Exponential Family Distributions (contd.), Generative Models for Supervised Learning

CS772A: Probabilistic Machine Learning

Piyush Rai

Plan today

- Exponential Family Distributions
  - Conjugate priors, posterior, and PPD
- Generative approach to supervised learning

Bayesian Inference for Expon. Family Distributions

- Already saw that the total likelihood given N i.i.d. observations $\mathcal{D} = \{x_1, \ldots, x_N\}$

$$p(\mathcal{D}|\theta) \propto \exp\left[\theta^\top \phi(\mathcal{D}) - NA(\theta)\right] \quad \text{where} \quad \phi(\mathcal{D}) = \sum_{i=1}^{N} \phi(\boldsymbol{x}_i)$$

- Let's choose the following prior (note: looks similar in terms of $\boldsymbol{\theta}$ within exp)

$$p(\theta|\nu_0, \boldsymbol{\tau}_0) = h(\theta)\exp\left[\theta^\top \tau_0 - \nu_0 A(\theta) - A_c(\nu_0, \boldsymbol{\tau}_0)\right]$$

- Ignoring the prior's log-partition function $A_c(\nu_0, \boldsymbol{\tau}_0) = \log \int_\theta h(\theta)\exp\left[\theta^\top \tau_0 - \nu_0 A(\theta)\right] d\theta$

$$p(\theta|\nu_0, \boldsymbol{\tau}_0) \propto h(\theta)\exp\left[\theta^\top \tau_0 - \nu_0 A(\theta)\right]$$

- Comparing the prior's form with the likelihood, note that
  - ν_0 is like the number of "pseudo-observations" coming from the prior
  - τ_0 is the total sufficient statistics of the pseudo-observations (τ_0 / ν_0 per pseudo-obs)

The Posterior

- The likelihood and prior were

$$p(\mathcal{D}|\theta) \propto \exp\left[\theta^\top \phi(\mathcal{D}) - NA(\theta)\right] \quad \text{where} \quad \phi(\mathcal{D}) = \sum_{i=1}^{N} \phi(\boldsymbol{x}_i)$$

$$p(\theta|\nu_0, \boldsymbol{\tau}_0) \propto h(\theta) \exp\left[\theta^\top \tau_0 - \nu_0 A(\theta)\right]$$

Assume its log partition function denoted as $A_c(\nu_0, \tau_0)$

- The posterior $p(\theta|\mathcal{D}) \propto p(\theta)p(\mathcal{D}|\theta)$ therefore will be

$$p(\theta|\mathcal{D}) \propto h(\theta) \exp\left[\theta^\top (\tau_0 + \phi(\mathcal{D})) - (\nu_0 + N)A(\theta)\right]$$

Its log partition function will be $A_c(\nu_0 + N, \tau_0 + \phi(\mathcal{D}))$

Posterior is also from the same family as the prior

Happens when the prior is conjugate to the likelihood

- Every exp family likelihood has a conjugate prior having the form above
- Posterior's hyperparams τ_0', ν_0' obtained by adding "stuff" to prior's hyperparams

$$\nu_0' \leftarrow \nu_0 + N$$

$$\tau_0' \leftarrow \tau_0 + \phi(\mathcal{D})$$

Number of pseudo-observations plus number of actual observations

Suff-stats of pseudo-obervations plus suff-stats of actual observations

Another equivalent form

$$p(\theta|\mathcal{D}) \propto h(\theta) \exp\left[\theta^\top (\nu_0 + N) \frac{\nu_0 \bar{\tau}_0 + \phi(\mathcal{D})}{\nu_0 + N} - (\nu_0 + N)A(\theta)\right]$$

$\bar{\tau}_0 = \tau_0 / \nu_0$

$$\nu_0' \leftarrow \nu_0 + N$$

$$\bar{\tau}_0' \leftarrow \frac{\nu_0 \bar{\tau}_0 + N\bar{\phi}}{\nu_0 + N}$$

$\bar{\phi} = \frac{\phi(D)}{N}$

Convex comb of avg suff-stats of pseudo obs and actual obs

Posterior Predictive Distribution

- Assume some training data $\mathcal{D} = \{x_1, \ldots, x_N\}$ from some exp-fam distribution
- Assume some test data $\mathcal{D}' = \{\tilde{x}_1, \ldots, \tilde{x}_{N'}\}$ from the same distribution
- The posterior pred. distr. of $\mathcal{D}'$

