Advanced Structured Prediction

Editors:

Tamir Hazan tamir.hazan@technion.ac.il
Technion - Israel Institute of Technology
Technion City, Haifa 32000, Israel

George Papandreou gpapan@google.com
Google Inc.
340 Main St., Los Angeles, CA 90291 USA

Daniel Tarlow dtarlow@microsoft.com
Microsoft Research
Cambridge, CB1 2FB, United Kingdom

This is a draft version of the author chapter.

The MIT Press
Cambridge, Massachusetts
London, England

1 Perturbation Models and PAC-Bayesian Generalization Bounds

Joseph Keshet joseph.keshet@biu.ac.il
Bar-Ilan University
Ramat-Gan, Israel

Subhransu Maji smaji@cs.umass.edu
University of Massachusetts Amherst
Amherst, MA, USA

Tamir Hazan tamir.hazan@technion.ac.il
Technion - Israel Institute of Technology
Haifa, Israel

Tommi Jaakkola tommi@csail.mit.edu
Massachusetts Institute of Technology - MIT
Cambridge, MA, USA

In this chapter we explore the generalization power of perturbation models. Learning parameters that minimize the expected task loss of perturbation models amounts to minimizing PAC-Bayesian generalization bounds. We provide an elementary derivation of PAC-Bayesian generalization bounds, while focusing on their Bayesian components, namely their predictive probabilities and their posterior distributions. We connect their predictive probabilities to perturbation models and their posterior distributions to the smoothness of the PAC-Bayesian bound. Consequently, we derive algorithms that minimize PAC-Bayesian generalization bounds using stochastic gradient descent and explore their effectiveness on speech and visual recognition tasks.

1.1 Introduction

Learning and inference in complex models drives much of the research in machine learning applications ranging from computer vision to natural language processing to computational biology (Blake et al., 2004; Rush and Collins; Sontag et al., 2008). Each such task has its own measure of performance, such as the intersection-over-union score in visual object segmentation, the BLEU score in machine translation, the word error rate in speech recognition, the NDCG score in information retrieval, and so on. The *inference* problem in such cases involves assessing the likelihood of possible structured-labels, whether they be objects, parsers, or molecular structures. Given a training dataset of instances and labels, the *learning* problem amounts to estimation of the parameters of the inference engine, so as to minimize the desired measure of performance, or *task loss*.

The structures of labels are specified by assignments of random variables, and the likelihood of the assignments are described by a potential function. Usually it is only feasible to infer the most likely or maximum a-posteriori (MAP) assignment, rather than sampling according to their likelihood. Indeed, substantial effort has gone into developing inference algorithms for predicting MAP assignments, either based on specific parametrized restrictions such as super-modularity (e.g., Boykov et al., 2001) or by devising approximate methods based on linear programming relaxations (e.g., Sontag et al., 2008).

Learning the parameters of the potential function greatly influences the prediction accuracy. In supervised settings, the learning algorithm is provided with training data which is composed of pairs of data instances and their labels. For example, data instances can be images or sentences and their labels may be the foreground-background segmentation of these images or the correct translations of these sentences. The goal of the learning procedure is to find the potential function for which its MAP prediction for a training data instance is the same as its paired training label. The goodness of fit between the MAP predicted label and the training label is measured by a loss function. Unfortunately, the prediction function is non-smooth as well as non-convex and direct task loss minimization is hard in practice (McAllester et al., 2010).

To overcome the shortcomings of direct task loss minimization, the task loss function is replaced with a surrogate loss function. There are various surrogate loss functions, some of them are convex (and non-smooth), while others are smooth (and non-convex). The structured hinge loss, a convex upper bound to the task loss, is the surrogate loss function used both in

max-margin Markov models (Taskar et al., 2004) and in structural SVMs (Tsochantaridis et al., 2006). Unfortunately, the the error rate of the structured hinge loss minimizer does not converge to the to the error rate of the Bayesian optimal linear predictor in the limit of infinite training data, even when the task loss is the 0-1 loss (McAllester, 2006; Tewari and Bartlett, 2007). The structured ramp loss (Do et al., 2008) is another surrogate loss function that proposes a tighter bound to the task loss than the structured hinge loss. In contrast to the hinge loss, the structured ramp loss was shown to be strongly consistent (McAllester and Keshet, 2011). In general both the hinge loss and the structured ramp loss functions require the task loss function to be decomposable in the size of the output label. Decomposable task loss functions are required in order to solve the loss-augmented inference that is used within the training procedure (Ranjbar et al., 2013), and evaluation metrics like intersection-over-union or word error rate, which are not decomposable, need to be approximated when utilized in these training methods.

Conditional random fields (Lafferty et al., 2001) utilizes the negative log-likelihood as a surrogate loss function. Minimizing this loss amounts to maximizing the log-likelihood of the conditional Gibbs distribution of the training data. While this is a convex function with a nice probabilistic properties, it is unrelated to the task loss, and hence not expected to optimize the risk. Alternatively, one may integrate the task loss function by minimizing the expected loss, while averaging with respect to the Gibbs distribution (Gimpel and Smith, 2010). This approach is computationally appealing since it effortlessly deal with non-decomposable loss functions, while shifting the computational burden to sampling from the Gibbs distribution. Unfortunately, sampling from the Gibbs distribution is provably hard Jerrum and Sinclair (1993); Goldberg and Jerrum (2007)

Recently, several works (Keshet et al., 2011; Papandreou and Yuille, 2011; Tarlow et al., 2012) have constructed probability models through MAP predictions. These "perturb-max" models describe the robustness of the MAP prediction to random changes of its parameters. Therefore, one can draw unbiased samples from these distributions using MAP predictions. Interestingly, when using perturbation models to compute the expected loss minimization one would ultimately minimizes PAC-Bayesian generalization bounds (McAllester, 2003; Langford and Shawe-Taylor, 2002; Seeger, 2003; Catoni, 2007; Germain et al., 2009; Keshet et al., 2011; Seldin et al., 2012).

This chapter explores the Bayesian aspects that emerge from PAC-Bayesian generalization bounds. We focus on their predictive probability models, which turn to be perturbation models as well as on PAC-Bayesian posterior distributions. We also focus on its algorithmic aspects, both of the

predictive probability and the posterior distribution, so that they could be used to minimize the risk bound efficiently. We demonstrate the effectiveness of minimizing these bounds on visual and speech recognition problems.

