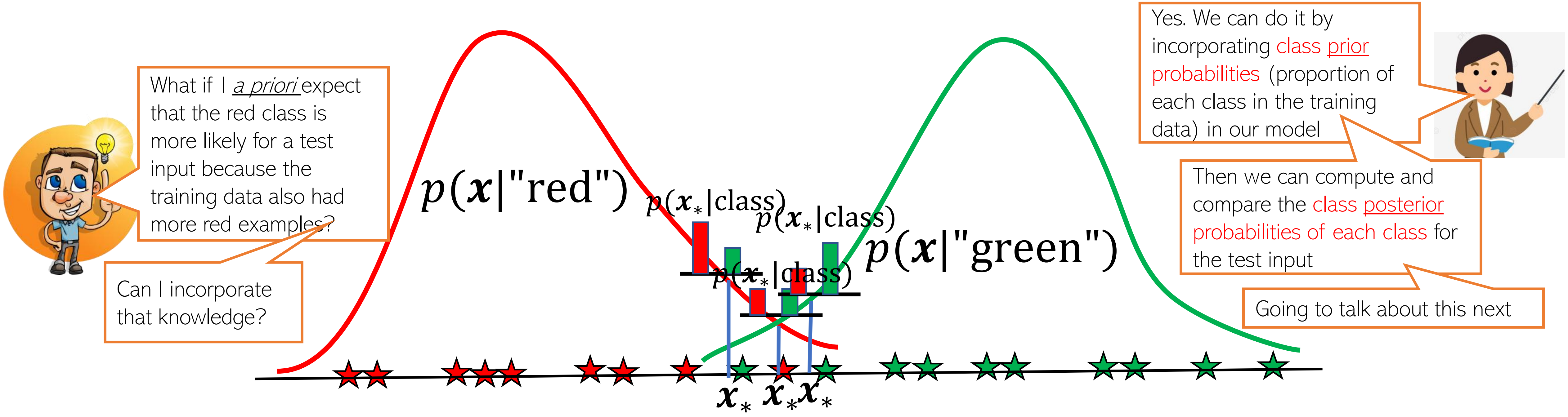
Probabilistic Models for Supervised Learning (3): Generative Classification and Regression

CS771: Introduction to Machine Learning

Piyush Rai

Generative Classification: A Basic Idea

- Learn the probability distribution of inputs from each class ("class-conditional")



- Usually assume some form (e.g., Gaussian) and estimate the parameters of that distribution (using MLE/MAP/fully Bayesian approach)
- Predict label of a test input $\boldsymbol{x}_*$ by comparing its probabilities under each class
  - Or can report the probability of belonging to each class (soft prediction)

Generative Classification: More Generally..

Roughly speaking, what's the fraction of each class in the training data

- Consider a classification problem with $K \geq 2$ classes
- The class prior probability of each class $k \in \{1,2,\dots,K\}$ is $p(y = k)$
- Can use Bayes rule to compute class posterior probability for a test input $\boldsymbol{x}_*$

Class prior distribution for class k

Class-conditional distribution of inputs from class k

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

This is just the marginal distribution of the joint distribution in the numerator (summed over all K values of y_*)

θ collectively denotes the parameters the joint distribution of inputs and labels depends on

Setting $p(y_* = k|\theta) = 1/K$ will give us the approach that predicts by comparing the probabilities $p(\boldsymbol{x}_*|y_* = k, \theta)$ of $\boldsymbol{x}_*$ under each of the classes

- We will first estimate the parameters of class prior and class-conditional distributions. Once estimated, we can use the above rule to predict the label for any test input
  - Can use MLE/MAP/fully Bayesian approach. <u>We will only consider MLE/MAP here</u>

Estimating Class Priors

Note: Can also do MAP estimation using a Dirichlet prior on π (this is akin to using Beta prior for doing MAP estimation for the bias of a coin). May try this as an exercise

