10-708: Probabilistic Graphical Models 10-708, Spring 2014

29 : Posterior Regularization

Lecturer: Eric P. Xing *Scribes: Felix Juefei Xu, Abhishek Chugh*

1 Introduction

This is the last lecture which tends to tie together everything we learn so far. What we learned this semester doesn't have to be practiced and applied in an isolated fashion. There is a possibility of grand integration with a method that benefits from many aspects. The main title here, posterior regularization, is actually smaller than what is going to be covered in this lecture. This lecture is not just about regularization, but about the integrative paradigm for learning graphical models. The leaning of graphical models is primarily built on the maximum likelihood principle because that is the most common loss function we define on the graphs. However, as we have seen from previous lectures, there are multiple ways to define alternative loss functions. For example, we can put prior distribution to Bayesian estimation over the model and parameters, so that in the end we can choose to optimize the posterior probability of the model given the data. Very recently, we also have learned some kernel methods, which are new ways of designing loss function structures over the graphical models. In the last lecture, we also learned to bring the max-margin principle as an alternative to drive the derivation of an optimal graphical model in terms of coefficients on certain features or potential functions. All these methods have advantages as well as disadvantages. The greedy question to ask is, can we land on the middle part, which is the integration of everything, with the hope to harness the advantages of all of these methods as shown in Figure 1. We hope the disadvantages are mutually exclusive while the advantages can coexist. One of the recent attempt to do so is the regularized Bayesian inference.

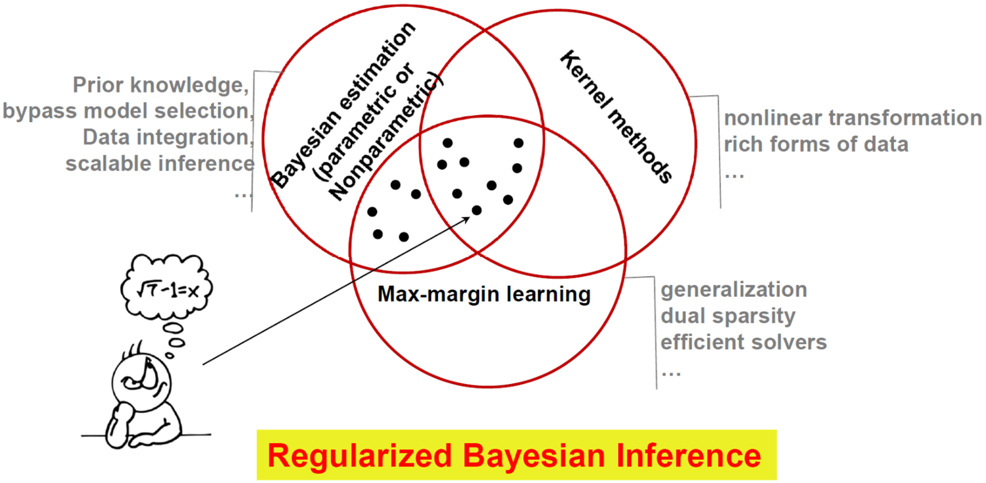

Figure 1: Intersection of the advantages from various methods.

2 Bayesian Inference

The Bayesian inference depends on a prior distribution over the model, and a likelihood function of the data given the model. This, in many cases, is a designed distribution, which means that depending on prior distribution over the model, prior knowledge can be encoded in the design. The goal is to find the tradeoff between the evidence and the prior knowledge. This is a very common model and we can see Bayesian versions of almost everything, e.g. Baeysian logistic regression, LDA is a Bayesian version of older dictionary learning (latent semantic indexing), and non-parametric Bayesian models which gives more flexibility.

Bayesian inference is a coherent framework of dealing with uncertainties:

$$p(\mathcal{M}|\mathbf{x}) = \frac{p(\mathbf{x}|\mathcal{M})\pi(\mathcal{M})}{\int p(\mathbf{x}|\mathcal{M})\pi(\mathcal{M})d\mathcal{M}} \tag{1}$$

where $\mathcal{M}$ is a model from some hypothesis space and $\mathbf{x}$ is observed data. Bayes' rule offers a mathematically rigorous computational mechanism for combining prior knowledge with incoming evidence.