1.2 Background

Learning complex models typically involves reasoning about the states of discrete variables whose labels (assignments of values) specify the discrete structures of interest. The learning task which we consider in this work is to fit parameters $\boldsymbol{w}$ that produce the most accurate prediction $y \in \mathcal{Y}$ for a given object x. Structures of labels are conveniently described by a discrete product space $\mathcal{Y} = \mathcal{Y}_1 \times \cdots \times \mathcal{Y}_n$. We describe the potential of relating a label y to an object x with respect to the parameters w by real valued functions $\theta(y; x, w)$. Maximum a-posteriori prediction amounts to compute the best scoring label:

$$\text{(MAP predictor)} \qquad \hat{y}_w(x) = \arg\max_y \ \theta(y; x, w), \tag{1.1}$$

where $y = (y_1, ..., y_n)$.

We measure the goodness of fit by a loss function $L : \mathcal{Y} \times \mathcal{Y} \to [0, 1]$. The loss of the MAP predictor for an object-label pair is $L(\hat{y}_w(x), y)$. We assume that the object-label pairs in the world are distributed according to an unknown distribution $\mathcal{D}$. The risk of the MAP predictor that is parametrized by w, denoted by $R(w)$ is the expected loss

$$R(w) = \mathbb{E}_{(x,y)\sim\mathcal{D}}\Big[L(\hat{y}_w(x), y)\Big] \tag{1.2}$$

Our goal is to learn the parameters w and consequently their predictor $\hat{y}_w(x)$ which minimizes the risk, that is,

$$w^* = \arg\min_w \ \mathbb{E}_{(x,y)\sim\mathcal{D}}\Big[L(\hat{y}_w(x), y)\Big]. \tag{1.3}$$

Since the distribution $\mathcal{D}$ is unknown, we use a training dataset S of independent and identically distributed (i.i.d.) samples of pairs (x, y) from $\mathcal{D}$. We then define the empirical risk to be

$$R_S(w) = \mathbb{E}_{(x,y)\sim S}\Big[L(\hat{y}_w(x), y)\Big] = \frac{1}{|S|}\sum_{(x,y)\in S} L(\hat{y}_w(x), y) \tag{1.4}$$

A direct minimization of the empirical risk is computationally unappealing as it is a non-smooth and non-convex function of w. Alternatively, the loss function in the empirical risk is replaced with a surrogate loss, and an additional regularization term is added to avoid overfitting of the parameters

and add stability. The objective of the learning procedure is therefore

$$w^* = \arg\min_w \ \mathbb{E}_{(x,y)\sim S}\Big[L(\hat{y}_w(x), y)\Big] + \lambda\,\Omega(w), \tag{1.5}$$

where $\Omega(w)$ is a regularization function and λ is a trade-off parameter.

It is possible to decrease the empirical risk by upper bounding the task loss function with a convex surrogate, as applied in structured-SVM that is governed by the hinge-loss:

$$L_{hinge}(x, y, w) = \max_{\hat{y}\in Y} \ \{L(\hat{y}, y) + \theta(\hat{y}; x, w) - \theta(y; x, w)\} \tag{1.6}$$

It is straightforward to verify that the hinge-loss $L_{hinge}(x, y, w)$ upper bounds the task loss $L(\hat{y}_w(x), y)$ since

$$L(\hat{y}_w(x), y) \le L(\hat{y}_w(x), y) + \theta(\hat{y}_w(x); x, w) - \theta(y; x, w) \le L_{hinge}(x, y, w).$$

Moreover, the hinge-loss is a convex function of w as it is a maximum of linear functions of w. The hinge-loss leads to "loss adjusted inference" since computing its value requires more than just MAP inference $\hat{y}_w(x)$. In particular, when the loss function is more involved than the MAP prediction, as happens in computer vision problems (e.g., PASCAL VOC loss) or language processing tasks (e.g., BLEU loss), learning with structured-SVMs is computationally hard.

The prediction $\hat{y}_w(x)$ as well as "loss adjusted inference" rely on the potential structure to compute the MAP assignment. Potential functions are conveniently described by a family R of subsets of variables $r \subset \{1, ..., n\}$, called regions. We denote by y_r the set of labels that correspond to the region r, namely $(y_i)_{i\in r}$ and consider the following potential functions $\theta(y; x, w) = \sum_{r\in R} \theta_r(y_r; x, w)$. Thus, MAP prediction can be formulated as an integer linear program:

$$b^* \in \arg\max_{b_r(y_r)} \sum_{r, y_r} b_r(y_r)\theta_r(y_r; x, w) \tag{1.7}$$

$$s.t. \quad b_r(y_r) \in \{0, 1\}, \ \sum_{y_r} b_r(y_r) = 1, \quad \sum_{y_s \setminus y_r} b_s(y_s) = b_r(y_r) \quad \forall r \subset s$$

The correspondence between MAP prediction and integer linear program solutions is $(\hat{y}_w(x))_i = \arg\max_{y_i} b_i^*(y_i)$. Although integer linear program solvers provide an alternative to MAP prediction, they may be restricted to problems of small size. This restriction can be relaxed when one replaces the integral constraints $b_r(y_r) \in \{0, 1\}$ with nonnegative constraints $b_r(y_r) \ge 0$. These linear program relaxations can be solved efficiently using different convex max-product solvers, and whenever these solvers produce an integral solution it is guaranteed to be the MAP prediction (Sontag et al., 2008).

A substantial effort has been invested to solve this integer linear program in some special cases, particularly when $|r| \leq 2$. In this case, the potential function corresponds to a standard graph: $\theta(y; x, w) = \sum_{i \in V} \theta_i(y_i; x, w) + \sum_{i,j \in E} \theta_{i,j}(y_i, y_j; x, w)$. If the graph has no cycles, MAP prediction can be computed efficiently using the belief propagation algorithm Pearl (1988). There are cases where MAP prediction can be computed efficiently for graph with cycles. A potential function is called supermodular if it is defined over $\mathcal{Y} = \{-1, 1\}^n$ and its pairwise interactions favor adjacent states to have the same label, i.e., $\theta_{i,j}(-1, -1; x, w) + \theta_{i,j}(1, 1; x, w) \geq \theta_{i,j}(-1, 1; x, w) + \theta_{i,j}(1, -1; x, w)$. In such cases MAP prediction reduces to computing the min-cut (graph-cuts) algorithm.

