Learning Efficient Random Maximum A-Posteriori Predictors with Non-Decomposable Loss Functions

Authored by:

Tommi Jaakkola
Joseph Keshet
Tamir Hazan
Subhransu Maji

Abstract

In this work we develop efficient methods for learning random MAP predictors for structured label problems. In particular, we construct posterior distributions over perturbations that can be adjusted via stochastic gradient methods. We show that every smooth posterior distribution would suffice to define a smooth PAC-Bayesian risk bound suitable for gradient methods. In addition, we relate the posterior distributions to computational properties of the MAP predictors. We suggest multiplicative posteriors to learn super-modular potential functions that accompany specialized MAP predictors such as graph-cuts. We also describe label-augmented posterior models that can use efficient MAP approximations, such as those arising from linear program relaxations.

1 Paper Body

Learning and inference in complex models drives much of the research in machine learning applications ranging from computer vision, natural language processing, to computational biology [1, 18, 21]. 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 best describe the labels of observed instances. The goodness of fit is usually measured by a loss function. 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 feasible to only find the most likely or maximum a-posteriori (MAP) assignment, rather than sampling according to their likelihood. Indeed, substantial effort has gone into

developing algorithms for recovering MAP assignments, either based on specific parametrized restrictions such as super-modularity [2] or by devising approximate methods based on linear programming relaxations [21]. Learning MAP predictors is usually done by structured-SVMs that compare a ?loss adjusted? MAP prediction to its training label [25]. In practice, most loss functions used decompose in the same way as the potential function, so as to not increase the complexity of the MAP prediction task. Nevertheless, non-decomposable loss functions capture the structures in the data that we would like to learn. Bayesian approaches for expected loss minimization, or risk, effortlessly deal with nondecomposable loss functions. The inference procedure samples a structure according to its likelihood, and computes its loss given a training label. Recently [17, 23] 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 incorporating perturbmax models to Bayesian loss minimization one would ultimately like to use the PAC-Bayesian risk [11, 19, 3, 20, 5, 10]. Our work explores the Bayesian aspects that emerge from PAC-Bayesian risk minimization. We focus on computational aspects when constructing posterior distributions, so that they could be used 1

to minimize the risk bound efficiently. We show that any smooth posterior distribution would suffice to define a smooth risk bound which can be minimized through gradient decent. In addition, we relate the posterior distributions to the computational properties of MAP predictors. We suggest multiplicative posterior models to learn super-modular potential functions, that come with specialized MAP predictors such as graph-cuts [2]. We also describe label-augmented posterior models that can use MAP approximations, such as those arising from linear program relaxations [21].

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 w that produce to most accurate prediction y ? Y to a given object x. Structures of labels are conveniently described by a discrete product space Y = Y1 ? ? ? ? ? Yn . We describe the potential of relating a label y to an object x with respect to the parameters w by real valued functions ?(y; x, w). Our goal is to learn the parameters w that best describe the training data (x, y) ? S. Within Bayesian perspectives, the distribution that one learns given the training data is composed from a distribution over the parameter space qw (?) and over the labels space P [y—w, x] ? exp ?(y; x, w). Using the Bayes rule we derive the predictive distribution over the structures Z P [y—x] = P [y—?, x]qw (?)d? (1) Unfortunately, sampling algorithms over complex models are provably hard in theory and tend to be slow in many cases of practical interest [7]. This is in contrast to the maximum a-posteriori (MAP) prediction, which can be computed efficiently for many practical cases, even when sampling is

provably hard. (MAP predictor)

yw (x) = arg max ?(y; x, w) y1 ,...,yn

(2)

