Abstract: MAP Structured Prediction by Sampling

Shankar Vembu SHANKAR.VEMBU@IAIS.FRAUNHOFER.DE
Thomas Gärtner THOMAS.GAERTNER@IAIS.FRAUNHOFER.DE
Mario Boley MARIO.BOLEY@IAIS.FRAUNHOFER.DE
Fraunhofer IAIS, Schloß Birlinghoven, 53754 Sankt Augustin, Germany

We consider maximum a posteriori (MAP) parameter estimation for structured prediction with exponential family models. In this setting, the main difficulty lies in the computation of the partition function and the first-order moment of the sufficient statistics. We consider the case that efficient algorithms for exact uniform sampling from the output space exist. This assumption is orthogonal to the typical assumptions made in structured output learning. It holds, in particular, for the highly relevant problem of sampling potent drugs. Under our uniform sampling assumption, we show that exactly computing the partition function is intractable (Section 2), but it can be approximated efficiently (Section 3). Furthermore, we show that the first-order moment of the sufficient statistics can be approximated (Section 4) and that we can sample according to the estimated distribution (Section 5). We also present some application settings (Section 6).

1. Preliminaries and Problems

We use $[\![n]\!]$ to denote $\{1,\ldots,n\}$. Let $\mathcal{X}$ and $\mathcal{Y}$ be the input and the output space respectively, where $\mathcal{Y}$ is parameterised by some finite alphabet Σ. For instance, $\mathcal{Y}$ can consist of strings, trees, or graphs over Σ. Let $\{x_i, y_i\}_{i\in[\![m]\!]} \subseteq \mathcal{X}\times\mathcal{Y}$ be a set of observations. Our goal is to find θ as the MAP parameters of the conditional exponential family model:

$$p(y \mid x, \theta) = \exp(\langle\phi(x,y),\theta\rangle)/Z(\theta|x) \ ,$$

where $\phi(x,y)$ are the joint sufficient statistics of x and y, and $Z(\theta|x) = \sum_{y\in\mathcal{Y}} \exp(\langle\phi(x,y),\theta\rangle)$ is the partition function. Imposing a normal prior on θ, this leads to minimising the following function:

$$\frac{\|\theta\|^2}{2\sigma^2} + \frac{1}{m}\sum_{i=1}^{m} \ln Z(\theta \mid x_i) - \left\langle \sum_{i=1}^{m} \phi(x_i,y_i), \theta \right\rangle \ ,$$

where $\sigma^2 > 0$ is the variance of the prior. The first difficulty lies in the following problem:

PARTITION: For a class of output structures $\mathcal{Y}$ over an alphabet Σ, an input structure $x \in \mathcal{X}$, a polynomial time computable map $\phi : \mathcal{X}\times\mathcal{Y} \to \mathbb{R}^d$, and a parameter θ, compute the partition function $Z(\theta|x)$.

To apply efficient iterative optimisation methods we also need the gradient of the log-partition function, i.e., the first order moment of the sufficient statistics $\nabla_\theta \ln Z(\theta|x) = \mathbb{E}_{y\sim p(y|x,\theta)}\phi(x,y)$. We thus also consider the following problem:

MOMENT: For a class of output structures $\mathcal{Y}$ over an alphabet Σ, an input structure $x \in \mathcal{X}$, a polynomial time computable map $\phi : \mathcal{X}\times\mathcal{Y} \to \mathbb{R}^d$, a parameter vector θ, and a vector z, compute $\left\langle \mathbb{E}_{y\sim p(y|x,\theta)}\phi(x,y), z\right\rangle$.

Throughout this paper, we assume that $\|\phi(x,y)\| \le R$, $\|\theta\| \le B$, and $\|z\| \le G$ for constants R, B, and G. We call an algorithm efficient if it has runtime polynomial in $|\Sigma|$ and the size of the observations.

2. Hardness of PARTITION

We first show that no algorithm can efficiently solve PARTITION on the class of problems for which an efficient approach to uniform sampling exists.

Theorem 2.1 *Unless P=NP, there is no efficient algorithm for* PARTITION *on the class of problems for which we can efficiently sample output structures uniformly at random.*

To prove this theorem we suppose such an algorithm existed, consider a particular class of structures, and show that the algorithm could then be used to solve an NP-hard decision problem. We use that (a) cyclic permutations of subsets of the alphabet Σ can be sampled uniformly at random in time polynomial in $|\Sigma|$; and (b) there is no efficient algorithm for PARTITION for the set of cyclic permutations of subsets of the alphabet Σ with $\phi_{uv}(x,y) = 1$ if $\{u,v\} \in y$ and 0 otherwise. Here (a) follows from (Jerrum et al., 1986). To prove (b), we show that by applying such an algorithm to a mul-

tiple of the adjacency matrix of an arbitrary graph and comparing the result with $|\Sigma|^3$ we could then decide if the graph has a Hamiltonian cycle.

