Exponential Family Distributions and Generative Supervised Learning

CS772A: Probabilistic Machine Learning

Piyush Rai

Announcement

- Quiz 1 on Monday Feb 2, 18:15-19:00 (45 minutes)
- Homework 1 out by end of next week

Plan Today

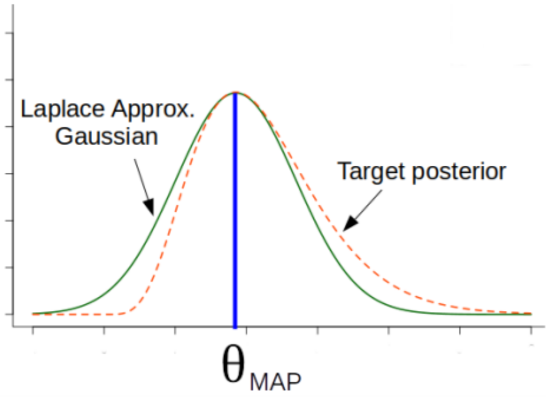
- Laplace’s Approximation (derivation and some properties)
- Exponential Family Distributions
- Generative Models for Supervised Learning

Laplace's Approximation

- Consider a posterior distribution that is intractable to compute

$$p(\theta|\mathcal{D}) = \frac{p(\mathcal{D},\theta)}{p(\mathcal{D})} = \frac{p(\mathcal{D}|\theta)p(\theta)}{p(\mathcal{D})}$$

- Laplace approximation approximates the above using a Gaussian distribution



$$p(\theta|\mathcal{D}) \approx \mathcal{N}(\theta|\theta_{MAP}, \mathbf{\Lambda}^{-1})$$

Tells us about the space (curvature) of the true posterior around θ_{MAP}

$$\theta_{MAP} = \text{argmax}_\theta \log p(\theta|\mathcal{D})$$

Negative of the Hessian, i.e., the second derivative of the log joint, at θ_{MAP}

$$\mathbf{\Lambda} = -\nabla_\theta^2 \log p(\theta|\mathcal{D})\Big|_{\theta=\theta_{MAP}} = -\nabla_\theta^2 \log p(\mathcal{D},\theta)\Big|_{\theta=\theta_{MAP}}$$

- Laplace's approx. is based on a second-order Taylor approx. of the posterior

Derivation of the Laplace's Approximation

$$\boxed{p(\mathcal{D}) \approx \exp\left(\log p(\mathcal{D}, \theta_{MAP})\right) \times (2\pi)^{D/2} \det(\boldsymbol{\Lambda})^{1/2}}$$

We also get a Laplace approximation of the marginal likelihood (for free!)

Note: Sometimes marginal likelihood is also called model evidence

- Let's write the Bayes rule as

$$p(\theta|\mathcal{D}) = \frac{p(\mathcal{D}, \theta)}{p(\mathcal{D})} = \frac{p(\mathcal{D}, \theta)}{\int p(\mathcal{D}, \theta) d\theta} = \frac{\exp[\log p(\mathcal{D}, \theta)]}{\int \exp[\log p(\mathcal{D}, \theta)] d\theta}$$

- Consider second-order Taylor approximation of a function $f(\theta)$ around some θ_0

$$f(\theta) \approx f(\theta_0) + (\theta - \theta_0)^\top \nabla_\theta f(\theta_0) + \frac{1}{2}(\theta - \theta_0)^\top \nabla_\theta^2 f(\theta_0)(\theta - \theta_0)$$

- Assuming $f(\theta) = \log p(\mathcal{D}, \theta)$ and $\theta_0 = \theta_{MAP}$

Constant w.r.t. θ

Same as $\nabla^2 \log p(\theta_{MAP}|\mathcal{D})$

$$\log p(\mathcal{D}, \theta) \approx \log p(\mathcal{D}, \theta_{MAP}) + \frac{1}{2}(\theta - \theta_{MAP})^\top \nabla_\theta^2 \log p(\mathcal{D}, \theta_{MAP})(\theta - \theta_{MAP})$$