Recently, [17, 23] suggested to change of the Bayesian posterior probability models to utilize the MAP prediction in a deterministic manner. These perturb-max models allow to sample from the predictive distribution with a single MAP prediction:

def (Perturb-max models) P [y—x] = P??qw y = y? (x) (3) A is decomposed along a graphical model if it has the form ?(y; x, w) = P potential function P i?V ?i (yi ; x, w) + i,j?E ?i,j (yi , yj ; x, w). If the graph has no cycles, MAP prediction can be computed efficiently using the belief propagation algorithm. Nevertheless, 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 Y = {?1, 1}n and its pairwise interactions favor adjacent states to have the same label, i.e., ?i,j (?1, ?1; x, w)+?i,j (1, 1; x, w) ? ?i,j (?1, 1; x, w)+?i,j (1, ?1; x, w). In such cases MAP prediction reduces to computing the min-cut (graph-cuts) algorithm. Recently, a sequence of works attempt to solve the MAP prediction task for non-supermodular potential function as well as general regions. These cases usually involve potentials function that are described by a family R of subsets of variables r ? {1, ..., n}, called regions. We denote by yr the set of labels that correspond to the region r, namely (yi)i?r and consider the following potential P functions ?(y; x, w) = r?R ?r (yr ; x, w). Thus, MAP prediction can be formulated as an integer linear program: X b? ? arg max br (yr)?r (yr ; x, w) (4) br (yr)

s.t.

r,yr

br (yr) ? {0, 1},

X

br (yr) = 1,

yr

X

bs (ys) = br (yr) ?r ? s

ys yr

The correspondence between MAP prediction and integer linear program solutions is (yw (x))i = arg maxyi b?i (yi). 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 br (yr) ? {0, 1} with nonnegative constraints br (yr) ? 0. These 2

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 [21]. Given training data of object-label pairs, the learning objective is to estimate a predictive distribution over the structured-labels. The goodness of fit is measured by a loss function L(? y , y). As we focus on randomized MAP predictors our goal is to learn the parameters

w that minimize the expected perturb-max prediction loss, or randomized risk. We define the randomized risk at a single instancelabel pair as X

R(w, x, y) = P??qw y? = y? (x) L(? y , y). y??Y

Alternatively, the randomized risk takes the form R(w, x, y) = E??qw [L(y? (x), y)]. The randomized risk originates within the PAC-Bayesian generalization bounds. Intuitively, if the training set is an independent sample, one would expect that best predictor on the training set to perform well on unlabeled objects at test time.

3

Minimizing PAC-Bayesian generalization bounds

Our approach is based on the PAC-Bayesian risk analysis of random MAP predictors. In the following we state the PAC-Bayesian generalization bound for structured predictors and describe the gradients of these bounds for any smooth posterior distribution. The PAC-Bayesian generalization bound describes the expected loss, or randomized risk, when considering the true distributions over object-labels in the world R(w) = E(x,y)?? [R(w, x, y)]. It upper P 1 bounds the randomized risk by the empirical randomized risk RS (w) = —S— (x,y)?S R(w, x, y) and a penalty term which decreases proportionally to the training set size. Here we state the PACBayesian theorem, that holds uniformly for all posterior distributions over the predictions. Theorem 1. (Catoni [3], see also [5]). Let L(? y , y) ? [0, 1] be a bounded loss function. Let p(?) be any probability density functionRand let qw (?) be a family of probability density functions parameterized by w. Let KL(qw ——p) = qw (?) log(qw (?)/p(?)). Then, for any ? ? (0, 1] and for any real number ? ¿ 0, with probability at least 1 ? ? over the draw of the training set the following holds simultaneously for all w

KL(qw ——p) + log(1/?) 1 ?RS (w) + R(w) ? 1 ? exp(??) —S— For completeness we present a proof sketch for the theorem in the appendix. This proof follows Seeger?s PAC-Bayesian approach [19], and extended to the structured label case [13]. The proof technique replaces prior randomized risk, with the posterior randomized risk that holds uniformly for every w, while penalizing this change by their KL-divergence. This change-of-measure step is close in spirit to the one that is performed in importance sampling. The proof is then concluded by simple convex bound on the moment generating function of the empirical risk. To find the best posterior distribution that minimizes the randomized risk, one can minimize its empirical upper bound. We show that whenever the posterior distributions have smooth probability density functions qw (?), the perturb-max probability model is smooth as a function of w. Thus the randomized risk bound can be minimized with gradient methods. Theorem 2. Assume qw (?) is smooth as a function of its parameters, then the PAC-Bayesian bound is smooth as a function of w: h i 1 X ?w RS (w) = E??qw ?w [log qw (?)]L(y? (x), y) —S— (x,y)?S