- Estimating class priors $p(y = k)$ is usually straightforward in gen. classification
- Roughly speaking, it is the proportion of training examples from each class
  - Note: The above is true only when doing MLE (as we will see shortly)
  - If estimating class priors using MAP/fully Bayesian, they will be a "smooth version" of the proportions (because of the effect of regularization)
- The class prior distribution is assumed to be a discrete distribution (multinoulli)

$\pi_k = p(y = k)$

These probabilities sum to 1: $\sum_{k=1}^{K} \pi_k = 1$

Generalization of Bernoulli

$$p(y|\boldsymbol{\pi}) = \text{multinoulli}(y|\pi_1, \pi_2, \ldots, \pi_K) = \prod_{k=1}^{K} \pi_k^{\mathbb{I}[y=k]}$$

- Given N i.i.d. labelled examples $\{(x_n, y_n)\}_{n=1}^{N}$, $y_n \in \{1,2,\ldots,K\}$ the MLE soln

$$\boldsymbol{\pi}_{MLE} = \arg\max_{\boldsymbol{\pi}} \sum_{n=1}^{N} \log p(y_n|\boldsymbol{\pi})$$

Subject to constraint $\sum_{k=1}^{K} \pi_k = 1$

Can use Lagrange based opt. (note that we have an equality constraint)

Exercise: Verify that the MLE solution will be $p(y = k) = \pi_k = N_k/N$ where $N_k = \sum_{n=1}^{N} \mathbb{I}[y = k]$ (the frac. of class k examples)

Estimating Class-Conditionals

To be estimated using inputs from class k

- Can assume an appropriate distribution $p(\boldsymbol{x}|y = k, \theta)$ for inputs of each class
- If $\boldsymbol{x}$ is D-dim, it will be a D-dim. distribution. Choice depends on various factors
  - Nature of input features, e.g.,
    - If $\boldsymbol{x} \in \mathbb{R}^D$, can use a D-dim Gaussian $\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$
    - If $\boldsymbol{x} \in \{0,1\}^D$, can use D Bernoullis (one for each feature)
    - Can also choose more flexible/complex distributions if possible to estimate
  - Amount of training data available
    - With little data from a class, difficult to estimate the params of its class-cond. distribution

Some workarounds: Use strong regularization, or a simple form of the class-conditional (e.g., use a spherical/diagonal rather than a full covariance if the class-cond is Gaussian), or assume features are independent given class ("naïve Bayes" assumption)

A big issue especially if the number of features (D) is very large

- Once decided the form of class-cond, estimate $\boldsymbol{\theta}$ via MLE/MAP/Bayesian infer.
  - This essentially is a density estimation problem for the class-cond.
  - In principle, can use any density estimation method

Gen. Classifn. using Gaussian Class-conditionals

- The generative classification model $p(y=k|\boldsymbol{x}) = \frac{p(y=k)p(\boldsymbol{x}|y=k)}{p(\boldsymbol{x}|\theta)}$

A benefit of modeling each class by a distribution (recall that LwP had issues)

- Assume each class-conditional $p(\boldsymbol{x}|y=k)$ to be a Gaussian

$$\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{1}{\sqrt{(2\pi)^D|\boldsymbol{\Sigma}_k|}}\exp[-(\boldsymbol{x}-\boldsymbol{\mu}_k)^\top\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_k)]$$

Since the Gaussian's covariance models its shape, we can learn the shape of each class ☺

- Class prior is multinoulli (we already saw): $p(y=k) = \pi_k, \pi_k \in (0,1), \sum_{k=1}^K \pi_k = 1$
- Let's denote the parameters of the model collectively by $\boldsymbol{\theta} = \{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^K$
  - Can estimate these using MLE/MAP/Bayesian inference
  - Already saw the MLE solution for $\boldsymbol{\pi}$: $\pi_k = N_k/N$ (can also do MAP)
  - MLE solution for $\boldsymbol{\mu}_k = \frac{1}{N_k}\sum_{y_n=k}\boldsymbol{x}_n$, $\boldsymbol{\Sigma}_k = \frac{1}{N_k}\sum_{y_n=k}(\boldsymbol{x}_n-\boldsymbol{\mu}_k)(\boldsymbol{x}_n-\boldsymbol{\mu}_k)^\top$