1.3 PAC-Bayesian Generalization Bounds

The PAC-Bayesian generalization bound asserts that the overall risk of predicting w can be estimated by the empirical risk over a finite training set. This is essentially a measure concentration theorem: the expected value (risk) can be estimated by its (empirical) sampled mean. Given an object-label sample $(x, y) \sim \mathcal{D}$, the loss function $L(\hat{y}_w(x), y)$ turns out to be a bounded random variable in the interval $[0, 1]$. In the following we assume that the training data $S = \{(x_1, y_1), ..., (x_m, y_m)\}$ is sampled i.i.d. from the distribution $\mathcal{D}$, and is denoted by $S \sim \mathcal{D}^m$. The measure concentration of a sampled average is then described by the moment generating function, also known as the Hoeffding lemma:

$$\mathbb{E}_{S \sim \mathcal{D}^m} \Big[\exp \Big(\sigma \left(R(w) - R_S(w) \right) \Big) \Big] \leq \exp(\sigma^2 / 8m), \tag{1.8}$$

for all $\sigma \in \mathbb{R}$.

We average over all possible parameters and therefore take into account all possible predictions $\hat{y}_w(x)$:

Lemma 1.1. *Let $L(\hat{y}, y) \in [0, 1]$ be a bounded loss function. Let $p(w)$ be any probability density function over the space of parameters. Then, for any positive number $\sigma > 0$ holds*

$$\mathbb{E}_{S \sim \mathcal{D}^m} \mathbb{E}_{w \sim p} \Big[\exp \Big(\sigma (R(w) - R_S(w)) \Big) \Big] \leq \exp(\sigma^2 / 8m) \tag{1.9}$$

The above bound measures the expected (exponentiated) risk of Gibbs predictors. Gibbs predictors $\hat{y}_w(x)$ are randomized predictors, determined by $w \sim p$. The probability distribution $p(w)$ is determined before seeing the training data and is therefore considered to be a prior distribution over the parameters. $p(w)$ may be any probability distribution over the space of

parameters and it determines the amount of influence of any parameter w to the overall expected risk. Therefore when computing the expected risk it also takes into account the desired parameters w^*, which are intuitively the risk minimizer. For example, the prior distribution may be the centered normal distribution $p(w) \propto \exp(\|w\|^2/2)$. Since a centered normal distribution is defined for every w, it also assigns a weight to w^*. However, the centered normal distribution rapidly decays outside of a small radius around the center, and if the desired parameters w^* are far from the center, the above expected risk bound only consider a negligible part of it.

The core idea of PAC-Bayesian theory is to shift the Gibbs classifier to be centered around the desired parameters w^*. Since these parameters are unknown, the PAC-Bayesian theory applies to all possible parameters u. Such bounds are called uniform.

Lemma 1.2. *Consider the setting of Lemma 1.1. Let $q_u(w)$ be any probability density function over the space of parameters with expectation u. Let $D_{\mathrm{KL}}(q_u||q) = \int q_u(w) \log(q_u(w)/p(w))dw$ be the KL-divergence between two distributions. Then, for any set $S = \{(x_1, y_1), ..., (x_m, y_m)\}$ the following holds simultaneously for all u:*

$$\mathbb{E}_{w\sim p}\Big[\exp\big(R(w) - R_S(w)\big)\Big] \geq \exp\Big(\mathbb{E}_{w\sim q_u}[R(w) - R_S(w)] - D_{\mathrm{KL}}(q_u||p)\Big) \tag{1.10}$$

Proof. The proof includes two steps. The first step transfers the prior $p(w)$ to the posterior $q_u(w)$. To simplify the notation we omit the subscript of the posterior distribution, writing it as $q(w)$.

$$\mathbb{E}_{w\sim p}\Big[\exp\big(R(w) - R_S(w)\big)\Big] = \mathbb{E}_{w\sim q}\Big[\frac{p(w)}{q(w)}\exp\big(R(w) - R_S(w)\big)\Big] \tag{1.11}$$

We move the ratio $p(w)/q(w)$ to the exponent, thus the right hand-side equals

$$\mathbb{E}_{w\sim q}\Big[\exp\big(R(w) - R_S(w) - \log\frac{q(w)}{p(w)}\big)\Big] \tag{1.12}$$

The second step of the proof uses the convexity of the exponent function to derive a lower bound to this quantity with

$$\exp\Big(\mathbb{E}_{w\sim q}[R(w) - R_S(w)] - \mathbb{E}_{w\sim q}[\log(q(w)/p(w))]\Big). \tag{1.13}$$

The proof then follows from the definition of the KL-divergence as the expectation of $\log(q(w)/p(w))$. □

We omit σ from Lemma 1.2 to simplify the notation. The same proof holds for $\sigma(R(w) - R_S(w))$, for any positive σ. The lemma holds for any S, thus

also holds in expectation, i.e., when taking expectations on both sides of the inequality. Combining both lemmas above we get

$$\mathbb{E}_{S\sim\mathcal{D}^m}\Big[\exp\Big(\mathbb{E}_{w\sim q_u}[\sigma(R(w)-R_S(w))]-D_{\mathrm{KL}}(q_u||p)\Big)\Big] \leq \exp(\sigma^2/8m) \quad (1.14)$$

This bound holds uniformly (simultaneously) for all u and particularly to the (empirical) risk minimizer w^*. This bound holds in expectation over the samples of training sets. It implies a similar bound that holds in high probability via Markov inequality:

Theorem 1.3. *Consider the setting of the above Lemmas. Then, for any $\delta \in (0,1]$ and for any real number $\lambda > 0$, with a probability of at least $1-\delta$ over the draw of the training set, the following holds simultaneously for all u*

$$\mathbb{E}_{w\sim q_u}\Big[R(w)\Big] \;\leq\; \mathbb{E}_{w\sim q_u}\Big[R_S(w)\Big] + \lambda D_{\mathrm{KL}}(q_u||p) + \frac{1}{\lambda\sqrt{8m}} + \lambda\log\frac{1}{\delta} \quad (1.15)$$

Proof. Markov inequality asserts that $Pr[Z \leq EZ/\delta] \geq 1-\delta$. The theorem follows by setting $Z = \exp\Big(\mathbb{E}_{w\sim q_u}[\lambda(R(w)-R_S(w))]-D_{\mathrm{KL}}(q_u||p)]\Big)$ and using Equation (1.14). □

The above bound is a standard PAC-Bayesian bound that appears in various versions in the literature (McAllester, 2003; Langford and Shawe-Taylor, 2002; Seeger, 2003; Catoni, 2007; Seldin, 2009; Germain et al., 2009; Keshet et al., 2011; Seldin et al., 2012).