Moreover, the KL-divergence is a smooth function of w and its gradient takes the form: h i ?w KL(qw ——p) = E??qw ?w [log qw (?)] log(qw (?)/p(?)) + 1 R Proof: First we note that R(w, x, y) = qw (?)L(y? (x), y)d?. Since qw (?) is a probability density function and L(? y , y) ? [0, 1] we can differentiate under

the integral (cf. [4] Theorem 2.27). Z ?w R(w, x, y) = ?w qw (?)L(y? (x), y)d? 3

Using the identity ?w qw (?) = qw (?)?w log(qw (?)) the first part of the proof follows. The second part of the proof follows in the same manner, while noting that ?w (qw (?) log qw (?)) = (?w qw (?))(log qw (?) + 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 qw (?) ? exp(?k? ? wk2 /2), thus the gradient of its log-density is the linear moment ?, i.e., ?w [log qw] = ? ? w. Taking any smooth distribution qw (?), we can find the parameters w 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 ?? [log qw (?)]L(y? (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 ?w [log qw (?)](log(qw (?)/p(?)) + 1). This approach, called stochastic approximation or online gradient descent, amounts to use the stochastic gradient update rule

w ? w ? ? ? ??w [log qw (?)] L(y? (x), y) + log(qw (?)/p(?)) + 1 where ? is the learning rate. Next, we explore different posterior distributions from computational perspectives. Specifically, we show how to learn the posterior model so to ensure the computational efficiency of its MAP predictor.

4

Learning posterior distributions efficiently

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 2 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. Let q0 (?) be a smooth probability density function with zero mean and set the posterior distribution using additive shifts qw (?) = q0 (? ? w). Let H(q) = ?E??q [log q(?)] be the entropy function. Then KL(qw ——p) = ?H(q0) ? E??q0 [log p(? + w)] In particular, if p(?) ? exp(?k?k2) is Gaussian then ?w KL(qw ——p) = w Proof: KL(qw ——p) = ?H(qw) ? E??qw [log p(?)]. By a linear change of variable, ?? = ? ? w it follows that H(qw) = H(q0) thus ?w H(qw) = 0. Similarly E??qw [log p(?)] = E??q0 [log p(? + w)]. Finally, if p(?) is Gaussian then E??q0 [log p(? + w)] = ?w2 ? E??q0 [? 2]. This result implies that every additively-

shifted smooth posterior distribution may consider the KLdivergence penalty as the square regularization when using a Gaussian prior p(?) ? exp(?k?k2). This generalizes the standard claim on Gaussian posterior distributions [11], for which q0 (?) 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. 4

4.1

Learning constrained posterior models

MAP predictions can be computed efficiently in important practical cases, e.g., supermodular potential functions satisfying ?i,j (?1, ?1; x, w) + ?i,j (1, 1; x, w) ? ?i,j (?1, 1; x, w) + ?i,j (1, ?1; x, w). Whenever we restrict ourselves to symmetric potential function ?i,j (yi , yj ; x, w) = wi,j yi yj , supermodularity translates to nonnegative constraint on the parameters wi,j ? 0. In order to model posterior distributions that allow efficient sampling we define models over the constrained parameter space. Unfortunately, the additive posterior models qw (?) = q0 (? ? w) are inappropriate for this purpose, as they have a positive probability for negative ? values and would generate nonsupermodular 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 to incorporate the parameters of the posterior distribution in a multiplicative manner into a distribution over the nonnegative real numbers. For any distribution q? (?) we determine a posterior distribution with parameters w as qw (?) = q? (?/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 2. For any probability distribution q? (?), let q?,w (?) = q? (?/w)/w be the parametrized posterior distribution. Then KL(q?,w ——p) = ?H(q?) ? log w ? E??q? [log p(w?)] R? Define the Gamma function ?(?) = 0 ? ??1 exp(??). If p(?) = q? (?) = ? ??1 exp(??)/?(?) have the Gamma distribution with parameter ?, then E??q? [log p(w?)] = (? ? 1) log p w ? ?w. Alternatively, if p(?) are truncated Gaussians then E??q? [log p(w?)] = ? ?2 w2 + log ?/2. Proof: The entropy of multiplicative posterior models naturally implies the log-barrier function: Z

