) or primal sparsity (i.e., ℓ_1-CRFs), especially when the number of training data is small. The better performance of ℓ_1-M^{3}N compared to ℓ_1-CRFs demonstrates the promise of primal-sparse max-margin models.

6. Conclusion

We have presented the ℓ_1-norm max-margin Markov network (ℓ_1-M^{3}N) and a novel adaptive M^{3}N (AdapM^{3}N), which enjoy both primal and dual sparsity, and analyzed their close connections to the Laplace M^{3}N (LapM^{3}N), which is pseudo-primal sparse due to a smooth shrinkage effect. We show that ℓ_1-M^{3}N is an extreme case of LapM^{3}N, and ℓ_1-M^{3}N is equivalent to an AdapM^{3}N. We also develop a robust EM-style algorithm to learn an ℓ_1-M^{3}N. The algorithm helps uncover the difference between ℓ_1-M^{3}N and LapM^{3}N. On both synthetic and real web data, we show the promise of simultaneously (pseudo-) primal and dual sparse models over the competing ones which enjoy either dual or primal sparsity.

Acknowledgements

This work was done while J.Z. was a visiting researcher at CMU under a support from NSF DBI-0546594 and DBI-0640543 awarded to E.X.; J.Z. is also supported by Chinese NSF Grant 60621062 and 60605003; National Key Foundation R&D Projects 2003CB317007, 2004CB318108 and 2007CB311003; and Basic Research Foundation of Tsinghua National TNList Lab.

References

Altun, Y., Tsochantaridis, I., & Hofmann, T. (2003). Hidden Markov support vector machines. *International Conference on Machine Learning*, 3–10.

Bartlett, P, Collins, M, Taskar, B & McAllester, D (2004). Exponentiated gradient algorithms for large margin structured classification. *NIPS*, 113–120.

Bradley, D., & Bagnell, A. (2008). Differentiable sparse coding. *Neur. Info. Proc. Sys.*, 113–120.

Duchi, J., Shalev-Shwartz, S., Singer, Y., & Chandra, T. (2008). Efficient projection onto the ℓ_1-ball for learning in high dimensions. *ICML*, 272–279.

Figueiredo, M. (2003). Adaptive sparseness for supervised learning. *IEEE Trans. on Pattern Analysis and Machince Intelligence*, *25*, 1150–1159.

Friedman, J., Hastie, T., Rosset, S., Tibshirani, R., & Zhu, J. (2004). Discussion of boosting papers. *Annals of Statistics*, 102–107.

Grandvalet, Y. (1998). Least absolute shrinkage is equivalent to quadratic penalization. *Inter. Conf. on Aartifical Neural Networks*, 201–206.

Lafferty, J., McCallum, A., & Pereira, F. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *International Conference on Machine Learning*, 282–289.

Lee, S.-I., Ganapathi, V., & Koller, D. (2006). Efficient structure learning of Markov networks using ℓ_1-regularization. *Neur. Info. Proc. Sys.*, 817–824.

Qi, Y., Minka, T., Picard, R., & Ghahramani, Z. (2004). Predictive automatic relevance determination by expectation propagation. *ICML*, 671–678.

Ratliff, N. D., Bagnell, J. A., & Zinkevich, M. A. (2007). (Online) Subgradient methods for structured prediction. *AI Statistics*, .

Smola, A., Vishwanathan, S., & Le, Q. (2007). Bundle methods for machine learning. *NIPS*, 1377–1384.

Taskar, B., Guestrin, C., & Koller, D. (2003). Max-margin Markov networks. *Neur. Info. Proc. Sys.*.

Taskar, B., Lacoste-Julien, S., & Jordan, M. I. (2006). Structured prediction via the extragradient method. *Neur. Info. Proc. Sys.*, 1345–1352.

Tibshirani, R. (1996). Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Soceity, Series B*, 267–288.

Tipping, M. E. (2001). Sparse bayesian learning and the relevance vector machine. *JMLR*, 211–244.

Tsochantaridis, I., Hofmann, T., Joachims, T., & Altun, Y. (2004). Support vector machine learning for interdependent and structured output spaces. *Inter. Conf. on Mach. Learn.*, 823–830.

Wainwright, M., Ravikumar, P., & Lafferty, J. (2006). High-dimensional graphical model selection using ℓ_1-regularized logistic regression. *NIPS*, 1465–1472.

Zhu, J., Nie, Z., Zhang, B., & Wen, J.-R. (2008a). Dynamic hierarchical Markov random fields for integrated web data extraction. *JMLR*, 1583–1614.

Zhu, J., Rosset, S., Hastie, T., & Tibshirani, R. (2004). 1-norm support vector machines. *Advances in Neural Information Processing Systems*, 49–56.

Zhu, J., & Xing, E. (2009). Maximum entropy discrimination Markov networks. *ArXiv 0901.2730*.

Zhu, J., Xing, E., & Zhang, B. (2008b). Laplace maximum margin Markov networks. *ICML*, 1977–1984.