3. Approximating PARTITION

In this section, we give an approximation algorithm for PARTITION under the assumption that we can sample efficiently from the distributions $p(y \mid x, \theta)$. This assumption will be reduced to our uniform sampling assumption in Section 5.

Definition 3.1 *Suppose $f : P \to \mathbb{R}^+$ is a function that maps problem instances P to positive real numbers. A* randomised approximation scheme *for P is a randomised algorithm that takes as input an instance $p \in P$ and an error parameter $\epsilon > 0$, and produces as output a number Q such that*

$$\Pr[(1-\epsilon)f(p) \leq Q \leq (1+\epsilon)f(p)] \geq \frac{3}{4} \ .$$

A randomised approximation scheme is said to be fully polynomial *(FPRAS) if it runs in time polynomial in the size of p and $1/\epsilon$.*

Concentration inequalities are often used in machine learning to bound the deviation of an approximation from its real value. For PARTITION, however, this leads to bounds that degrade with $|\mathcal{Y}|$ which typically grows exponentially in our input. We hence employ the approach of (Jerrum & Sinclair, 1996) to express the partition function as a telescoping product of ratios of partition functions and obtain:

Theorem 3.1 *There is an FPRAS for* PARTITION *on the class of output structures for which it is possible to sample efficiently according to the distributions $p(y \mid x, \theta)$.*

With real parameters $0 = \beta_0 < \beta_1 \cdots < \beta_l = 1$, known as the cooling schedule, we express the partition function as the telescoping product

$$\frac{Z(\theta|x)}{Z(\beta_{l-1}\theta|x)} \times \frac{Z(\beta_{l-1}\theta|x)}{Z(\beta_{l-2}\theta|x)} \times \cdots \frac{Z(\beta_1\theta|x)}{Z(\beta_0\theta|x)} \times Z(\beta_0\theta|x) \ .$$

In particular, with an integer parameter $p \geq 3$, we choose the following cooling schedule: $l = p\lceil R||\theta||\rceil$; $\beta_j = j/(pR\|\theta\|)$ for all $j \in [\![l-1]\!]$. Now, define the random variable $f_i(y) = \exp[(\beta_{i-1} - \beta_i)\langle\phi(x,y),\theta\rangle]$, for all $i \in [\![l]\!]$. Observe that $f_i(y)$ with y chosen according to $p(y \mid x, \beta_i\theta)$ is then an unbiased estimator for the ratio $\rho_i = \frac{Z(\beta_{i-1}\theta|x)}{Z(\beta_i\theta|x)}$. This ratio can now be estimated by sampling according to the distribution $p(y \mid x, \beta_i\theta)$ and computing the sample mean of f_i. It can be seen that sufficiently low variance of each estimator is achieved already with a polynomial number of samples. The final estimator $Z(\theta|x)$ is then the product of the reciprocals of the individual ratios.

4. Approximating MOMENT

We now describe how to approximate the gradient-vector multiplications with provable guarantees using concentration inequalities. The gradient-vector multiplication is

$$\langle \nabla_\theta \ln Z(\theta|x), z\rangle = \mathbb{E}_{y\sim p(y|x,\theta)} \langle \phi(x,y), z\rangle \ .$$

We use Hoeffding's inequality to bound the deviation of $\langle \nabla_\theta \ln Z(\theta|x), z\rangle$ from its estimate $\langle d(\theta|x), z\rangle$ on a finite sample of size S, where

$$d(\theta|x) = \frac{1}{S}\sum_{i=1}^{S} \phi(x, y_i) \ ,$$

and the sample is drawn according to $p(y \mid x, \theta)$.

Note that by Cauchy-Schwarz's inequality $|\langle \phi(x,y_i), z\rangle| \leq RG$ for all $i \in [\![S]\!]$. Applying Hoeffding's inequality, we then obtain the following exponential tail bound:

$$\Pr(|\langle \nabla_\theta \ln Z(\theta|x) - d(\theta|x), z\rangle| \geq \epsilon) \leq 2\exp\left(\frac{-\epsilon^2 S}{2R^2G^2}\right) \ .$$

5. Sampling Techniques

The main contribution of this section is a Metropolis process (Metropolis et al., 1953) that can be used to sample structures from $p(y \mid x, \theta)$ given that there exists a uniform sampler for $\mathcal{Y}$. The following Markov chain (META) is hence the last remaining step needed to reduce approximating PARTITION and MOMENT to uniform sampling: In any state y, select the next state z uniformly at random and move to z with probability $\min\left(1, \frac{p(z|x,\theta)}{p(y|x,\theta)}\right)$.

We have two results regarding the mixing time of this chain using coupling (Aldous, 1983) and coupling from the past (Propp & Wilson, 1996), respectively.