? ? =?/w q? (? ?) log q? (? ?) ? log w d? ? = ?H(q?) ? log w. ?H(q?,w) = Similarly, E??q?,w [log p(?)] = E??q? [log p(w?)]. 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 KLdivergence. Thus the multiplicative posterior effortlessly

enforces the constraints of its parameters. This property suggests that using multiplicative rules are computationally favorable. Interestingly, using a prior model with Gamma distribution adds to the barrier function a linear regularization term kwk1 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 ?. 4.2

Learning posterior models with approximate MAP predictions

MAP prediction can be phrased as an integer linear program, stated in Equation (4). 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 perturb-max models to incorporate approximate MAP solutions. Approximate MAP predictions are are described by the feasible set of their linear program relaxations, that is usually called the local polytope: n o X X L(R) = br (yr) : br (yr) ? 0, br (yr) = 1, ?r ? s bs (ys) = br (yr) yr

ys yr

Linear programs 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 perturb-max model that is defined in Equation (3) can be effortlessly extended to the finite set of the local polytope extreme points [15]. This approach has two flaws. First, linear program solutions might not be extreme points, and decoding such a point usually requires additional 5

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 the randomized risk. Theorem 3. Consider approximate MAP predictions that arise from relaxation of the MAP prediction problem in Equation (4). X arg max br (yr)?r (yr ; x, w) s.t. b ? L(R) br (yr)

r,yr

?

Then any optimal solution b is described by a vector y?w (x) in the finite power sets over the regions, Y? ? ?r 2Yr : y?w (x) = (? yw,r (x))r?R

where

y?w,r (x) = {yr : b?r (yr) ¿ 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 P multipliers ?r?s (yr) that correspond to the marginal constraints ys yr bs (ys) = br (yr), and considering the probability constraints as the domain of the primal program, we derive the dual program n o X X X max ?r (yr ; x, w) + ?c?r (yc) ? ?r?p (yr) r

yr

c:c?r

p:p?r

Lagrange optimality constraints (or equivalently, Danskin Theorem) determine the primal optimal solutions b?r (yP r) to be probability distributions over the set arg maxyr {?r (yr ; x, w) + P ? ? ?w,r (x) c:c?r ?c?r (yc) ? p:p?r ?r?p (yr)} that satisfy the marginalization constraints. Thus y is the information that identifies the primal optimal solutions, i.e., any other primal feasible solution that has the same y?w,r (x) is also a primal optimal solution. This theorem extends Proposition 3 in [6] 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(? yw (x), y). One can verify that the results in Section 3 follow through, e.g., by considering loss functions L : Y? ? Y? ? [0, 1] while the training examples labels belong to the subset Y ? Y? .

5

Empirical evaluation

We perform experiments on an interactive image segmentation. We use the Grabcut dataset proposed by Blake et al. [1] 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). 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 [2, 1]. We learn parameters for the ?Gaussian Mixture Markov Random Field? (GMMRF) formulation of [1] using P a potential funcn tion over foreground/background segmentations Y = {?1, 1} : ?(y; x, w) = l?V ?i (yi ; x, w) + P ? (y , y ; x, w). The local potentials are ? (y ; x, w) = w log P (y —x), where wyi are i,j i j i i y i i i,j?E parameters to be learned while P (yi —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 ?i,j (yi , yj ; x, w) = wa exp(?(xi ? xj)2)yi yj , where xi denotes the intensity of image x at pixel i, and wa are the parameters to be learned for the angles a ? {0, 90, 45, ?45}? . These potential functions are supermodular as long as the parameters wa 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 ?t = to?+t for 250 iterations. 6

Method Our method Structured SVM (hamming loss) Structured SVM (all-zero loss) GMMRF (Blake et al. [1]) Perturb-and-MAP ([17])

Grabcut loss 7.77% 9.74% 7.87% 7.88% 8.19%

PASCAL loss 5.29% 6.66% 5.63% 5.85% 5.76%

Table 1: Learning the Grabcut segmentations using two different loss func-

tions. Our learned parameters outperform structured SVM approaches and Perturb-and-MAP moment matching

Figure 1: Two examples of image (left), input ?trimap? (middle) and the final segmentation (right) produced using our learned parameters.