Parametric Bayesian Inference: If the model $\mathcal{M}$ is represented as a finite set of parameters, then you can write down the prior $\pi(\theta)$ on just the parameters θ. The posterior distribution is:

$$p(\theta|\mathbf{x}) = \frac{p(\mathbf{x}|\theta)\pi(\theta)}{\int p(\mathbf{x}|\theta)\pi(\theta)d\theta} \propto p(\mathbf{x}|\theta)\pi(\theta) \tag{2}$$

Here are some examples of conjugacy in picking the appropriate prior.

- Gaussian distribution prior + 2D Gaussian likelihood → Gaussian posterior distribution
- Dirichlet distribution prior + 2D Multinomial likelihood → Dirichlet posterior distribution
- Sparsity-inducing priors + some likelihood models → Sparse Bayesian inference

Nonparametric Bayesian Inference: If the model $\mathcal{M}$ is rich with an infinite set of parameters, we can bring priors that are non-parametric. With that, we can obtain posterior distribution of the model as follows:

$$p(\mathcal{M}|\mathbf{x}) = \frac{p(\mathbf{x}|\mathcal{M})\pi(\mathcal{M})}{\int p(\mathbf{x}|\mathcal{M})\pi(\mathcal{M})d\mathcal{M}} \propto p(\mathbf{x}|\mathcal{M})\pi(\mathcal{M}) \tag{3}$$

Here, the θ is replaced by $\mathcal{M}$ because not only the parameters θ are to be estimated, but also the model $\mathcal{M}$ itself can also be a random variable and needs to be estimated.

Of course, this formula is only symbolically true. In reality, you cannot write a closed-form for something that is infinite. That is why, we resort to process definitions such as Dirichlet process, Indian buffet process, and Gaussian process. They allow you to easily construct conditional distributions of one instance given all the other instances in infinite dimensional feature space.

2.1 Can We Further Control the Posterior Distribution?

Suppose we want even richer properties over the posterior distribution, for example, we want certain dimensions of the posterior to take particular values, or we want the prediction to respect some margin constraints, etc.

It is desirable to further regularize the posterior distribution. By doing so, it gives an extra freedom to perform Bayesian inference; it is arguably more direct to control the behavior of models and it can be easier and more natural in some examples.

We want to somehow regularize the posterior distribution. Directly controlling the posterior distribution is hard because it is a "distribution", not a point estimator of a coefficient.

Hard constraint, such as Dirichlet priors, or all the value of multinomial parameters. It is difficult to put a limit on what parameter is within feasible space, and what is not. A good remedy is to look at soft constraints, where layers of feasible space is nested. Depending on where the parameter is, appropriate penalties can be applied. The soft constraints are more desirable because they provide a tradeoff between violation and fitness of the data as shown in Figure 2.

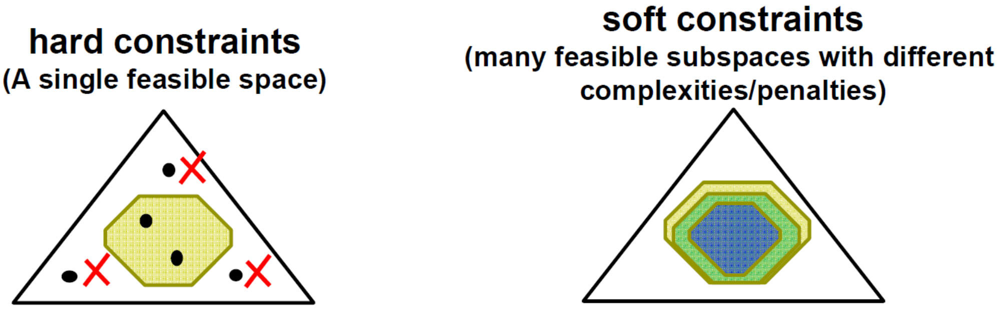
hard constraints
(A single feasible space)
soft constraints
(many feasible subspaces with different complexities/penalties)

Figure 2: Difference between hard constraints and soft constraints.