Can also do MAP estimation for $\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k$ using a Gaussian prior on $\boldsymbol{\mu}_k$ and inverse Wishart prior on $\boldsymbol{\Sigma}_k$

Exercise: Try to derive this. I will provide a separate note containing the derivation

- If using point est (MLE/MAP) for $\boldsymbol{\theta}$, predictive distribution will be

Can predict the most likely class for the test input $\boldsymbol{x}_*$ by comparing these probabilities for all values of k

$$p(y_* = k|\mathbf{x}_*, \theta) = \frac{\pi_k|\boldsymbol{\Sigma}_k|^{-1/2}\exp\left[-\frac{1}{2}(\mathbf{x}_* - \boldsymbol{\mu}_k)^\top\boldsymbol{\Sigma}_k^{-1}(\mathbf{x}_* - \boldsymbol{\mu}_k)\right]}{\sum_{k=1}^K \pi_k|\boldsymbol{\Sigma}_k|^{-1/2}\exp\left[-\frac{1}{2}(\mathbf{x}_* - \boldsymbol{\mu}_k)^\top\boldsymbol{\Sigma}_k^{-1}(\mathbf{x}_* - \boldsymbol{\mu}_k)\right]}$$

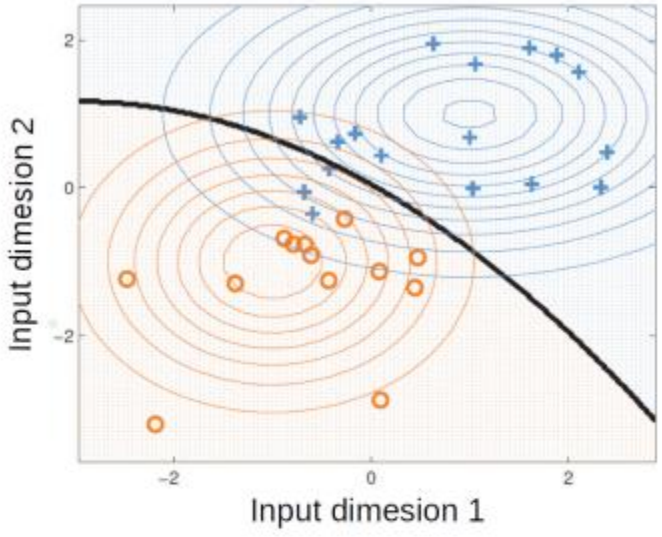
Note that the exponent has a Mahalanobis distance like term. Also, accounts for the fraction of training examples in class k

Decision Boundary with Gaussian Class-Conditional

- As we saw, the prediction rule when using Gaussian class-conditional

$$p(y=k|\mathbf{x},\theta) = \frac{\pi_k |\boldsymbol{\Sigma}_k|^{-1/2} \exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)\right]}{\sum_{k=1}^{K} \pi_k |\boldsymbol{\Sigma}_k|^{-1/2} \exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)\right]}$$

- The decision boundary between any pair of classes will be a quadratic curve



Reason: For any two classes k and k' at the decision boundary, we will have $p(y=k|x,\theta) = p(y=k'|x,\theta)$. Comparing their logs and ignoring terms that don't contain $\boldsymbol{x}$, can easily see that

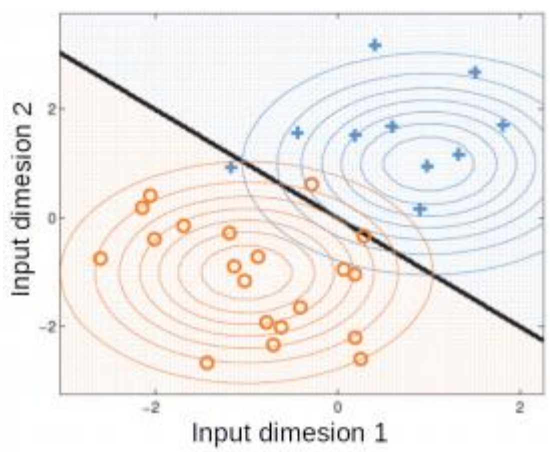
$$(\mathbf{x}-\boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1}(\mathbf{x}-\boldsymbol{\mu}_k) - (\mathbf{x}-\boldsymbol{\mu}_{k'})^\top \boldsymbol{\Sigma}_{k'}^{-1}(\mathbf{x}-\boldsymbol{\mu}_{k'}) = 0$$

