10-708: Probabilistic Graphical Models, Spring 2015

24 : Regularized Bayesian learning of GMs

Lecturer: Eric P. Xing *Scribes: Rose C. Kanjirathinkal, Yiming Gu*

1 Non-Parametric Bayesian Inference

The general Bayesian framework for learning a model is expressed as: $p(M|x) = \frac{p(x|M)\pi(M)}{\int p(x|M)\pi(M)dM}$, where M is a model from the hypothesis space and x is the observed data. This is a coherent framework for dealing with uncertainties and for combining prior knowledge with evidence.

In Parametric Bayesian Inference, the above framework is slightly modified to represent the model as a finite set of parameters θ: $p(\theta|x) = \frac{p(x|\theta)\pi(\theta)}{\int p(x|\theta)\pi(\theta)d\theta}$. This formulation easily accommodates cases where the model θ is infinite dimensional; this leads to the Non Parametric Bayesian Formulation. Examples include Dirichlet Process Prior (Figure 1), Indian Buffet Process Prior (Figure 2), or Gaussian Process Prior (Figure 3), all with Multinomial/Gaussian/Softmax likelihood.

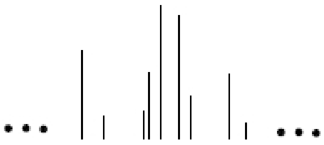

Figure 1: Dirichlet Process Prior

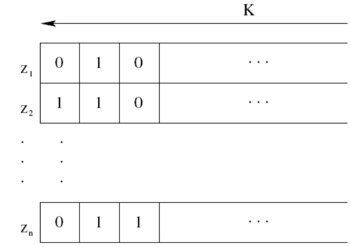

Figure 2: Indian Buffet Process Prior

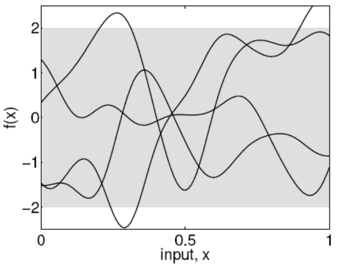

Figure 3: Gaussian Process Prior

Not fixing the number of parameters beforehand is advantageous because it lets the learning process decide the number of parameters in accordance with the data. This also alleviates the model selection problem since the complexity of the model is determined by the data. Since there is no artificial bound on the number of parameters, it allows the model complexity to grow as more data is observed.

2 Controlling Posterior Distribution

From the formulation in the above section, it is clear that the Prior distribution controls the Posterior distribution. However, it is desirable to be able to further regularize the posterior distribution, not just indirectly through the prior. By adding constraints directly to the posterior gives an extra freedom to perform Bayesian inference and can be easier and/or natural in some cases. These constraints could be hard constraints, i.e. they lead to a single feasible subspace. Or, soft constraints, which produces many feasible subspaces with different complexities or penalties.

To add constraints, first reformulate the Bayesian inference formulation as:

$$\begin{aligned}&\min_{p(M)} KL(p(M)\|\pi(M)) \ - \mathbb{E}_{p(M)}[\log p(x|M)]\\ &s.t.\, p(M) \in \mathcal{P}_{prob}\end{aligned}$$

which is equivalent to the simpler formulation:

$$\begin{aligned}&\min_{q\geq 0} KL(q\|P(M|x))\\ &s.t. \ \int q = 1\end{aligned}$$

Note that such a minimum q always exists and is equal to $P(M|x)$.

Given such a formulation, it is easy to see how to add more constraints, as described in *[K. Ganchev, J. Graca, J. Gillenwater, and B. Taskar, Posterior Regularization for Structured Latent Variable Models, JMLR 2010]*.

$$\begin{aligned}&\inf_{q(M),\xi} KL(q(M)\|\pi(M)) - \int_M \log p(x|M)q(M)dM + U(\xi)\\ &s.t: q(M) \in \mathcal{P}_{post}(\xi)\end{aligned}$$

where, example constraints could be of the form:

$$\begin{aligned}\mathcal{P}_{post}(\xi) &= \Big\{q(M) \mid \forall t = 1,...,T, h(Eq(\psi_t;x)) \leq \xi_t\Big\} \text{ and,}\\ U(\xi) &= \sum_{t=1}^{T} \mathbb{I}(\xi_t = \gamma_t) = \mathbb{I}(\xi = \gamma)\end{aligned}$$

where, ξ's are the slack variables and $h(\cdot)$ can be thought of a convex constraint