After some reformulation of Bayesian inference, we can achieve a state where it is easier to further control the posterior distribution as follows. The Bayes' rule is equivalent to:

$$\underset{p(\mathcal{M})}{\text{minimize}}\, \mathrm{KL}\left(p(\mathcal{M}) \| \pi(\mathcal{M})\right) - \mathbb{E}_{p(\mathcal{M})}[\log p(\mathbf{x}|\mathcal{M})] \tag{4}$$

$$\text{subject to} \;\; p(\mathcal{M}) \in \mathcal{P}_{\text{prob}} \tag{5}$$

Throughout the course, we hear a lot about variational inference problems. When we do approximate inference, we usually turn it into a variational problem. It means that the solution lies in the solution to an optimization problem. Instead of dealing with the very difficult p distribution, you are now dealing with a q distribution of x which will then lead to a solution over the function of p and q. This is a typical flavor of variational solution. One example would be the KL-divergence among p and q.

The equivalent reformulation is a minimization problem of the KL-divergence between the posterior $p(\mathcal{M})$ and the prior $\pi(\mathcal{M})$ and minus a function of the likelihood $\mathbb{E}_{p(\mathcal{M})}[\log p(\mathbf{x}|\mathcal{M})]$, and such that the posterior is a valid distribution $p(\mathcal{M}) \in \mathcal{P}_{\text{prob}}$.

we can now cast the Bayesian inference as an optimization problem over augmented KL-divergence. In the original closed-form formulation, everything is pretty much fixed, and there is not many things we can do about it. On the other hand, the new formulation gives the freedom of the entire solution space where we can inject the control points.

E.T. Jaynes (1988): "this fresh interpretation of Bayes' theorem could make the use of Bayesian methods more attractive and widespread, and stimulate new developments in the general theory of inference".

We will show how we can manipulate this variational formulation and further control the posterior distribution as follows.

Expanding the reformulation by rewriting the likelihood term into the integral term and adding additional slack term $U(\xi)$, we have:

$$\inf_{q(\mathbf{M}),\xi} \mathrm{KL}(q(\mathbf{M})\|\pi(\mathbf{M})) - \int_{\mathbf{M}} \log p(\mathcal{D}|\mathbf{M})q(\mathbf{M})d\mathbf{M} + U(\xi) \tag{6}$$

$$\text{subject to } \ q(\mathbf{M}) \in \mathcal{P}_{\text{post}}(\xi) \tag{7}$$

where, for example

$$\mathcal{P}_{\text{post}}(\xi) \overset{\text{def}}{=} \{q(\mathbf{M})|\forall t = 1,\ldots,T, h(Eq(\psi_t;\mathcal{D})) \leq \xi_t\} \tag{8}$$

and

$$U(\xi) = \sum_{t=1}^{T} \mathbb{I}(\xi_t = \gamma_t) = \mathbb{I}(\xi = \gamma) \tag{9}$$

Through the additional term, the posterior $q(\mathbf{M})$ of the model is limited to some space of distributions, where, for example, q should satisfy $0 \leq q(\cdot) \leq 1$ and also the normalizability: $\int q(\cdot) = 1$.

More aggressively, we can set the q distribution to be something else. For example, the q distribution can be used for prediction, or for computing some values. And we can also set a limit to the magnitude to the q distribution to be ξ. It means that not all the q's are valid. In addition, we add the slacking variable, which basically sets the upper bound of the control variables. Hence, this is a constrained optimization problem, and solving it requires convex duality theory. Suppose we know how to do that, it gives us much more flexibility in practicing in regularized Bayesian inference.

What's shown in the above formulation is the base form of Bayesian inference. Once we are to construct a fancier set of the valid posterior distribution, then we are in the domain of the regularized Bayesian inference.

2.2 Maximum Entropy Discrimination Markov Networks

We have seen the maximum entropy discrimination Markov networks, where all posterior distribution is defined on the $\mathbf{w}$, which are the coefficients of some graph potentials as follows. Also the illustration of the KL-divergence applied in maximum entropy discrimination Markov networks is shown in Figure 3.

$$\text{P1}: \qquad \underset{p(\mathbf{w}),\xi}{\text{minimize}}\, \mathrm{KL}(p(\mathbf{w})\|p_0(\mathbf{w})) + U(\xi) \tag{10}$$

$$\text{subject to } \ p(\mathbf{w}) \in \mathcal{F}_1, \xi_i \geq 0, \forall i \tag{11}$$

This is the generalized maximum entropy or regularized KL-divergence. We call this objective Structured MaxEnt discrimination (SMED).