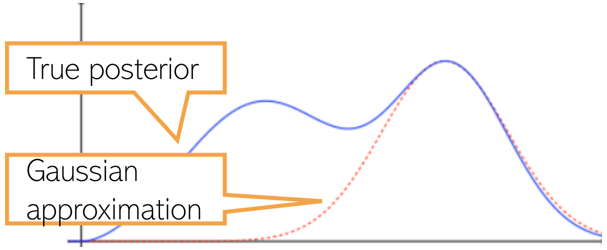
$$\boxed{\begin{aligned} p(\theta|\mathcal{D}) &\propto \exp\left[-\frac{1}{2}(\theta - \theta_{MAP})^\top(-\nabla_\theta^2 \log p(\mathcal{D}, \theta_{MAP}))(\theta - \theta_{MAP})\right] \\ &= \mathcal{N}(\theta|\theta_{MAP}, \boldsymbol{\Lambda}^{-1}) \quad (\text{where } \boldsymbol{\Lambda} = -\nabla_\theta^2 \log p(\mathcal{D}, \theta_{MAP}) = -\mathbf{H}) \end{aligned}}$$

Properties of Laplace’s Approximation

- Straightforward if posterior’s derivatives (first/second) can be computed easily
- Expensive if parameter $\boldsymbol{\theta}$ is very high dimensional

E.g., a deep neural network, or even in simpler models (e.g., logistic reg with a very large number of features

  - Reason: We need to compute and invert Hessian of size $D \times D$ (D is the # of params)
- Can do badly if the (true) posterior is multimodal

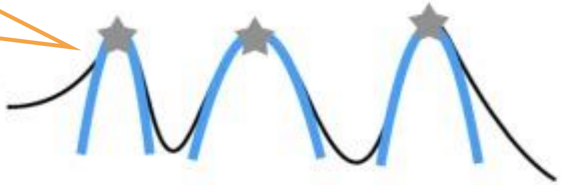


True posterior

Gaussian approximation

For multimodal posteriors, can use a mixture of Laplace approximations*

Useful for deep learning models



If K local modes, then define the approx. posterior as a mixture of K Gaussians

$$p(\theta|D) \approx \sum_{k=1}^{K} \pi^{(k)} \mathcal{N}(\theta|\theta_{MAP}^{(k)}, {H^{(k)}}^{-1})$$

(see paper cited below for details)

- Used only when $\boldsymbol{\theta}$ is a real-valued vector (because of Gaussian approximation)
- Note: Even if we have a <u>non-probabilistic</u> model (loss function + regularization), we can obtain an approx “posterior” for that model using the Laplace’s approximation
  - Optima of the regularized loss function will be Gaussian’s mean
  - Inverse of the second derivative of the regularized loss function will be covariance matrix

*Mixtures of Laplace Approximations for Improved Post-Hoc Uncertainty in Deep Learning (Eschenhagen et al, 2021)

Generalized Linear Models and Exponential Family

- (Probabilistic) Linear Regression: when response $\boldsymbol{y}$ is real-valued

$$p(y|\boldsymbol{x},\boldsymbol{w}) = \mathcal{N}(y|\boldsymbol{w}^\top\boldsymbol{x},\beta^{-1})$$

- Logistic Regression: when response $\boldsymbol{y}$ is binary (0/1)

$$p(y|\boldsymbol{x},\boldsymbol{w}) = \text{Bernoulli}[y|\sigma(\boldsymbol{w}^\top\boldsymbol{x})] = \left[\frac{\exp(\boldsymbol{w}^\top\boldsymbol{x})}{1+\exp(\boldsymbol{w}^\top\boldsymbol{x})}\right]^y\left[\frac{1}{1+\exp(\boldsymbol{w}^\top\boldsymbol{x})}\right]^{1-y}$$

- Both are examples of a Generalized Linear Model (GLM)
  - The model depends on the inputs $\boldsymbol{x}$ via a linear model $\boldsymbol{w}^\top\boldsymbol{x}$
- GLM is defined using an exponential family distribution

$$p(y|\boldsymbol{x},\boldsymbol{w}) = \text{ExpFam}[y|f(\boldsymbol{w}^\top\boldsymbol{x})]$$

MLE/MAP of $\boldsymbol{w}$ is easy for GLMs (due to convex objective, thanks to exp-family). Posterior usually requires approximations if likelihood and prior are not conjugate pairs (Laplace approximation or other methods used)

- ExpFam can be any suitable distribution depending on the nature of outputs, e.g.,
  - Gaussian for reals, Bernoulli for binary, Poisson for Count, gamma for positive reals
- ExpFam distributions are more generally useful in other contexts as well

Exp. Family (Pitman, Darmois, Koopman, 1930s)