Exp. Fam. likelihood w.r.t. test data

Posterior (same form as the prior due to conjugacy)

$$p(\mathcal{D}'|\mathcal{D}) = \int p(\mathcal{D}'|\theta)p(\theta|\mathcal{D})d\theta$$

$$= \int \underbrace{\left[\prod_{i=1}^{N'} h(\tilde{\mathbf{x}}_i)\right]}_{\text{constant w.r.t. } \theta} \exp\left[\theta^\top \phi(\mathcal{D}') - N' A(\theta)\right] h(\theta) \exp\left[\theta^\top(\tau_0 + \phi(\mathcal{D})) - (\nu_0 + N)A(\theta) - \underbrace{A_c(\nu_0 + N, \boldsymbol{\tau}_0 + \phi(\mathcal{D}))}_{\text{constant w.r.t. } \theta}\right] d\theta$$

- This gets further simplified into

$$p(\mathcal{D}'|\mathcal{D}) = \left[\prod_{i=1}^{N'} h(\tilde{\mathbf{x}}_i)\right] \frac{\int h(\theta) \exp\left[\theta^\top(\tau_0 + \phi(\mathcal{D}) + \phi(\mathcal{D}')) - (\nu_0 + N + N')A(\theta)\right] d\theta}{\exp\left[A_c(\nu_0 + N, \boldsymbol{\tau}_0 + \phi(\mathcal{D}))\right]}$$

$$= \left[\prod_{i=1}^{N'} h(\tilde{\mathbf{x}}_i)\right] \frac{Z_c(\nu_0 + N + N', \boldsymbol{\tau}_0 + \phi(\mathcal{D}) + \phi(\mathcal{D}'))}{\exp\left[A_c(\nu_0 + N, \boldsymbol{\tau}_0 + \phi(\mathcal{D}))\right]}$$

Posterior Predictive Distribution

Thus PPD as well as marginal likelihood has closed form expression when working with exp-family distributions

- Since $A_c = \log Z_c$ or $Z_c = \exp(A_c)$, we can write the PPD as

$$p(\mathcal{D}'|\mathcal{D}) = \left[\prod_{i=1}^{N'} h(\tilde{\mathbf{x}}_i)\right] \frac{Z_c(\nu_0 + N + N', \boldsymbol{\tau}_0 + \phi(\mathcal{D}) + \phi(\mathcal{D}'))}{Z_c(\nu_0 + N, \boldsymbol{\tau}_0 + \phi(\mathcal{D}))}$$

$$= \left[\prod_{i=1}^{N'} h(\tilde{\mathbf{x}}_i)\right] \exp\left[A_c(\nu_0 + N + N', \boldsymbol{\tau}_0 + \phi(\mathcal{D}) + \phi(\mathcal{D}')) - A_c(\nu_0 + N, \boldsymbol{\tau}_0 + \phi(\mathcal{D}))\right]$$

- Therefore the posterior predictive is proportional to
  - Ratio of two partition functions of two "posterior distributions" (one with $N + N'$ examples and the other with N examples)
  - Exponential of the difference of the corresponding log-partition functions
- Note that the form of Z_c (and A_c) will simply depend on the chosen conjugate prior
- Very useful result. Also holds for $N = 0$
  - In this case $p(\mathcal{D}') = \int p(\mathcal{D}'|\theta)p(\theta)d\theta$ is simply the marginal likelihood of test data $\mathcal{D}'$

Summary

- Exp. family distributions are very useful for modeling diverse types of data/parameters
- Conjugate priors to exp. family distributions make parameter updates very simple
- Other quantities such as posterior predictive can be computed in closed form
- Useful in designing generative classification models. Choosing class-conditional from exponential family with conjugate priors helps in parameter estimation
- Useful in designing generative models for unsupervised learning
- Used in designing Generalized Linear Models: Model $p(y|x)$ using exp. fam distribution
  - Linear regression (with Gaussian likelihood) and logistic regression are GLMs
- Will see several use cases when we discuss approx inference algorithms (e.g., Gibbs sampling, and especially variational inference)