The posterior distribution must satisfy the set constraint $\mathcal{F}_1$, which incorporates the margin constraints as:

$$\mathcal{F}_1 = \left\{p(\mathbf{w}) : \int p(\mathbf{w})[\Delta F_i(\mathbf{y};\mathbf{w}) - \Delta\ell_i(\mathbf{y})]d\mathbf{w} \geq -\xi_i, \forall i, \forall \mathbf{y} \neq \mathbf{y}^i\right\} \tag{12}$$

where the integration is the expected margin constraints. Once we learn the posterior, it can be used for structured prediction. The average from distribution of maximum margin Markov network is as follows:

$$h_1(\mathbf{x};p(\mathbf{w})) = \arg\max_{\mathbf{y}\in\mathcal{Y}(\mathbf{x})} \int p(\mathbf{w})F(\mathbf{x},\mathbf{y};\mathbf{w})d\mathbf{w} \tag{13}$$

In this case, it is used for learning the max-margin Markov network which is a very simple form of a regularized Bayesian inference. The question we need to ask is: can we go beyond? and do something fancier? The rest of this lecture will show a variety of examples, and show how this whole thing can go really wild and flexible.

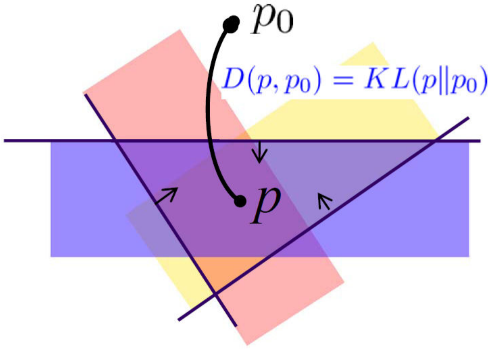

Figure 3: Illustration of Maximum Margin Markov Network.

3 Can We Use This Scheme to Learn Models other than Markov Networks?

Maximum Entropy Discrimination Markov Networks (MEDN) has the following three major advantages.

First of all, MEDN is an averaging model, with PAC-Bayesian prediction error guarantee shown as below:

$$\mathrm{Pr}_Q(M(h, \mathbf{x}, \mathbf{y}) \leq 0) \leq \mathrm{Pr}_{\mathcal{D}}(M(h, \mathbf{x}, \mathbf{y}) \leq \gamma) + O\left(\sqrt{\frac{\gamma^{-2}\mathrm{KL}(p\|p_0)\ln(N|\mathcal{Y}) + \ln N + \ln \delta^{-1}}{N}}\right) \tag{14}$$

Second advantage is the entropy regularization, where useful biases can be introduced. We realize that we can play with the prior of the coefficients. If Gaussian prior is used, we recover the standard max-margin Markov model. If Laplacian prior is used, we can recover certain feasibility set of the posterior, and obtain primal shrinkage effect of the model, which leads to sparse max-margin Markov model. It means, for example, when designing the high-dimensional SVM, not only the decision boundary is dependent on a few support vector, but also the decision hyperplane has intrinsic low dimension because many of the entries are zero due to this ℓ_1 shrinkage. The objective for this example is shown below:

$$\underset{\mu,\xi}{\text{minimize}} \sqrt{\lambda}\sum_{k=1}^{K}\left(\sqrt{\mu_k^2 + \frac{1}{\lambda}} - \frac{1}{\sqrt{\lambda}}\log\frac{\sqrt{\lambda\mu_k^2+1}+1}{2}\right) + C\sum_{i=1}^{N}\xi_i \tag{15}$$

$$\text{subject to } \ \mu^\top \Delta\mathbf{f}_i(\mathbf{y}) \geq \Delta\ell_i(\mathbf{y}) - \xi_i; \xi_i \geq 0, \forall i, \forall \mathbf{y} \neq \mathbf{y}^i. \tag{16}$$

Third advantage is the ability of integrating generative and discriminative principles. We don't need to stick to Markov model, because what we are learning is the distribution of some coefficients over the graphical model. It is possible to include other ingredients in the graphical model such as a hidden variable.