3 Case Studies

3.1 Maximum Entropy Discrimination Markov Network (MED-MN)

Inspecting the evolution of the Max-Margin Learning Paradigms from SVM to Max-Margin Markov Networks (M3N) to Max Entropy Discrimination (MED) to Maximum Entropy Discrimination Markov Network (MED-MN), the above formulation with constraints can be seen. In the specific case of MED-MN, the formulation takes the following form:

$$\begin{aligned}&\min_{p(w),\xi} \ KL(p(w)\|p_0(w)) + U(\xi)\\ &s.t. \ p(w) \in \mathcal{F}_1, \xi_i \geq 0, \forall i\end{aligned}$$

where,

$$\mathcal{F}_1 = \Big\{p(w) : \int p(w)[\Delta F_i(y;w) - \Delta l_i(y)]dw \geq -\xi_i, \forall i, \forall y \neq y^i\Big\}$$

and the prediction uses the mean of the posterior

$$h_1(x;p(w)) = \arg\max_{y\in\mathcal{Y}(x)} \int p(w)F(x,y;w)dw$$

3.2 Partially Observed MaxEntNet (PoMEN)

In this case, partially labeled data is available. Therefore, PoMEN learning uses $p(w,z)$ instead of just $p(w)$. The formulation is as below:

$$\begin{aligned}\min_{p(w,\{z\}),\xi} \; & KL(p(w,\{z\})\|p_0(w,\{z\})) + U(\xi)\\ s.t.\; & p(w,\{z\}) \in \mathcal{F}_2, \xi_i \geq 0, \forall i\end{aligned}$$

where,

$$\mathcal{F}_2 = \Big\{p(w,\{z\}) : \sum_z \int p(w,z)[\Delta F_i(y,z;w) - \Delta l_i(y)]dw \geq -\xi_i, \forall i, \forall y \neq y^i\Big\}$$

And the prediction is made as

$$h_2(x) = \arg\max_{y\in\mathcal{Y}(x)} \sum_z \int p(w,z)F(x,y,z;w)dw$$

4 Learning using Alternating Minimization Algorithm

The basic idea of Alternating Minimization Algorithm for solving a problem of the form $\min_{x,y} f(x,y)$, is to fix at time t, the value of x at that time, given by x_t and solve for $y_t = \min_y f(x_t, y)$. Next, fix the value of y at y_t and solve for x. This is easier to solve because when one variable is fixed, the rest of the problem is convex.

In the case of PoMEN discussed above, the Alternating Minimization algorithm takes the following form with the factorization assumption, $p_0(w,\{z\}) = p_0(w)\prod_{i=1}^N p_0(z_i)$ and $p(w,\{z\}) = p(w)\prod_{i=1}^N p(z_i)$.

Step 1: Keep $p(z)$ fixed, and optimize over $p(w)$ as:

$$\begin{aligned}\min_{p(w),\xi} \; & KL(p(w)\|p_0(w)) + C\sum_i \xi_i\\ s.t.\; & p(w) \in \mathcal{F}_1', \xi_i \geq 0, \forall i\\ \mathcal{F}_1' = & \Big\{p(w) : \int p(w)\mathbb{E}_{p(z)}[\Delta F_i(y,z;w) - \Delta l_i(y)]dw \geq -\xi_i, \forall i, \forall y\Big\}\end{aligned}$$

Step 2: Keep $p(w)$ fixed, and optimize over $p(z)$ as:

$$\begin{aligned}\min_{p(w),\xi} \; & KL(p(z)\|p_0(z)) + C\xi_i\\ s.t.\; & p(z) \in \mathcal{F}_1^*, \xi_i \geq 0\\ \mathcal{F}_1^* = & \Big\{p(z) : \sum_z p(z) \int p(w)[\Delta F_i(y,z;w) - \Delta l_i(y)]dw \geq -\xi_i, \forall i, \forall y\Big\}\end{aligned}$$

5 Predictive Latent Subspace Learning

Follows the Bayes' formulation as the optimization problem,

$$\min_{p(M)} \; KL(p(M)\|\pi(M)) + E_{p(M)}[\log p(x|M)]$$
$$s.t. \;\; p(M) \in p_{prob}$$

In the case of Predictive Latent Subspace Learning, the M denotes any subspace model and p denotes a parametric Bayesian prior.