- Defines a class of distributions. An Exponential Family distribution is of the form

$$\boxed{p(\mathbf{x}|\theta) = \frac{1}{Z(\theta)} h(\mathbf{x}) \exp[\theta^\top \phi(\mathbf{x})] \quad = \quad h(\mathbf{x}) \exp[\theta^\top \phi(\mathbf{x}) - A(\theta)]}$$

- $\boldsymbol{x} \in \mathcal{X}^m$ is the r.v. being modeled ($\mathcal{X}$ denotes some space, e.g., $\mathbb{R}$ or $\{0,1\}$)
- $\boldsymbol{\theta} \in \mathbb{R}^d$: Natural parameters or canonical parameters defining the distribution
- $\boldsymbol{\phi}(\boldsymbol{x}) \in \mathbb{R}^d$: Sufficient statistics (another random variable)
  - Knowing this quantity suffices to estimate parameter $\boldsymbol{\theta}$ from $\boldsymbol{x}$
- $Z(\boldsymbol{\theta}) = \int h(\boldsymbol{x})\exp[\boldsymbol{\theta}^\top\boldsymbol{\phi}(\boldsymbol{x})]d\boldsymbol{x}$: Partition Function
- $A(\boldsymbol{\theta}) = \log Z(\boldsymbol{\theta})$: Log-partition function (also called cumulant function)
- $\boldsymbol{h}(\boldsymbol{x})$: A constant (doesn't depend on $\boldsymbol{\theta}$)

Expressing a Distribution in Exp. Family Form

- Recall the form of exp-fam distribution $p(x|\theta) = h(x)\exp[\theta^\top \phi(x) - A(\theta)]$
- To write any exp-fam dist $p()$ in the above form, write it as $\exp(\log p())$

$$
\begin{aligned}
\exp\left(\log \text{Binomial}(x|N,\mu)\right) &= \exp\left(\log \binom{N}{x}\mu^x(1-\mu)^{N-x}\right) \\
&= \exp\left(\log \binom{N}{x} + x\log\mu + (N-x)\log(1-\mu)\right) \\
&= \binom{N}{x}\exp\left(x\log\frac{\mu}{1-\mu} - N\log(1-\mu)\right)
\end{aligned}
$$

- Now compare the resulting expression with the exponential family form

$$
p(x|\theta) = h(x)\exp[\theta^\top \phi(x) - A(\theta)]
$$

.. to identify the natural parameters, sufficient statistics, log-partition function, etc.

(Univariate) Gaussian as Exponential Family

- Let's try to write a univariate Gaussian in the exponential family form

$$p(\mathbf{x}|\theta) = h(\mathbf{x})\exp[\theta^\top \phi(\mathbf{x}) - A(\theta)]$$

- Recall the PDF of a univar Gaussian (already has exp, so less work needed :))

$$\mathcal{N}(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right] = \frac{1}{\sqrt{2\pi}}\exp\left[\frac{\mu}{\sigma^2}x - \frac{1}{2\sigma^2}x^2 - \frac{\mu^2}{2\sigma^2} - \log\sigma\right]$$

$$= \frac{1}{\sqrt{2\pi}}\exp\left[\begin{bmatrix}\frac{\mu}{\sigma^2}\\ -\frac{1}{2\sigma^2}\end{bmatrix}^\top \begin{bmatrix}x\\ x^2\end{bmatrix} - \left(\frac{\mu^2}{2\sigma^2} + \log\sigma\right)\right]$$

$$\theta = \begin{bmatrix}\frac{\mu}{\sigma^2}\\ -\frac{1}{2\sigma^2}\end{bmatrix} = \begin{bmatrix}\theta_1\\ \theta_2\end{bmatrix} \qquad \phi(x) = \begin{bmatrix}x\\ x^2\end{bmatrix} \quad \text{, and} \quad \begin{bmatrix}\mu\\ \sigma^2\end{bmatrix} = \begin{bmatrix}-\frac{\theta_1}{2\theta_2}\\ -\frac{1}{2\theta_2}\end{bmatrix}$$

$$h(x) = \frac{1}{\sqrt{2\pi}} \qquad A(\theta) = \frac{\mu^2}{2\sigma^2} + \log\sigma = \frac{-\theta_1^2}{4\theta_2} - \frac{1}{2}\log(-2\theta_2) - \frac{1}{2}\log(2\pi)$$