Decision boundary contains all inputs $\boldsymbol{x}$ that satisfy the above

This is a quadratic function of $\boldsymbol{x}$ (this model is sometimes referred to Quadratic Discriminant Analysis)

Decision Boundary with Gaussian Class-Conditional

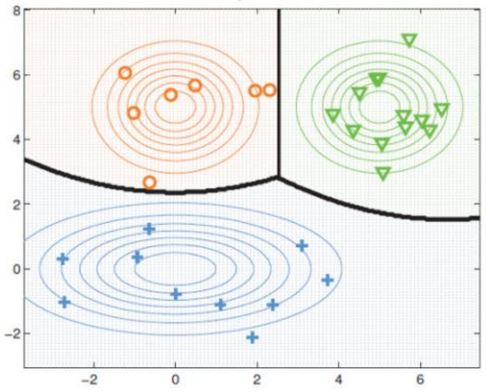
- Assume all classes are modeled using the same covariance matrix $\boldsymbol{\Sigma}_k = \boldsymbol{\Sigma}, \forall k$
- In this case, the decision boundary b/w any pair of classes will be linear

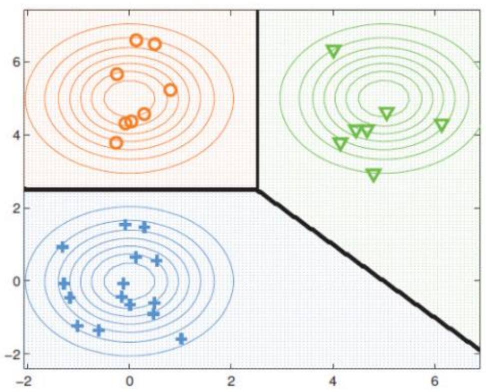


Reason: Again using $p(y = k|x, \theta) = p(y = k'|x, \theta)$, comparing their logs and ignoring terms that don't contain $\boldsymbol{x}$, we have

$$(\mathbf{x} - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu}_k) - (\mathbf{x} - \boldsymbol{\mu}_{k'})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu}_{k'}) = 0$$

Quadratic terms of $\boldsymbol{x}$ will cancel out; only linear terms will remain; hence decision boundary will be a linear function of $\boldsymbol{x}$ (Exercise: Verify that we can indeed write the decision boundary between this pair of classes as $\boldsymbol{w}^\top \boldsymbol{x} + b = 0$ where $\boldsymbol{w}$ and b depend on $\boldsymbol{\mu}_k, \boldsymbol{\mu}_{k'}$ and $\boldsymbol{\Sigma}$)





If we assume the covariance matrices of the assumed Gaussian class-conditionals for any pair of classes to be equal, then the learned separation boundary b/w this pair of classes will be linear; otherwise, quadratic as shown in the figure on left

A Closer Look at the Linear Case

- For the linear case (when $\boldsymbol{\Sigma}_k = \boldsymbol{\Sigma}, \forall k$), the class posterior probability

$$p(y = k|\mathbf{x}, \theta) \propto \pi_k \exp\left[-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu}_k)\right]$$

- Expanding further, we can write the above as

$$p(y = k|\mathbf{x}, \theta) \propto \exp\left[\boldsymbol{\mu}_k^\top \Sigma^{-1}\mathbf{x} - \frac{1}{2}\boldsymbol{\mu}_k^\top \Sigma^{-1}\boldsymbol{\mu}_k + \log \pi_k\right] \exp\left[\mathbf{x}^\top \Sigma^{-1}\mathbf{x}\right]$$