Here is one such real-world example: hierarchical labeling of websites as shown in Figure 4. The gist is that we have a dataset which is intrinsically structured, but the label only exists at the base level. For a website, the label may only tell you some part is an image or text, etc. Usually, when people do machine labeling, it is often beneficial to include latent structures, which helps to partition the parameters into smaller spaces. It also helps better understanding of the lower-level grouping details. Prediction can make use of such bundle information especially during boosting stages. Items that are grouped together tend to have consistent labels. All these information can be encoded in this hierarchy of latent labels.

The difficulties comes in the training stage for such a fancy model because the data is usually not labeled with these meta-labels. The labels only come from the bottom layer.

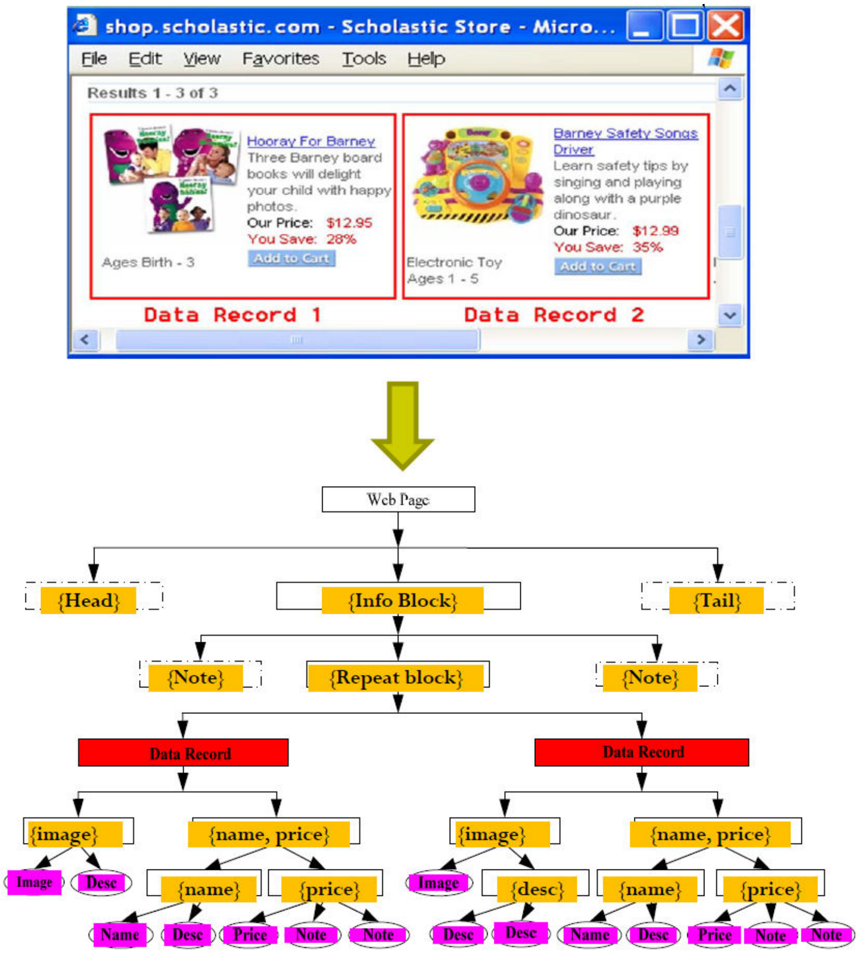

Figure 4: Illustration of Latent Hierarchical Labeling for Webpages.

In old days, people solve this by an EM-type of leaning where we impute and estimate the parameters via a maximum likelihood fashion.

Can we learn a max-margin predictor for this model which contains latent variables? Pushing latent variables into SVM-type framework is very hard. It is not only computationally difficult, but also heuristically it gets into big errors.

Unsupervised clustering using EM-type learning with margin constraints could be dangerous. One intuition is that, after random initialization, two points grouped together may have different labels. So the max-margin classifier will try to push those two data points apart. EM will escalate the error in this case. However, if the classifier is based on probabilistic model, the "pushing" decision will be made with respect to the centroid of the clusters, which would mitigate errors using EM algorithms. That is a problem with a non-probabilistic max-margin solution.