Other Examples

- Many other distribution belong to the exponential family
  - Bernoulli
  - Beta
  - Gamma
  - Multinoulli/Multinomial
  - Dirichlet
  - Multivariate Gaussian
  - .. and many more (https://en.wikipedia.org/wiki/Exponential_family)
- Note: Not all distributions belong to the exponential family, e.g.,
  - Uniform distribution ($x \sim \text{Unif}(a, b)$)
  - Student-t distribution
  - Mixture distributions (e.g., mixture of Gaussians)

Log-Partition Function

- The log-partition function is $A(\theta) = \log Z(\theta) = \log \int h(\boldsymbol{x}) \exp[\theta^\top \phi(\boldsymbol{x})] d\boldsymbol{x}$
- $\boldsymbol{A(\theta)}$ is also called the cumulant function
- Derivatives of $\boldsymbol{A(\theta)}$ can be used to generate the cumulants of the sufficient statistics
- Exercise: Assume $\boldsymbol{\theta}$ to be a scalar (thus $\boldsymbol{\phi(x)}$ is also scalar). Show that the first and the second derivatives of $\boldsymbol{A(\theta)}$ are

$$\begin{aligned} \frac{dA}{d\theta} &= \mathbb{E}_{p(\boldsymbol{x}|\theta)}[\phi(\boldsymbol{x})] \\ \frac{d^2A}{d\theta^2} &= \mathbb{E}_{p(\boldsymbol{x}|\theta)}[\phi^2(\boldsymbol{x})] - \left[\mathbb{E}_{p(\boldsymbol{x}|\theta)}[\phi(\boldsymbol{x})]\right]^2 = \text{var}[\phi(\boldsymbol{x})] \end{aligned}$$

- Above result also holds when $\boldsymbol{\theta}$ and $\boldsymbol{\phi(x)}$ are vector-valued (the “var” will be “covar”)
- Important: $\boldsymbol{A(\theta)}$ is a convex function of $\boldsymbol{\theta}$. Why?

MLE for Exponential Family Distributions

- Assume data $\mathcal{D} = \{x_1, \dots, x_N\}$ drawn i.i.d. from an exp. family distribution

$$p(x|\theta) = h(x)\exp[\theta^\top \phi(x) - A(\theta)]$$

- To do MLE, we need the overall likelihood -- a product of the individual likelihoods

$$p(\mathcal{D}|\theta) = \prod_{i=1}^{N} p(\mathbf{x}_i|\theta) = \left[\prod_{i=1}^{N} h(\mathbf{x}_i)\right] \exp\left[\theta^\top \sum_{i=1}^{N} \phi(\mathbf{x}_i) - NA(\theta)\right] = \left[\prod_{i=1}^{N} h(\mathbf{x}_i)\right] \exp\left[\theta^\top \phi(\mathcal{D}) - NA(\theta)\right]$$

- To estimate $\boldsymbol{\theta}$ (as we'll see shortly), we only need $\phi(\mathcal{D}) = \sum_{i=1}^{N} \phi(\mathbf{x}_i)$ and N
- Size of $\boldsymbol{\phi}(\mathcal{D}) = \sum_{i=1}^{N} \boldsymbol{\phi}(x_i)$ does not grow with $\boldsymbol{N}$ (same as the size of each $\boldsymbol{\phi}(x_i)$)
- Only exponential family distributions have finite-sized sufficient statistics
  - No need to store all the data; can simply update the sufficient statistics as data comes
  - Useful in probabilistic inference with large-scale data sets and "online" parameter estimation

Bayesian Inference for Expon. Family Distributions

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

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

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

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

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

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

- 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

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

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

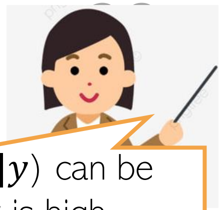
Can incorporate knowledge of frequency ("size") of each class in training data

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



- The generative classification model

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

Marginal probability of belonging to class k

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

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

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

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
$\boldsymbol{X}_\text{k} = \{\boldsymbol{x}_n : y_n = k\}$

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 $\boldsymbol{k}$ and normalize

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

PPD of y_*

PPD of $\boldsymbol{x}_*$

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

Compute for every value of $\boldsymbol{k}$ and normalize

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 $p(\boldsymbol{x}|y)$
- Can also use it for regression (define $p(\boldsymbol{x}, y)$ via some distr. and obtain $p(y|\boldsymbol{x})$ 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 $\boldsymbol{z}$ (see “Deep Hybrid Models: Bridging Discriminative and Generative Approaches“, UAI 2017)