We use two different loss functions for training/testing for 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 pixels 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 [17] and structured SVM. We use a stochastic gradient approach with a decaying step size for 1000 iterations. Using structured SVM, solving loss-augmented inference maxy??Y {L(y, y?) + ?(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, y?) ? 0. To ensure that the weights remain non-negative we project the weights into the non-negative side after each iteration. Table 1 shows the results of learning using various methods. For the GrabCut loss, our method obtains comparable results to the GMMRF framework of [1], 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 [22]). Using the all-zero loss tends to produce better results in practice as seen in Table 1. Figure 1 shows some examples images, the input trimap, and the segmentations obtained using our approach.

6

Related work

Recent years have introduced many optimization techniques that lend efficient MAP predictors for complex models. These MAP predictors can be integrated to learn complex models using structuredSVM [25]. Structured-SVM has a drawback, as its MAP prediction is adjusted by the loss function, therefore it has an augmented complexity. Recently, there has been an effort to efficiently integrate non-decomposable loss function into structured-SVMs [24]. However this approach does not hold for any loss function. Bayesian approaches to loss minimization treat separately the prediction process and the loss incurred, [12]. However, the Bayesian approach depends on the efficiency of its sampling procedure, but unfortunately, sampling in complex models is harder that the MAP prediction task [7]. The recent works [17, 23, 8, 9, 16] integrate efficient MAP predictors into Bayesian modeling. [23] describes the Bayesian perspectives, while [17, 8] describe their relations to the Gibbs distribution and moment matching. [9] provide unbiased samples form the Gibbs distribution using MAP predictors and [16] present their measure concentration properties.

Other strategies for producing 7

(pseudo) samples efficiently include Herding [26]. However, these approaches do not consider risk minimization. The perturb-max models in Equation (3) play a key role in PAC-Bayesian theory [14, 11, 19, 3, 20, 5, 10]. The PAC-Bayesian approaches focus on generalization bounds to the object-label distribution. However, the posterior models in the PAC-Bayesian approaches were not extensively studied in the past. In most cases the posterior model remained undefined. [10] investigate linear predictors with Gaussian posterior models to have a structured-SVM like bound. This bound holds uniformly for every ? and its derivation is quite involved. In contrast we use Catoni?s PAC-Bayesian bound that is not uniform over ? but does not require the log —S— term [3, 5]. The simplicity of Catoni?s bound (see Appendix) makes it amenable to different extensions. In our work, we extend these results to smooth posterior distributions, while maintaining the quadratic regularization form. We also describe posterior distributions for non-linear models. In different perspective, [3, 5] describe the optimal posterior, but unfortunately there is no efficient sampling procedure for this posterior model. In contrast, our work explores posterior models which allow efficient sampling. We investigate two posterior models: the multiplicative models, for constrained MAP solvers such as graph-cuts, and posterior models for approximate MAP solutions.

7

Discussion

Learning complex models requires one to consider non-decomposable loss functions that take into account the desirable structures. We suggest to use 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 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 graph-cuts 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 explorations 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.

A

Proof sketch for Theorem 1

Theorem 2.1 in [5]: For any distribution D over object-labels pairs, for any w-parametrized distribution qw , for any prior distribution p, for any ? ? (0, 1], and for any convex function D : [0, 1] ? [0, 1] ? R, with probability at least 1 ? ? over the draw of the training set the divergence D(E??qw RS (?), E??qw

R(?)) is upper bounded simultaneously for all w by 1 i 1 h KL(qw ——p) + log E??p ES?Dm exp mD(RS (?), R(?)) —S— ? For D(RS (?), R(?)) = F(R(?)) ? ?RS (?), the bound reduces to a simple convex bound onthe moment generating function of the empirical risk: ES?Dm exp mD(RS (?, x, y), R(?, x, y)) = exp(mF(R(?)))ES?Dm exp(?m?RS (?)) Since the exponent function is a convex function of RS (?) = RS (?) ? 1 + (1 ? RS (?)) ? 0, the moment generating function bound is exp(??RS (?)) ? RS (?) exp(??) + (1 ? RS (?)). Since ES RS (?) = R(?), the right term in the risk bound in can be made 1 when choosing F(R(?)) to be the inverse of the moment generating function bound. This is Catoni?s bound [3, 5] for the structured labels case. To derive Theorem 1 we apply exp(?x) ? 1 ? x to derive the lower bound (1 ? exp(??))E??qw R(?) ? ?E??qw RS (?) ? D(E??qw RS (?), E??qw R(?)). 8