So for partially observed MaxEnDNet (PoMEN), where only the partially labeled data are given: $\mathcal{D} = \{< \mathbf{x}^i, \mathbf{y}^i, \mathbf{z}^i >\}_{i=1}^N$, we are to learn $\mathbf{w}$ and $\mathbf{z}$ jointly, $\mathbf{z}$ is all the hidden variables, $\mathbf{w}$ is the same as before.

The objective function is:

$$\underset{p(\mathbf{w},\{\mathbf{z}\}),\xi}{\text{minimize}}\, \text{KL}(p(\mathbf{w},\{\mathbf{z}\})\|p_0(\mathbf{w},\{\mathbf{z}\})) + U(\xi) \tag{17}$$

$$\text{subject to } \; p(\mathbf{w},\{\mathbf{z}\}) \in \mathcal{F}_2, \xi_i \geq 0, \forall i. \tag{18}$$

where

$$\mathcal{F}_2 = \left\{ p(\mathbf{w},\{\mathbf{z}\}) : \sum_{\mathbf{z}} \int p(\mathbf{w},\mathbf{z})[\Delta F_i(\mathbf{y},\mathbf{z};\mathbf{w}) - \Delta \ell_i(\mathbf{y})]d\mathbf{w} \geq -\xi_i, \forall i, \forall \mathbf{y} \neq \mathbf{y}^i \right\} \tag{19}$$

And the prediction can be made using:

$$h_2(\mathbf{x}) = \underset{y \in \mathcal{Y}(\mathbf{x})}{\arg\max} \sum_{\mathbf{z}} \int p(\mathbf{w},\mathbf{z}) F(\mathbf{x},\mathbf{y},\mathbf{z};\mathbf{w}) d\mathbf{w} \tag{20}$$

This formulation leads to an elegant model, which fully respects the graphical model inference procedure. The inference is still following EM, with multiplicative assumption via an alternating procedure. To be more specific, the factorization assumption is:

$$p_0(\mathbf{w},\{\mathbf{z}\}) = p_0(\mathbf{w}) \prod_{i=1}^{N} p_0(\mathbf{z}_i) \tag{21}$$

$$p(\mathbf{w},\{\mathbf{z}\}) = p(\mathbf{w}) \prod_{i=1}^{N} p(\mathbf{z}_i) \tag{22}$$

The alternating minimization is as follows: in step 1, we keep $p(\mathbf{z})$ fixed, and optimize over $p(\mathbf{w})$:

$$\underset{p(\mathbf{w}),\xi}{\text{minimize}}\, \text{KL}(p(\mathbf{w})\|p_0(\mathbf{w})) + C \sum_i \xi_i \tag{23}$$

$$\text{subject to } \; p(\mathbf{w}) \in \mathcal{F'}_1, \xi_i \geq 0, \forall i \tag{24}$$

$$\mathcal{F'}_1 = \left\{ p(\mathbf{w}) : \int p(\mathbf{w}) E_{p(\mathbf{z})}[\Delta F_i(\mathbf{y},\mathbf{z};\mathbf{w}) - \Delta \ell_i(\mathbf{y})]d\mathbf{w} \geq -\xi_i, \forall i, \forall \mathbf{y} \right\} \tag{25}$$

In step 2, we keep $p(\mathbf{w})$ fixed, and optimize over $p(\mathbf{z})$:

$$\underset{p(\mathbf{w}),\xi}{\text{minimize}}\, \text{KL}(p(\mathbf{w})\|p_0(\mathbf{w})) + C\xi_i \tag{26}$$

$$\text{subject to } \; p(\mathbf{z}) \in \mathcal{F^*}_1, \xi_i \geq 0, \forall i \tag{27}$$

$$\mathcal{F^*}_1 = \left\{ p(\mathbf{w}) : \sum_{\mathbf{z}} p(\mathbf{z}) \int p(\mathbf{w}) E_{p(\mathbf{z})}[\Delta F_i(\mathbf{y},\mathbf{z};\mathbf{w}) - \Delta \ell_i(\mathbf{y})]d\mathbf{w} \geq -\xi_i, \forall i, \forall \mathbf{y} \right\} \tag{28}$$

The alternating steps will reduce to an LP with a polynomial number of constraints, which is the same as the previously discussed fully observed max-margin Markov model.