1.4 Algorithms

Recall that our goal is to find the parameters that minimize the risk as in Equation (1.3). As we stated in (1.5), the empirical risk can be replaced by a surrogate loss function and a regularization term. In our case, the training objective is defined as follows

$$w^* = \arg\min_u \;\; \mathbb{E}_{w\sim q_u}\Big[R_S(w)\Big] \;+\; \lambda D_{\mathrm{KL}}(q_u||p), \quad (1.16)$$

where $D_{\mathrm{KL}}(q_u||p)$ is the regularization term, λ is the regularization parameter, and the surrogate loss is the generalized probit loss defined as

$$\mathbb{E}_{w\sim q_u}\Big[L(\hat{y}_w(x),y)\Big], \quad (1.17)$$

and can be derived from the linearity of the expectation and Equation (1.4). Note that the minimizer of the objective in Equation (1.16) is also the minimizer of the right-hand side of the bound in Equation (1.15).

We now turn to show that whenever the posterior distributions have smooth probability density functions $q_u(w)$, the perturbation probability model is as smooth as a function of u. Thus the randomized risk bound can be minimized with gradient methods to approach the desired u.

Theorem 1.4. *Assume $q_u(w)$ is as smooth as a function of its parameters, then the PAC-Bayesian bound is as smooth as a function of u:*

$$\nabla_u \mathbb{E}_{w\sim q_u}\Big[R_S(w)\Big] = \frac{1}{m}\sum_{(x,y)\in S} \mathbb{E}_{w\sim q_u}\Big[\nabla_u[\log q_u(w)]L(y_w(x),y)\Big]$$

Moreover, the KL-divergence is a smooth function of w and its gradient takes the form:

$$\nabla_u D_{\mathrm{KL}}(q_u||p) = \mathbb{E}_{w\sim q_u}\Big[\nabla_u[\log q_u(w)]\Big(\log(q_u(w)/p(w))+1\Big)\Big]$$

Proof. $\mathbb{E}_{w\sim q_u} R_S(w) = \frac{1}{m}\sum_{i=1}^m \int q_u(w)L(\hat{y}_w(x_i),y_i)dw$. Since $q_u(w)$ is a probability density function and $L(\hat{y},y)\in[0,1]$ we can differentiate under the integral (cf. Folland (1999) Theorem 2.27). The gradient is

$$\nabla_u \mathbb{E}_{w\sim q_u}\Big[R_S(w)\Big] = \frac{1}{m}\sum_{i=1}^m \int \nabla_u q_u(w)L(\hat{y}_w(x),y)dw. \tag{1.18}$$

Using the identity $\nabla_u q_u(w) = q_u(w)\nabla_u \log(q_u(w))$ the first part of the proof follows. The second part of the proof follows in the same manner, while noting that $\nabla_u(q_u(w)\log q_u(w)) = (\nabla_u q_u(w))(\log q_u(w)+1)$. □

The gradient of the randomized empirical risk is governed by the gradient of the log-probability density function of its corresponding posterior model. For example, Gaussian model with mean w and identity covariance matrix has the probability density function $q_u(w)\propto \exp(-\|w-u\|^2/2)$, thus the gradient of its log-density is the linear moment γ, i.e., $\nabla_u[\log q_u] = w-u$.

Taking any smooth distribution $q_u(w)$, we can find the parameters u by descending along the stochastic gradient of the PAC-Bayesian generalization bound. The gradient of the randomized empirical risk is formed by two expectations, over the sample points and over the posterior distribution. Computing these expectations is time consuming, thus we use a single sample $\nabla_u[\log q_u(w)]L(y_w(x),y)$ as an unbiased estimator for the gradient. Similarly we estimate the gradient of the KL-divergence with an unbiased estimator which requires a single sample of $\nabla_u[\log q_u(w)](\log(q_u(w)/p(w))+1)$. This approach, called stochastic approximation or online gradient descent,

amounts to use of the stochastic gradient update rule, where η is the learning rate. Next, we explore different posterior distributions from computational perspectives. Specifically, we show how to learn the posterior model so as to ensure the computational efficiency of its MAP predictor.

1.5 The Bayesian Perspective

PAC-Bayesian theory has a strong Bayesian ingredient. It integrates over uncertainty of its parameters using the posterior distribution. This important aspect guarantees a uniform generalization bound, over all possible posterior parameters. As a consequence of this theory, a new predictive distribution emerges, the perturbation model, that connects the posterior distribution to the task loss.

1.5.1 Predictive distribution

The PAC-Bayesian risk give rise to novel distribution models that involve optimization and perturbation. The risk averages over all parameters. $\mathbb{E}_{w\sim q_u}[R(w)] = \mathbb{E}_{w\sim q_u}[L(\hat{y}_w(x), y)]$. To reveal the underlining Bayesian model we aggregate all parameters w that result in the same prediction

$$p(y|x;u) = \mathbb{P}_{w\sim q_u}[y = \hat{y}_w(x)] \tag{1.19}$$

This novel probability distribution measures how much stable a prediction is under random perturbation of the parameters. The appealing property of this distribution is that unlike the Gibbs distribution, it is easy to draw unbiased samples for as long as optimizing is easy. Since this perturbation model is defined by perturbation and optimization it is also called perturb-max or perturb-and-map model.

1.5.2 Posterior distributions

The posterior distribution accounts for the space of parameters that can be learned. The ability to efficiently apply MAP predictors is key to the success of the learning process. Although MAP predictions are NP-hard in general, there are posterior models for which they can be computed efficiently. For example, whenever the potential function corresponds to a graphical model with no cycles, MAP prediction can be efficiently computed for any learned parameters w.

Learning unconstrained parameters with random MAP predictors provides some freedom in choosing the posterior distribution. In fact, Theorem 1.4

suggests that one can learn any posterior distribution by performing gradient descent on its risk bound, as long as its probability density function is smooth. We show that for unconstrained parameters, additive posterior distributions simplify the learning problem, and the complexity of the bound (i.e., its KL-divergence) mostly depends on its prior distribution.