2 References

[1] Andrew Blake, Carsten Rother, Matthew Brown, Patrick Perez, and Philip Torr. Interactive image segmentation using an adaptive gmmrf model. In ECCV 2004, pages 428?441. 2004. [2] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. PAMI, 2001. [3] O. Catoni. Pac-bayesian supervised classification: the thermodynamics of statistical learning. arXiv preprint arXiv:0712.0248, 2007. [4] G.B. Folland. Real analysis: Modern techniques and their applications, john wiley & sons. New York, 1999. [5] P. Germain, A. Lacasse, F. Laviolette, and M. Marchand. Pac-bayesian learning of linear classifiers. In ICML, pages 353?360. ACM, 2009. [6] 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. [7] L.A. Goldberg and M. Jerrum. The complexity of ferromagnetic ising with local fields. Combinatorics Probability and Computing, 16(1):43, 2007. [8] T. Hazan and T. Jaakkola. On the partition function and random maximum a-posteriori perturbations. In Proceedings of the 29th International Conference on Machine Learning, 2012. [9] T. Hazan, S. Maji, and T. Jaakkola. On sampling from the gibbs distribution with random maximum a-posteriori perturbations. Advances in Neural Information Processing Systems, 2013. [10] J. Keshet, D. McAllester, and T. Hazan. Pac-bayesian approach for minimization of phoneme error rate. In ICASSP, 2011. [11] John Langford and John Shawe-Taylor. Pac-bayes & margins. Advances in neural information processing systems, 15:423?430, 2002. [12] Erich Leo Lehmann and George Casella. Theory of point estimation, volume 31. 1998. [13] Andreas Maurer. A note on the pac bayesian theorem. arXiv preprint cs/0411099, 2004. [14] D. McAllester. Simplified pac-bayesian margin bounds. Learning Theory and Kernel Machines, pages 203?215, 2003. [15] D. McAllester, T. Hazan, and J. Keshet. Direct loss minimization for structured prediction. Advances in Neural Information Processing Systems, 23:1594?1602, 2010. [16] Francesco Orabona, Tamir Hazan, Anand D Sarwate, and Tommi. Jaakkola. On measure concentration of random maximum a-posteriori perturbations. arXiv:1310.4227, 2013. [17] G. Papandreou

and A. Yuille. Perturb-and-map random fields: Using discrete optimization to learn and sample from energy models. In ICCV, Barcelona, Spain, November 2011. [18] A.M. Rush and M. Collins. A tutorial on dual decomposition and lagrangian relaxation for inference in natural language processing. [19] Matthias Seeger. Pac-bayesian generalisation error bounds for gaussian process classification. The Journal of Machine Learning Research, 3:233?269, 2003. [20] Yevgeny Seldin. A PAC-Bayesian Approach to Structure Learning. PhD thesis, 2009. [21] 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. [22] Martin Szummer, Pushmeet Kohli, and Derek Hoiem. Learning crfs using graph cuts. In Computer Vision?ECCV 2008, pages 582?595. Springer, 2008. [23] D. Tarlow, R.P. Adams, and R.S. Zemel. Randomized optimum models for structured prediction. In AISTATS, pages 21?23, 2012. [24] Daniel Tarlow and Richard S Zemel. Structured output learning with high order loss functions. In International Conference on Artificial Intelligence and Statistics, pages 1212?1220, 2012. [25] B. Taskar, C. Guestrin, and D. Koller. Max-margin Markov networks. Advances in neural information processing systems, 16:51, 2004. [26] Max Welling. Herding dynamical weights to learn. In Proceedings of the 26th Annual International Conference on Machine Learning, pages 1121?1128. ACM, 2009.

9