4 Predictive Latent Subspace Learning via a Large-Margin Approach

Finding latent subspace representations is an old topic which maps a high-dimensional representation into a latent low-dimensional representation, where each dimension can have some interpretable meaning, for example, a semantic topic. Examples are topic models (or latent Dirichlet allocation (LDA)), total scene latent space models, multi0view latent Markov models, principal component analysis, canonical component analysis, etc.

We are interested in the predictive subspace learning with supervision. Unsupervised latent subspace representations are generic but can be suboptimal for predictions. Many datasets are available with supervised side information, but they can be noisy. Good news is that they are not random noise. For example, labels and rating scores are usually assigned based on some intrinsic property of the data which is helpful to suppress noise and capture the most useful aspects of the data. The goal here is to Discover latent subspace representations that are both predictive and interpretable by exploring weak supervision information.

4.1 Latent Dirichlet Allocation

The generative procedure: for each document d, sample a topic proportion $\theta_d \sim \text{Dir}(\alpha)$, and for each word, sample a topic $Z_{d,n} \sim \text{Mult}(\theta_d)$ and sample a word $W_{d,n} \sim \text{Mult}(\beta_{z_{d,n}})$

The joint distribution for LDA is:

$$p(\theta, \mathbf{z}, \mathbf{W}|\alpha, \beta) = \prod_{d=1}^{D} p(\theta_d|\alpha) \left(\prod_{n=1}^{N} p(z_{dn}|\theta_d) p(w_{dn}|z_{dn}, \beta) \right) \tag{29}$$

But the exact inference is intractable.

The variational inference is:

$$q(\mathbf{z}, \theta) \sim p(\mathbf{z}, \theta|\mathbf{W}, \alpha, \beta) \tag{30}$$

$$\mathcal{L}(q) = -E_q[\log p(\theta, \mathbf{z}, \mathbf{W}|\alpha, \beta)] - \mathcal{H}(q(\mathbf{z}, \theta)) \geq -\log p(\mathbf{W}|\alpha, \beta) \tag{31}$$

We can minimize the variational bound to estimate parameters and infer the posterior distribution.

For maximum entropy discrimination LDA (MedLDA), the regression model is:

$$\text{P1}(\text{MedLDA}^r) : \underset{q,\alpha,\beta,\delta^2,\xi,\xi^*}{\text{minimize}} \; \mathcal{L}(q) + C \sum_{d=1}^{D} (\xi_d + \xi_d^*) \tag{32}$$

$$\text{subject to } \; \forall d : y_d - E[\eta^\top Z_d] \leq \epsilon + \xi_d, \mu_d; -y_d + E[\eta^\top Z_d] \leq \epsilon + \xi_d^*, \mu_d^*; \xi_d \geq 0, v_d; \xi_d^* \geq 0, v_d^* \tag{33}$$

and MedLDA classification model is:

$$\text{P2}(\text{MedLDA}^c) : \underset{q,q(\eta),\alpha,\beta,\xi}{\text{minimize}} \; \mathcal{L}(q) + C \sum_{d=1}^{D} (\xi_d) \tag{34}$$

$$\text{subject to } \; \forall d, y \neq y_d : E[\eta^\top \Delta \mathbf{f}_d(y)] \geq 1 - \xi_d; \xi_d \geq 0. \tag{35}$$

MedLDA makes the classification problem easier because MedLDA causes segregation of data classes. The margin constraint due to classification produces a bias in the projection of the data in the topical space. We also have a bias in learning the basis of the topics which makes the data more discriminative.

4.2 Comparison to others

Classifying documents is a complex process. The usual baseline for this problem is to running LDA on the data to project the topic into latent space. We then run second space lSVM on on these to get document labels.

Another approach to solving this problem is by using sLDA or DiscLDA which perform inference using probabilistic supervised topic models, modelling the predictive labels in the train gof the topics, producing a bias. These methods hence achieve somewhat better performance due to this. But these techniques are based on maximizing the likelihood rather than maximizing the separation margin. MedLDA on the other hand produces topic distribution and the predicted class directly. Since this is based on maximum margin principle it performs the best.