Corollary 1.5. *Let $q_0(\gamma)$ be a smooth probability density function with zero mean and set the posterior distribution using additive shifts $q_w(\gamma) = q_0(\gamma - w)$. Let $H(q) = -\mathbb{E}_{\gamma\sim q}[\log q(\gamma)]$ be the entropy function. Then*

$$D_{\mathrm{KL}}(q_w||p) = -H(q_0) - \mathbb{E}_{\gamma\sim q_0}[\log p(\gamma + w)]$$

In particular, if $p(\gamma) \propto \exp(-\|\gamma\|^2)$ is Gaussian then $\nabla_w D_{\mathrm{KL}}(q_w||p) = w$

Proof: $D_{\mathrm{KL}}(q_w||p) = -H(q_w) - \mathbb{E}_{\gamma\sim q_w}[\log p(\gamma)]$. By a linear change of variable $\hat{\gamma} = \gamma - w$ it follows that $H(q_w) = H(q_0)$ thus $\nabla_w H(q_w) = 0$. Similarly $\mathbb{E}_{\gamma\sim q_w}[\log p(\gamma)] = \mathbb{E}_{\gamma\sim q_0}[\log p(\gamma + w)]$. Finally, if $p(\gamma)$ is Gaussian then $\mathbb{E}_{\gamma\sim q_0}[\log p(\gamma + w)] = -w^2 - \mathbb{E}_{\gamma\sim q_0}[\gamma^2]$. □

This result implies that every additively-shifted smooth posterior distribution may consider the KL-divergence penalty as the square regularization when using a Gaussian prior $p(\gamma) \propto \exp(-\|\gamma\|^2)$. This generalizes the standard claim on Gaussian posterior distributions Langford and Shawe-Taylor (2002), for which $q_0(\gamma)$ are Gaussians. Thus one can use different posterior distributions to better fit the randomized empirical risk without increasing the computational complexity over Gaussian processes.

Learning unconstrained parameters can be efficiently applied to tree structured graphical models. This, however, is restrictive. Many practical problems require more complex models, with many cycles. For some of these models linear program solvers give efficient, although sometimes approximate, MAP predictions. For supermodular models there are specific solvers, such as graph-cuts, that produce fast and accurate MAP predictions. In the following we show how to define posterior distributions that guarantee efficient predictions, thus allowing efficient sampling and learning.

MAP predictions can be computed efficiently in important practical cases, e.g., supermodular potential functions satisfying $\theta_{i,j}(-1,-1;x,w) + \theta_{i,j}(1,1;x,w) \geq \theta_{i,j}(-1,1;x,w) + \theta_{i,j}(1,-1;x,w)$. Whenever we restrict ourselves to symmetric potential function $\theta_{i,j}(y_i,y_j;x,w) = w_{i,j}y_iy_j$, supermodularity translates to nonnegative constraint on the parameters $w_{i,j} \geq 0$. In order to model posterior distributions that allow efficient sampling we define models over the constrained parameter space. Unfortunately, the additive posterior models $q_w(\gamma) = q_0(\gamma - w)$ are inappropriate for this purpose, as they have a positive probability for negative γ values and would generate

non-supermodular models.

To learn constrained parameters one requires posterior distributions that respect these constraints. For nonnegative parameters we apply posterior distributions that are defined on the nonnegative real numbers. We suggest the incorporation of the parameters of the posterior distribution in a multiplicative manner into a distribution over the nonnegative real numbers. For any distribution $q_\alpha(\gamma)$ we determine a posterior distribution with parameters w as $q_w(\gamma) = q_\alpha(\gamma/w)/w$. We show that multiplicative posterior models naturally provide log-barrier functions over the constrained set of nonnegative numbers. This property is important to the computational efficiency of the bound minimization algorithm.

Corollary 1.6. *For any probability distribution $q_\alpha(\gamma)$, let $q_{\alpha,w}(\gamma) = q_\alpha(\gamma/w)/w$ be the parametrized posterior distribution. Then*

$$D_{\mathrm{KL}}(q_{\alpha,w}||p) = -H(q_\alpha) - \log w - \mathbb{E}_{\gamma\sim q_\alpha}[\log p(w\gamma)]$$

Define the Gamma function $\Gamma(\alpha) = \int_0^\infty \gamma^{\alpha-1}\exp(-\gamma)$. If $p(\gamma) = q_\alpha(\gamma) = \gamma^{\alpha-1}\exp(-\gamma)/\Gamma(\alpha)$ have the Gamma distribution with parameter α, then $\mathbb{E}_{\gamma\sim q_\alpha}[\log p(w\gamma)] = (\alpha-1)\log w - \alpha w$. Alternatively, if $p(\gamma)$ are truncated Gaussians then $\mathbb{E}_{\gamma\sim q_\alpha}[\log p(w\gamma)] = -\frac{\alpha}{2}w^2 + \log\sqrt{\pi/2}$.

Proof: The entropy of multiplicative posterior models naturally implies the log-barrier function:

$$-H(q_{\alpha,w}) \stackrel{\hat\gamma=\gamma/w}{=} \int q_\alpha(\hat\gamma)\Big(\log q_\alpha(\hat\gamma) - \log w\Big)d\hat\gamma = -H(q_\alpha) - \log w.$$

Similarly, $\mathbb{E}_{\gamma\sim q_{\alpha,w}}[\log p(\gamma)] = \mathbb{E}_{\gamma\sim q_\alpha}[\log p(w\gamma)]$. The special cases for the Gamma and the truncated normal distribution follow by a direct computation. □

The multiplicative posterior distribution would provide the barrier function $-\log w$ as part of its KL-divergence. Thus the multiplicative posterior effortlessly enforces the constraints of its parameters. This property suggests that using multiplicative rules is computationally favorable. Interestingly, using a prior model with Gamma distribution adds to the barrier function a linear regularization term $\|w\|_1$ that encourages sparsity. On the other hand, a prior model with a truncated Gaussian adds a square regularization term which drifts the nonnegative parameters away from zero. A computational disadvantage of the Gaussian prior is that its barrier function cannot be controlled by a parameter α.

1.6 Approximate Inference

We may use the flexibility of Bayesian models to extend perturbation models beyond MAP prediction, as in the case of approximate inference. MAP prediction can be phrased as an integer linear program, stated in Equation (1.7). The computational burden of integer linear programs can be relaxed when one replaces the integral constraints with nonnegative constraints. This approach produces approximate MAP predictions. An important learning challenge is to extend the predictive distribution of perturbation models to incorporate approximate MAP solutions. Approximate MAP predictions are are described by the feasible set of their linear program relaxations which is usually called the local polytope:

$$L(R) = \Big\{ b_r(y_r) : \; b_r(y_r) \geq 0, \; \sum_{y_r} b_r(y_r) = 1, \; \forall r \subset s \; \sum_{y_s \setminus y_r} b_s(y_s) = b_r(y_r) \Big\}$$

Linear program solutions are usually the extreme points of their feasible polytope. The local polytope is defined by a finite set of equalities and inequalities, thus it has a finite number of extreme points. The predictive distribution that is defined in Equation (1.19) can be effortlessly extended to the finite set of the local polytope's extreme points. This approach has two flaws. First, linear program solutions might not be extreme points, and decoding such a point usually requires additional computational effort. Second, without describing the linear program solutions one cannot incorporate loss functions that take the structural properties of approximate MAP predictions into account when computing the randomized risk.