Theorem 5.1 *The mixing time of* META *is bounded from above as follows:*

$$\lceil (\ln \epsilon^{-1}) / \ln(1 - \exp(-2BR))^{-1} \rceil \ ;$$

and the Markov chain META *can be used to obtain an exact sample according to the distribution $p(y \mid x, \theta)$ with expected running time bounded from above by* $\exp(2BR)$.

The implication of these results is that we only need to have an exact uniform sampler in order to obtain exact/approximate samples from $p(y \mid x, \theta)$. With an additional factor of $O(\ln(1/\delta))$ time, this is sufficient for an FPRAS.

6. Application Settings

We now describe how to sample exactly and uniformly at random for three combinatorial structures frequently used in machine learning. We then have all the ingredients to approximate PARTITION and MOMENT.

Vertices of a hypercube: The set of vertices of a hypercube is used as the output space in multi-label classification problems (see, for example, Elisseeff and Weston (2001)). An exact uniform sample can be obtained by tossing an unbiased coin for each label.

Permutations: The set of permutations is used as the output space in label ranking problems (see, for example, Dekel et al. (2003)). An exact sample can be obtained uniformly at random by generating a sequence (of length d, the number of labels) of integers where each integer is sampled uniformly from the set $[\![d]\!]$ *without* replacement.

Subtrees of a tree: Let $T = (V, E)$ denote a directed, rooted tree with root r. Let $\mathcal{Y}$ be the class of subtrees of T also rooted at r. Such rooted subtrees from a rooted tree find applications in multi-category hierarchical classification problems as considered by Cesa-Bianchi et al. (2006). To generate samples of subtrees uniformly at random we employ the reduction from uniform sampling to counting (Jerrum et al., 1986). We consider a string representation of the subtrees and need to count the number of suffixes that complete a given prefix into a valid string, i.e., a string that represents a tree. This can be accomplished using dynamic programming techniques similar to the one used in (Collins & Duffy, 2001)

7. Conclusions

We considered structured prediction problems for classes of output structures that can be sampled uniformly at random. This assumption is orthogonal to the typical assumptions made in other approaches (Collins, 2002; Taskar et al., 2005; Tsochantaridis et al., 2005). The assumptions made in these approaches rely on the problem of deciding if a given structure is optimal for a given input and hypothesis. If this problem is not in NP, they cannot be applied efficiently. For many combinatorial structures of interest, this problem is, however, coNP-complete and thus not in NP unless coNP=NP. A simple example for a class of hard output structures are cycles.

Assuming that we can uniformly sample output structures, we considered MAP parameter estimation for conditional exponential family models. We showed that while exactly computing the partition function is infeasible, the partition function as well as the first-order moment of the sufficient statistics can be approximated efficiently. Our results are applicable to many classes of combinatorial output structures including the highly relevant problem of sampling potent drugs via (Goldberg & Jerrum, 1997).

References

Aldous, D. (1983). Random walks on finite groups and rapidly mixing markov chains. *Séminaire de probabilités de Strasbourg*, *17*, 243–297.

Cesa-Bianchi, N., Gentile, C., & Zaniboni, L. (2006). Incremental algorithms for hierarchical classification. *JMLR*, *7*, 31–54.

Collins, M. (2002). Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. *Proc. of EMNLP*.

Collins, M., & Duffy, N. (2001). Convolution kernels for natural language. *NIPS 14*.

Dekel, O., Manning, C. D., & Singer, Y. (2003). Log-linear models for label ranking. *NIPS 16*.

Elisseeff, A., & Weston, J. (2001). A kernel method for multi-labelled classification. *NIPS 14*.

Goldberg, L. A., & Jerrum, M. (1997). Randomly sampling molecules. *Proc. of SODA*.

Jerrum, M., & Sinclair, A. (1996). The Markov chain Monte Carlo method: An approach to approximate counting and integration. In *Hochbaum DS(ed) Approximation Algorithms for NP–hard Problems*, 482–520. PWS Publishing, Boston, Mass.

Jerrum, M. R., Valiant, L. G., & Vazirani, V. V. (1986). Random generation of combinatorial structures from a uniform distribution. *TCS*, *32*, 169–188.

Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., & Teller., E. (1953). Equation of state calculation by fast computing machines. *J. of Chem. Phys.*, *21*, 1087–1092.

Propp, J. G., & Wilson, D. B. (1996). Exact sampling with coupled markov chains and applications to statistical mechanics. *Random Struct. Algorithms*, *9*, 223–252.

Taskar, B., Chatalbashev, V., Koller, D., & Guestrin, C. (2005). Learning structured prediction models: A large margin approach. *Proc. of ICML*.

Tsochantaridis, I., Joachims, T., Hofmann, T., & Altun, Y. (2005). Large margin methods for structured and interdependent output variables. *JMLR*, *6*, 1453–1484.