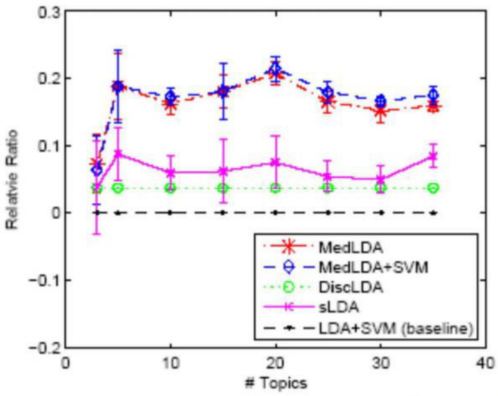

Figure 5: Comparison of different classifiers

If instead we perform SVM classification on the topic distributions produced by MedLDA, we don't get much better classification. This suggests that the original MedLDA is already exhausting the benefit in the data by separating in the max-margin fashion and thus using another SVM is thus not very useful.

MedLDA is also as fast as performing SVM on topic models. Because it iteratively performs the latent variable computation and topic estimation in almost the same manner as topic models without doing any probabilistic inference. sLDA is more expensive because it based on an joint probabilistic model which requires a heavy probabilistic inference over latent variables.

4.3 Posterior Regularization on Other Parametric models

The same technique can be applied to the scene classification problem. We just replace documents with images and image segments of "words". We can apply margin constraints on the labels and we get better performance than other algorithms.

Similarly, in RBM's the feature vectors are separated from the prediction layer by a latent layer. For making prediction, we can train a predictive RBM, we can put labels on the top layer, thus producing a discriminative RBM.

Thus the posterior regularization is generalised to many important parametric models and they universally produce good results.

5 Non Parametric Models

5.1 Mixture of SVMs

One of the main advantages of SVMs comes from its use of Kernels. This enables us to have a non-linear boundary to separate classes. But overfitting can be a problem with complicated (higher order) Kernels.

A technique used to avoid overfitting is to take a mixture of linear SVMs. Each of the linear SVM shave their own linear boundaries. A hidden random variable is used to determine which one of the SVM's boundaries should be used. We can use a low order kernel like RBF instead of linear kernel we can use if the linear classifier fail to capture some of the patterns in data.

5.2 Infinite SVMs

An important question while setting the mixture of SVMs is the number of SVMs to be used. Like in the case of mixture models for density estimation, we can setup a Dirichlet process prior to automatically set the number of SVMs to be used.

Specifically, we work on a Regularized Bayes framework and set the likelihood function to be a latent classification model built on mixture of SVMs. The SVMs are treated as density functions to define likelihood of the data. Since we don't know the number of component SVMs are needed, we set $\pi(\mathcal{M})$ (prior) as the Dirichlet Process. The likelihood is the standard Gaussian likelihood that reduces the crossover of data points across the boundary. We can, in this way, not lose the maximum-margin property of SVMs. The resulting distribution gives a prediction that satisfies margin constraints. This provides for a component classifier, i.e., if we know a component, we can make a linear prediction. We sum over all possible selectors (component SVM-like classifiers). Compared to other SVMs, classifying techniques, we get a more accurate classifier.

5.3 Infinite Latent SVMs

The idea here is to use an Indian Buffet Process prior rather than a DP prior (like Infinite SVM). In this case we not only select a component classifier, we also implicitly do feature selections on the given data. The model itself is a latent feature model, where the feature selection function selects the appropriate features from each data point for feature selection. Here again the classification performance is very good specifically the case where the data lies in a feature space of lower size.

6 Summary

In summarize, we have seen that graphic models can be used in conjunction with other Machine Learning ideas. The maximum marginalisation property can be integrated with topic modeling to produce fast and efficient classifiers.. We have seen that from ideas graphical models can be combined with that SVMs to produce Infinite SVMs and Infinite Latent SVMs by using non-parametric priors such as Dirichlet Process and Indian Buffet Process on regBayes. We can also put prior distribution to Bayesian estimation over the model and parameters, so that in the end we can choose to optimize the posterior probability of the model given the data.