Theorem 1.7. *Consider approximate MAP predictions that arise from relaxation of the MAP prediction problem in Equation (1.7).*

$$\arg\max_{b_r(y_r)} \quad \sum_{r, y_r} b_r(y_r) \theta_r(y_r; x, w) \quad s.t. \quad b \in L(R)$$

Then any optimal solution b^* is described by a vector $\tilde{y}_w(x)$ in the finite power sets over the regions $\tilde{\mathcal{Y}} \subset \times_r 2^{\mathcal{Y}_r}$:

$$\tilde{y}_w(x) = (\tilde{y}_{w,r}(x))_{r \in \mathcal{R}} \qquad where \qquad \tilde{y}_{w,r}(x) = \{ y_r : b^*_r(y_r) > 0 \}$$

Moreover, if there is a unique optimal solution b^* then it corresponds to an extreme point in the local polytope.

Proof: The program is convex over a compact set, thus strong duality holds. Fixing the Lagrange multipliers $\lambda_{r \to s}(y_r)$ that correspond to the marginal constraints $\sum_{y_s \setminus y_r} b_s(y_s) = b_r(y_r)$, and considering the probability

constraints as the domain of the primal program, we derive the dual program

$$\sum_r \max_{y_r} \Big\{ \theta_r(y_r; x, w) + \sum_{c:c\subset r} \lambda_{c\to r}(y_c) - \sum_{p:p\supset r} \lambda_{r\to p}(y_r) \Big\}$$

Lagrange optimality constraints (or equivalently, Danskin Theorem) determine the primal optimal solutions $b_r^*(y_r)$ to be probability distributions over the set $\arg\max_{y_r}\{\theta_r(y_r; x, w) + \sum_{c:c\subset r} \lambda^*_{c\to r}(y_c) - \sum_{p:p\supset r} \lambda^*_{r\to p}(y_r)\}$ that satisfy the marginalization constraints. Thus $\tilde{y}_{w,r}(x)$ is the information that identifies the primal optimal solutions, i.e., any other primal feasible solution that has the same $\tilde{y}_{w,r}(x)$ is also a primal optimal solution. □

This theorem extends Proposition 3 in Globerson and Jaakkola (2007) to non-binary and non-pairwise graphical models. The theorem describes the discrete structures of approximate MAP predictions. Thus we are able to define posterior distributions that use efficient, although approximate, predictions while taking into account their structures. To integrate these posterior distributions to randomized risk we extend the loss function to $L(\tilde{y}_w(x), y)$. One can verify that the results in Section 1.3 follow through, e.g., by considering loss functions $L : \tilde{\mathcal{Y}} \times \tilde{\mathcal{Y}} \to [0, 1]$ while the training examples labels belong to the subset $\mathcal{Y} \subset \tilde{\mathcal{Y}}$.

1.7 Empirical Evaluation

We presents two sets of experiments. The first set is a phoneme recognizer when the loss is frame error rate (Hamming distance) and phoneme error rate (normalized edit distance). The second set of experiments is an interactive image segmentation.

1.7.1 Phonetic recognition

We evaluated the proposed method on the TIMIT acoustic-phonetic continuous speech corpus (Lamel et al., 1986). The training set contains 462 speakers and 3696 utterances. We used the core test set of 24 speakers and 192 utterances and a development set of 50 speakers and 400 utterances as defined in (Sha and Saul, 2007) to tune the parameters. Following the common practice (Lee and Hon, 1989), we mapped the 61 TIMIT phonemes into 48 phonemes for training, and further collapsed from 48 phonemes to 39 phonemes for evaluation. We extracted 12 MFCC features and log energy with their deltas and double deltas to form 39-dimensional acoustic feature vectors. The window size and the frame size were 25 msec and 10 msec, respectively.

Similar to the output and transition probabilities in HMMs, our implementation has two sets of potentials. The first set of potential captures the confidence of a phoneme based on the acoustic. For each phoneme we define a potential function that is a sum over all acoustic features corresponding to that phoneme. Rather than sum the acoustic features directly, we sum them mapped through an RBF kernel. The kernel is approximated using the Taylor expansion of order 3. Below we report results with a context window of 1 frame and a context window of 9 frames.

The second set of potentials captures both the duration of each phoneme and the transition between phonemes. For each pair of phonemes $p, q \in P$ we define the potential as a sum over all transitions between phoneme p and q.

We applied the algorithm as discussed in Section 1.4 where we set the parameters over a development set. The probit expectation was approximated by a mean over 1000 samples. The initial weight vector was set to averaged weight vector of the Passive-Aggressive (PA) algorithm Crammer et al. (2006), which was trained with the same set of parameters and with 100 epochs as described in Crammer (2010).

Table 1.1 summarizes the results and compare the performance of the proposed algorithm to other algorithms for phoneme recognition. Although the algorithm aims at minimizing the phoneme error rate, we also report the frame error rate, which is the fraction of misclassified frames. A common practice is to split each phoneme segment into three (or more) states. Using such a technique usually improves performance (see for example Mohamed and Hinton (2010); Sung and Jurafsky (2010); Schwartz et al. (2006)). Here we report results on approaches which treat the phoneme as a whole, and defer the issues of splitting into states in our algorithm for future work. In the upper part of the table (above the line), we report results on approaches which make use of context window of 1 frame. The first two rows are two HMM systems taken from Keshet et al. (2006) and Cheng et al. (2009) with a single state corresponding to our setting. KSBSC Keshet et al. (2006) is a kernel-based recognizer trained with the PA algorithm. PA and DROP Crammer (2010) are online algorithms which use the same setup and feature functions described here. Online LM-HMM Cheng et al. (2009) and Batch LM-HMM Sha and Saul (2007) are algorithms for large margin training of continuous density HMMs. Below the line, at the bottom part of the table, we report the results with a context of 9 frames. CRF Morris and Fosler-Lussier (2008) is based on the computation of local posteriors with MLPs, which was trained on a context of 9 frames. We can see that our algorithm outperforms all algorithms except for the large margin HMMs. The difference between our algorithm and the LM-HMM algorithm might

Method	Frame error rate	Phoneme error rate
HMM (Cheng et al., 2009)	39.3%	42.0%
HMM (Keshet et al., 2006)	35.1%	40.9%
KSBSC (Keshet et al., 2006)	-	45.1%
PA (Crammer, 2010)	30.0%	33.4%
DROP (Crammer, 2010)	29.2%	31.1%
PAC-Bayes 1-frame	**27.7%**	**30.2%**
Online LM-HMM (Cheng et al., 2009)	25.0%	30.2%
Batch LM-HMM (Sha and Saul, 2007)	-	28.2%
CRF, 9-frames, MLP (Morris and Fosler-Lussier, 2008)	-	29.3%
PAC-Bayes 9-frames	**26.5%**	**28.6%**

Table 1.1: Reported results on TIMIT core test set.

be in the richer expressive power of the latter. Using a context of 9 frames the results of our algorithm are comparable to LM-HMM.