For the objective of mapping a high-dimensional representation, such as documents, voices, and pictures, into a latent low-dimensional representation, where each dimension can have some interpretable meaning, e.g., a semantic topic or a scene, there have been various of unsupervised learning methods, such as topic models (Blei 2003), total scene latent space models (Li 2009), and multi-view latent Markov models (Xing 2005).

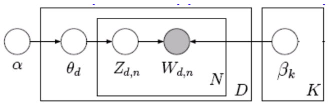

Figure 4: Topic Model

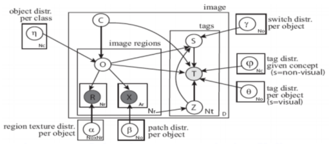

Figure 5: Total Scene Latent Space Model

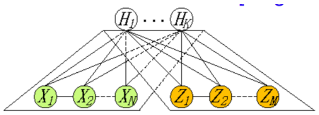

Figure 6: Multi-view Latent Markov Model

However, the problem of unsupervised learning methods mentioned above is that the unsupervised latent subspace representations are generic but can be suboptimal for predictions. For the objective of discovering latent subspace representations that are both **predictive** and **interpretable** by exploring weak supervision information, several supervised learning models are therefore developed. Moreover, from the datasets' point of view, many datasets are available with supervised side information, such as the review documents from Tripadvisor, or the label data from Flickr. These extra label information in the real-world applications further justified the usage of the class of supervised learning models.

Latent Dirichlet Allocation (Figure 7) is a well-know model as a two-stage generative model, for each document d:

1. Sample a topic proportion $\theta_d \sim Dir(\alpha)$
2. For each word:
   Sample a topic $Z_{d,n} \sim Multi(\theta_d)$
   Sample a word $W_{d,n} \sim Multi(\beta_{Z_{d,n}})$

MedLDA (Figure 8) is a max-margin discriminative variant of supervised topic models for both regression and classification. In contrast to the above two-stage procedure of using topic models for prediction tasks, the maximum entropy discrimination latent Dirichlet allocation (MedLDA) is an integration of max-margin learning and hierarchical Bayesian topic models by optimizing a single objective function with a set of expected margin constraints. MedLDA is a special instance of PoMEN (i.e., partially observed maximum entropy discrimination Markov network) (Zhu 2008), which was proposed to combine maxmargin learning and structured hidden variables in Markov networks, for discovering latent topic presentations of documents. In MedLDA, the parameters for the regression or classification model are learned in a max-margin sense;

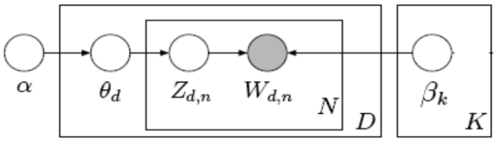

Figure 7: LDA

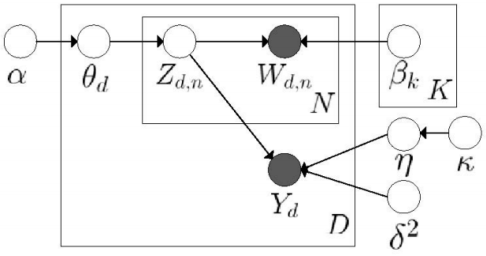

Figure 8: MedLDA

and the discovery of latent topics is coupled with the max-margin estimation of the model parameters. This interplay yields latent topic representations that are more suitable for supervised prediction tasks.

Specifically, MedLDA can be formalized as follows:

1. MedLDA Regression:

$$\min_{q,\alpha,\beta,\delta^2,\xi,\xi^*} \mathcal{L}(q) + C\sum_{d=1}^{D}(\xi_d + \xi_d^*)$$

$$s.t.\,\forall d : \begin{cases} y_d - E[\eta^T \bar{Z}_d] \le \epsilon - \xi_d, & \mu_d \\ -y_d - E[\eta^T \bar{Z}_d] \le \epsilon + \xi_d^*, & \mu_d^* \\ \xi_d \ge 0, & v_d \\ \xi_d^* \ge 0, & v_d^* \end{cases}$$

   where $\mathcal{L}(q)$ denotes the part of loss function that aims to fit the model, and $C\sum_{d=1}^{D}(\xi_d + \xi_d^*)$ denotes the part of loss function that represent the predictive accuracy.