Generative Supervised Learning

- The conditional distribution $p(y|x)$ can also be defined as

$$p(y|x) = \frac{p(x,y)}{p(x)}$$

Requires modeling the joint distribution of the inputs and outputs

In the discriminative approach for learning $p(y|x)$, we didn't model the inputs x but treated them as "given"

- Generative sup. learning is usually more work because $p(x,y)$ has to be estimated
- However, there are some benefits as well. For example, for classification

$p(y)$ is called the "class-prior" or "class-marginal" distribution

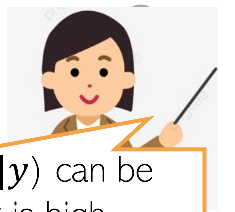
Can incorporate knowledge of frequency ("size") of each class in training data

Can incorporate knowledge of the distribution ("shape") of each class in training data

$$p(y|x) = \frac{p(x,y)}{p(x)} = \frac{p(y)p(x|y)}{p(x)}$$

Can assume simple/sophisticated types of distributions for the "class-conditional" distribution $p(x|y)$ and learned them using the training data of each class

Generative Supervised Learning



Note: Estimating $p(\boldsymbol{x}|y)$ can be difficult especially if $\boldsymbol{x}$ is high-dimensional and we don't have enough data from each class

- The generative classification model

Marginal probability of belonging to class k

Probability (density) of input $\boldsymbol{x}$ under class k

Probability of belonging to class k, conditioned on the input $\boldsymbol{x}$

$$p(y=k|\boldsymbol{x}) = \frac{p(y=k)p(\boldsymbol{x}|y=k)}{\sum_k p(y=k)p(\boldsymbol{x}|y=k)}$$

A way to handle this is to assume simpler forms for $p(\boldsymbol{x}|y)$ (e.g., Gaussian with diagonal/spherical covar – naïve Bayes) but it might sacrifice accuracy too

- We need to learn $p(y)$ and $p(\boldsymbol{x}|y)$ here given training data $(\boldsymbol{X}, \boldsymbol{y}) = \{(x_n, y_n)\}_{n=1}^N$

- Class prior/marginal distribution $p(y)$ will always be a discrete distribution, e.g.,
  - For $y \in \{0,1\}$, $p(y) = p(y|\pi) = \text{Bernoulli}(y|\pi)$ with $\boldsymbol{\pi} \in (0,1)$
  - For $y \in \{1,2,\ldots,K\}$, $p(y) = p(y|\boldsymbol{\pi}) = \text{multinoulli}(y|\boldsymbol{\pi})$ where $\boldsymbol{\pi} = [\pi_1, \ldots, \pi_K]$

$\sum_{k=1}^K \pi_k = 1$

- Class conditional distribution $p(\boldsymbol{x}|y)$ will depend on the nature of inputs, e.g.,
  - For $\boldsymbol{x} \in \mathbb{R}^D$, $p(\boldsymbol{x}|y=k)$ can be a multivariate Gaussian (one per class)

For π, can use Beta or Dirichlet (we have already seen these examples)

$$p(\boldsymbol{x}|y=k) = p(\boldsymbol{x}|\theta_k) = \mathcal{N}(\boldsymbol{x}|\mu_k, \Sigma_k)$$

Note: When estimating θ_k, we only need inputs from class k $\boldsymbol{X}_k = \{\boldsymbol{x}_n : y_n = k\}$

Will need appropriate prior distributions for π and $\{\theta_k\}_{k=1}^K$

- Can estimate π and $\{\theta_k\}_{k=1}^K$ using $(\boldsymbol{X}, \boldsymbol{y})$ via point est. or fully Bayesian infer.