1.7.2 Image segmentation

We perform experiments on an interactive image segmentation. We use the Grabcut dataset proposed by Blake et al. (2004) which consists of 50 images of objects on cluttered backgrounds and the goal is to obtain the pixel-accurate segmentations of the object given an initial "trimap" (see Figure 1.1). A trimap is an approximate segmentation of the image into regions that are well inside, well outside and the boundary of the object, something a user can easily specify in an interactive application.

A popular approach for segmentation is the GrabCut approach (Boykov et al., 2001; Blake et al., 2004). We learn parameters for the "Gaussian Mixture Markov Random Field" (GMMRF) formulation of Blake et al. (2004) using a potential function over foreground/background segmentations $Y = \{-1, 1\}^n$: $\theta(y; x, w) = \sum_{i \in V} \theta_i(y_i; x, w) + \sum_{i,j \in E} \theta_{i,j}(y_i, y_j; x, w)$. The local potentials are $\theta_i(y_i; x, w) = w_{y_i} \log P(y_i|x)$, where w_{y_i} are parameters to be learned while $P(y_i|x)$ are obtained from a Gaussian mixture model learned on the background and foreground pixels for an image x in the initial trimap. The pairwise potentials are $\theta_{i,j}(y_i, y_j; x, w) = w_a \exp(-(x_i - x_j)^2) y_i y_j$, where x_i denotes the intensity of image x at pixel i, and w_a are the parameters to be learned for the angles $a \in \{0, 90, 45, -45\}^\circ$. These potential functions are supermodular as long as the parameters w_a are nonnegative, thus MAP prediction can be computed efficiently with the graph-cuts algorithm. For these parameters we use multiplicative posterior

model with the Gamma distribution. The dataset does not come with a standard training/test split so we use the odd set of images for training and even set of images for testing. We use stochastic gradient descent with the step parameter decaying as $\eta_t = \frac{\eta}{t_o+t}$ for 250 iterations.

We use two different loss functions for training/testing our approach to illustrate the flexibility of our approach for learning using various task specific loss functions. The "GrabCut loss" measures the fraction of incorrect pixel labels in the region specified as the boundary in the trimap. The "PASCAL loss", which is commonly used in several image segmentation benchmarks, measures the ratio of the intersection and union of the foregrounds of ground truth segmentation and the solution.

As a comparison we also trained parameters using moment matching of MAP perturbations (Papandreou and Yuille, 2011) and structured SVM. We use a stochastic gradient approach with a decaying step size for 1000 iterations. Using structured SVM, solving loss-augmented inference $\max_{\hat{y}\in Y}\{L(y,\hat{y}) + \theta(y;x,w)\}$ with the hamming loss can be efficiently done using graph-cuts. We also consider learning parameters with all-zero loss function, i.e., $L(y,\hat{y}) \equiv 0$. To ensure that the weights remain non-negative we project the weights into the non-negative side after each iteration.

Table 1.2 shows the results of learning using various methods. For the GrabCut loss, our method obtains comparable results to the GMMRF framework of Blake et al. (2004), which used hand-tuned parameters. Our results are significantly better when PASCAL loss is used. Our method also outperforms the parameters learned using structured SVM and Perturb-and-MAP approaches. In our experiments the structured SVM with the hamming loss did not perform well – the loss augmented inference tended to focus on maximum violations instead of good solutions which causes the parameters to change even though the MAP solution has a low loss (a similar phenomenon was observed in Szummer et al. (2008). Using the all-zero loss tends to produce better results in practice as seen in Table 1.2. Figure 1.1 shows some sample images, the input trimap, and the segmentations obtained using our approach.

1.8 Discussion

Learning complex models requires one to consider non-decomposable loss functions that take into account the desirable structures. We suggest the use of the Bayesian perspectives to efficiently sample and learn such models using random MAP predictions. We show that any smooth posterior distribution would suffice to define a smooth PAC-Bayesian risk bound which

Method	Grabcut loss	PASCAL loss
Our method	**7.77%**	**5.29%**
Structured SVM (hamming loss)	9.74%	6.66%
Structured SVM (all-zero loss)	7.87%	5.63%
GMMRF (Blake et al., 2004)	7.88%	5.85%
Perturb-and-MAP (Papandreou and Yuille, 2011)	8.19%	5.76%

Table 1.2: Learning the Grabcut segmentations using two different loss functions. Our learned parameters outperform structured SVM approaches and Perturb-and-MAP moment matching

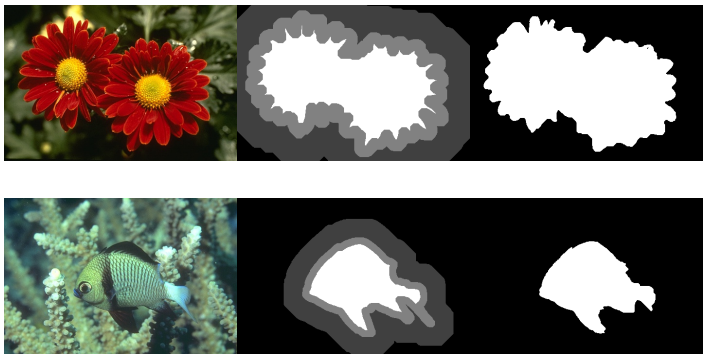

Figure 1.1: Two examples of image (*left*), input "trimap" (*middle*) and the final segmentation (*right*) produced using our learned parameters.

can be minimized using gradient decent. In addition, we relate the posterior distributions to the computational properties of the MAP predictors. We suggest multiplicative posterior models to learn supermodular potential functions that come with specialized MAP predictors such as the graph-cut algorithm. We also describe label-augmented posterior models that can use efficient MAP approximations, such as those arising from linear program relaxations. We did not evaluate the performance of these posterior models, and further exploration of such models is required.