2. MedLDA Classification:

$$\min_{q,\alpha,\beta,\delta^2,\xi} \mathcal{L}(q) + C\sum_{d=1}^{D}(\xi_d)$$

$$s.t.\,\forall d, y \neq y_d : \;\; E[\eta^T \Delta f_d(y)] \ge 1 - \xi_d;\; \xi_d \ge 0$$

MedLDA integrates the max-margin principle into the latent topic discovery process via optimizing one single objective function with a set of expected margin constraints. This integration yields a predictive topic representation that is more suitable for regression or classification. The empirical results on movie review and 20 Newsgroups data sets show the promise of MedLDA on text modeling and prediction accuracy. MedLDA represents the first step towards integrating the max-margin principle into supervised topic models.

Additional case studies can be seen from Upstream Scene Understanding Models (Figure 5) and Supervised Multi-view RBMs (Figure 6), where similar paradigm has been applied to learn the posterior latent variables. The results on MIT Indoor Scene, TRECVID 2003, and Flickr 13 Animal datasets proved the superiority of the class of Predictive Latent Subspace Learning over traditional methods like SVM.

6 Infinite Latent Support Vector Machines

To extend the Bayes' formulation as the optimization problem,

$$\begin{aligned}\min_{p(M)} \; & KL(p(M)\|\pi(M)) - E_{p(M)}[\log p(x|M)] \\ s.t. \; & p(M) \in p_{prob}\end{aligned}$$

(Zhu 2009) modifies the formulation to include slack variables and pose posterior constrains:

$$\begin{aligned}\min_{p(M)} \; & KL(p(M)\|\pi(M)) - E_{p(M)}[\log p(x|M)] + U(\xi) \\ s.t. \; & p(M) \in p_{prob}(\xi)\end{aligned}$$

In the case of infinite SVM:

- Model: latent class model
- Prior: Dirichlet process
- Likelihood: Gaussian likelihood
- Posterior constraints: Max-margin constraints

The infinite SVM has been a milestone of the first attempt to integrate Bayesian nonparametrics, large-margin learning, and kernel methods into one compact algorithm. The process of infinite SVM is as follows:

1. Use DP to decide which classifier to use
2. Given a component classifier:

$$F(y, x; z, \eta) = \eta_z^T f(y, x) = \sum_{i=1}^{\infty} \delta_{z,i} \eta^T f(y, x)$$

3. Overall discriminant function:

$$F(y, x) = E_q[F(y, x; z, \eta)] = \sum_{i=1}^{\infty} q(z = i) E_q[\eta_i]^T f(y, x)$$

4. prediction rule:

$$y^* = arg \max_y F(y, x)$$

5. Learning problem:

$$\min_{p(M)} \; KL(q(z, \eta)\|p_0(z, \eta)) + C_1 R(q(z, \eta))$$

Assumption and relaxation of the variational distribution is

$$q(z, \eta, \gamma, v) = \prod^{D} q(z_d) \prod^{T} q(\eta_t) \prod^{T} q(\gamma_t) \prod^{T-1} q(v_t)$$

To solve the trivial optimization problem, a gradient descent methods could be applied to solve optimal $q(\eta)$ by solving an SVM learning, $q(z)$, $q(\gamma)$, $q(v)$ by a close form update rules.

Comparing to infinite SVM, infinite latent SVM takes the form of:

$$\begin{aligned} \min_{p(M)} \; & KL(p(M) \| \pi(M)) - E_{p(M)}[\log p(x|M)] + U(\xi) \\ s.t. \; & p(M) \in p_{prob}(\xi) \end{aligned}$$

In the case of infinite SVM:

- Model: latent feature model
- Prior: Indian Buffet process
- Likelihood: Gaussian likelihood
- Posterior constraints: Max-margin constraints

The experiments of both infinite SVM and infinite latent SVM showed increased performance on both TRECVID2003 and Flickr image datasets.