Generative Classification: Making Predictions

- Once $\boldsymbol{\pi}$ and $\{\theta_k\}_{k=1}^K$ are learned, we are ready to make prediction for any test input $\boldsymbol{x}_*$
- Two ways to make the prediction
- Approach 1: If we have point estimates for $\boldsymbol{\pi}$ and $\{\theta_k\}_{k=1}^K$, say $\hat{\boldsymbol{\pi}}$ and $\{\hat{\theta}_k\}_{k=1}^K$. Then

$$p(y_* = k|\boldsymbol{x}_*) = \frac{p(y_* = k|\hat{\pi})p(\boldsymbol{x}_*|\hat{\theta}_k)}{\sum_k p(y = k|\hat{\pi})p(\boldsymbol{x}|\hat{\theta}_k)} \propto \hat{\pi}_k p(\boldsymbol{x}_*|\hat{\theta}_k)$$

Compute for every value of k and normalize

- Approach 2: If we have the full posterior for $\boldsymbol{\pi}$ and $\{\theta_k\}_{k=1}^K$. Then
  - Instead of using $p(y_* = k|\hat{\boldsymbol{\pi}})$, we will use $p(y_* = k|\boldsymbol{y}) = \int p(y_* = k|\pi)p(\pi|\boldsymbol{y})d\pi$
  - Instead of using $p(\boldsymbol{x}_*|\hat{\theta}_k)$, we will use $p(\boldsymbol{x}_*|\boldsymbol{X}_k) = \int p(\boldsymbol{x}_*|\theta_k)p(\theta_k|\boldsymbol{X}_k)d\theta_k$
  - Using these quantities, the prediction will be made as

PPD of y_*

PPD of $\boldsymbol{x}_*$

Compute for every value of k and normalize

$$p(y_* = k|x_*, \boldsymbol{X}, \boldsymbol{y}) = \frac{p(y_* = k|\boldsymbol{y})p(\boldsymbol{x}_*|\boldsymbol{X}_k)}{\sum_k p(y_* = k|\boldsymbol{y})p(\boldsymbol{x}_*|\boldsymbol{X}_k)} \propto p(y_* = k|\boldsymbol{y})p(\boldsymbol{x}_*|\boldsymbol{X}_k)$$

Note that we aren't using a single "best" value of the params π and θ_k unlike Approach 1

Generative Sup. Learning: Some Comments

- A very flexible approach for classification

Incorporate info about how frequent each class is in the training data ("class prior")

Incorporate info about the shape of each class

Consequently, can naturally learn nonlinear boundaries, too (without using kernel methods or deep learning)

$$p(y_* = k|\boldsymbol{x}_*) = \frac{p(y_* = k)p(\boldsymbol{x}_*|y_* = k)}{\sum_k p(y_* = k)p(\boldsymbol{x}_*|y_* = k)}$$

- Can handle missing labels and missing features
  - These can be treated as latent variables as estimated using methods such as EM

Will discuss this later

- Ability to handle missing labels makes it suitable for semi-supervised learning
- The choice of the class-conditional and proper estimation is important
  - Can leverage advances in deep generative models to learn very flexible forms for $\boldsymbol{p(x|y)}$
- Can also use it for regression (define $\boldsymbol{p(x, y)}$ via some distr. and obtain $\boldsymbol{p(y|x)}$ from it)
- Can also combine generative and discriminative approaches for supervised learning

Hybrids of Discriminative and Generative Models

- Both discriminative and generative models have their strengths/shortcomings
- Some aspects about discriminative models for sup. learning
  - Discriminative models have usually fewer parameters (e.g., just a weight vector)
  - Given "plenty" of training data, disc. models can usually outperform generative models

Recall prob linear regression and logistic reg

- Some aspects about generative models for sup. learning
  - Can be more flexible (we have seen the reasons already)
  - Usually have more parameters to be learned
  - Modeling the inputs (learning $\boldsymbol{p(x|y)}$) can be difficult for high-dim inputs
- Some prior work on combining discriminative and generative models. Examples:

$$\alpha \log p(y|x;\theta) + \beta \log p(x;\theta)$$

$$p(x, y, \theta_d, \theta_g) = p_{\theta_d}(y|x) p_{\theta_g}(x) p(\theta_d, \theta_g)$$

$$p(x, y, z) = p(y|x, z) \cdot p(x, z)$$

Approach 1 (McCullum et al, 2006) – modeling the joint $p(x, y|\theta)$ using a multi-conditional likelihood

Approach 2 (Lasserre et al, 2006) – Coupled parameters between discriminative and generative models

Approach 3 (Kuleshov and Ermon, 2017) – Coupling discriminative and generative models via a latent variable z (see "Deep Hybrid Models: Bridging Discriminative and Generative Approaches", UAI 2017)