The results here focus on posterior models that would allow for efficient sampling using MAP predictions. There are other cases for which specific posterior distributions might be handy, e.g., learning posterior distributions of Gaussian mixture models. In these cases, the parameters include the covariance matrix, thus would require to sample over the family of positive definite matrices.

1.9 References

A. Blake, C. Rother, M. Brown, P. Perez, and P. Torr. Interactive image segmentation using an adaptive gmmrf model. In *ECCV 2004*, pages 428–441. 2004.

Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *PAMI*, 2001.

O. Catoni. PAC-Bayesian supervised classification: the thermodynamics of statistical learning. *arXiv preprint arXiv:0712.0248*, 2007.

C.-C. Cheng, F. Sha, and L. K. Saul. A fast online algorithm for large margin training of continuous-density hidden Markov models. In *Interspeech*, 2009.

K. Crammer. Efficient online learning with individual learning-rates for phoneme sequence recognition. In *Proc. ICASSP*, 2010.

K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer. Online passive aggressive algorithms. *Journal of Machine Learning Research*, 7, 2006.

C. Do, Q. Le, C.-H. Teo, O. Chapelle, and A. Smola. Tighter bounds for structured estimation. In *Proceedings of NIPS (22)*, 2008.

G. Folland. Real analysis: Modern techniques and their applications, john wiley & sons. *New York*, 1999.

P. Germain, A. Lacasse, F. Laviolette, and M. Marchand. PAC-Bayesian learning of linear classifiers. In *ICML*, pages 353–360. ACM, 2009.

K. Gimpel and N. Smith. Softmax-margin crfs: Training log-linear models with cost functions. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 733–736. Association for Computational Linguistics, 2010.

A. Globerson and T. S. Jaakkola. Fixing max-product: Convergent message passing algorithms for MAP LP-relaxations. *Advances in Neural Information Processing Systems*, 21, 2007.

L. Goldberg and M. Jerrum. The complexity of ferromagnetic ising with local fields.

Combinatorics Probability and Computing, 16(1):43, 2007.

M. Jerrum and A. Sinclair. Polynomial-time approximation algorithms for the ising model. *SIAM Journal on computing*, 22(5):1087–1116, 1993.

J. Keshet, S. Shalev-Shwartz, S. Bengio, Y. Singer, and D. Chazan. Discriminative kernel-based phoneme sequence recognition. In *Interspeech*, 2006.

J. Keshet, D. McAllester, and T. Hazan. PAC-Bayesian approach for minimization of phoneme error rate. In *ICASSP*, 2011.

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

L. Lamel, R. Kassel, and S. Seneff. Speech database development: Design an analysis of the acoustic-phonetic corpus. In *DARPA Speech Recognition Workshop*, 1986.

J. Langford and J. Shawe-Taylor. PAC-Bayes & margins. *Advances in neural information processing systems*, 15:423–430, 2002.

K.-F. Lee and H.-W. Hon. Speaker independent phone recognition using hidden markov models. *IEEE Trans. Acoustic, Speech and Signal Proc.*, 37(2):1641–1648, 1989.

D. McAllester. Simplified PAC-Bayesian margin bounds. *Learning Theory and Kernel Machines*, pages 203–215, 2003.

D. McAllester. Generalization bounds and consistency for structured labeling. In B. Schölkopf, A. J. Smola, B. Taskar, and S. Vishwanathan, editors, *Predicting Structured Data*, pages 247–262. MIT Press, 2006.

D. McAllester and J. Keshet. Generalization bounds and consistency for latent structural probit and ramp loss. In *Proceeding of NIPS*, 2011.

D. McAllester, T. Hazan, and J. Keshet. Direct loss minimization for structured prediction. *Advances in Neural Information Processing Systems*, 23:1594–1602, 2010.

A. Mohamed and G. Hinton. Phone recognition using restricted boltzmann machines. In *Proc. ICASSP*, 2010.

J. Morris and E. Fosler-Lussier. Conditional random fields for integrating local discriminative classifiers. *IEEE Trans. on Acoustics, Speech, and Language Processing*, 16(3):617–628, 2008.

G. Papandreou and A. Yuille. Perturb-and-map random fields: Using discrete optimization to learn and sample from energy models. In *ICCV*, Barcelona, Spain, Nov. 2011. doi: 10.1109/ICCV.2011.

J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers, 1988.

M. Ranjbar, T. Lan, Y. Wang, S. Robinovitch, Z.-N. Li, and G. Mori. Optimizing nondecomposable loss functions in structured prediction. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 35(4):911–924, 2013.

A. Rush and M. Collins. A tutorial on dual decomposition and lagrangian relaxation for inference in natural language processing.

P. Schwartz, P. Matejka, and J. Cernocky. Hierarchical structures of neural networks for phoneme recognition. In *Proc. ICASSP*, 2006.

M. Seeger. Pac-bayesian generalisation error bounds for gaussian process classification. *The Journal of Machine Learning Research*, 3:233–269, 2003.

Y. Seldin. *A PAC-Bayesian Approach to Structure Learning*. PhD thesis, 2009.

Y. Seldin, F. Laviolette, N. Cesa-Bianchi, J. Shawe-Taylor, and P. Auer. Pac-bayesian inequalities for martingales. *Information Theory, IEEE Transactions on*, 58(12):7086–7093, 2012.

F. Sha and L. K. Saul. Comparison of large margin training to other discriminative methods for phonetic recognition by hidden Markov models. In *Proc. ICASSP*, 2007.

D. Sontag, T. Meltzer, A. Globerson, T. Jaakkola, and Y. Weiss. Tightening LP relaxations for MAP using message passing. In *Conf. Uncertainty in Artificial Intelligence (UAI)*, 2008.

Y.-H. Sung and D. Jurafsky. Hidden conditional random fields for phone recognition. In *Proc. ASRU*, 2010.

M. Szummer, P. Kohli, and D. Hoiem. Learning crfs using graph cuts. In *Computer Vision–ECCV 2008*, pages 582–595. Springer, 2008.

D. Tarlow, R. Adams, and R. Zemel. Randomized optimum models for structured prediction. In *AISTATS*, pages 21–23, 2012.

B. Taskar, C. Guestrin, and D. Koller. Max-margin Markov networks. *Advances in neural information processing systems*, 16:51, 2004.

A. Tewari and P. Bartlett. On the consistency of multiclass classification methods. *Journal of Machine Learning Research*, 8:1007–1025, 2007.

I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6(2):1453, 2006.