- Therefore, the above class posterior probability can be written as

$$p(y = k|\mathbf{x}, \theta) = \frac{\exp\left[\mathbf{w}_k^\top \mathbf{x} + b_k\right]}{\sum_{k=1}^K \exp\left[\mathbf{w}_k^\top \mathbf{x} + b_k\right]}$$

$\mathbf{w}_k = \Sigma^{-1}\boldsymbol{\mu}_k$ $\quad$ $b_k = -\frac{1}{2}\boldsymbol{\mu}_k^\top \Sigma^{-1}\boldsymbol{\mu}_k + \log \pi_k$

If all Gaussians class-cond have the same covariance matrix (basically, of all classes are assumed to have the same shape)

- The above has *exactly* the same form as softmax classification (thus softmax is a special case of a generative classification model with Gaussian class-conditionals)

A Very Special Case: LwP Revisited

- Note the prediction rule when $\boldsymbol{\Sigma}_k = \boldsymbol{\Sigma}, \forall k$

$$\hat{y} = \arg\max_k p(y=k|\mathbf{x}) = \arg\max_k \pi_k \exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)\right]$$

$$= \arg\max_k \log \pi_k - \frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)$$

- Also assume all classes to have equal no. of training examples, i.e., $\pi_k = 1/K$. Then

$$\hat{y} = \arg\min_k (\mathbf{x}-\boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)$$

The Mahalanobis distance matrix = $\boldsymbol{\Sigma}^{-1}$

- Equivalent to assigning $\boldsymbol{x}$ to the "closest" class in terms of a Mahalanobis distance
- If we further assume $\boldsymbol{\Sigma} = \boldsymbol{I}_D$ then the above is exactly the LwP rule

Generative Classification: Some Comments

- A simple but powerful approach to probabilistic classification
- Especially easy to learn if class-conditionals are simple
  - E.g., Gaussian with diagonal covariances ⇒ Gaussian naïve Bayes
  - Another popular model is multinomial naïve Bayes (widely used for document classification)
  - The naïve Bayes assumption: features are conditional independent given class label

$$p(\boldsymbol{x}|y=k) = \prod_{d=1}^{D} p(x_d|y=k)$$

Benefit: Instead of estimating a D-dim distribution which may be hard (if we don't have enough data), we will estimate D one-dim distributions (much simpler task)

- Can choose the form of class-conditionals $p(\boldsymbol{x}|y=k)$ based on the type of inputs $\boldsymbol{x}$

Will see such methods later

- Can handle missing data (e.g., if some part of the input $\boldsymbol{x}$ is missing) or missing labels

Will see such methods later

- Generative models are also useful for unsup. and semi-sup. learning

Generative Models for Regression

- Yes, we can even model regression problems using a generative approach
- Note that the output y is not longer discrete (so no notion of a class-conditional)
- However, the basic rule of recovering a conditional from joint would still apply

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

- Thus we can model the joint distribution $p(\boldsymbol{x},y|\theta)$ of features $\boldsymbol{x}$ and outputs $\boldsymbol{y} \in \mathbb{R}$
  - If features are real-valued the we can model $p(\boldsymbol{x},y|\theta)$ using a $(D+1)$-dim Gaussian
  - From this $(D+1)$-dim Gaussian, we can get $p(y|\boldsymbol{x},\theta)$using Gaussian conditioning formula
  - If joint is Gaussian, any subset of variables ($\boldsymbol{y}$ here), given the rest ($\boldsymbol{x}$ here) is also a Gaussian!
  - Refer to the Gaussian results from maths refresher slides for the result

Discriminative vs Generative

Proponents of discriminative models: Why bother modeling $\boldsymbol{x}$ if $\boldsymbol{y}$ is what you care about? Just model $\boldsymbol{y}$ directly instead of working hard to model x by learning the class-conditional

- Recall that discriminative approaches model $p(\boldsymbol{y}|\boldsymbol{x})$ directly
- Generative approaches model $p(\boldsymbol{y}|\boldsymbol{x})$ via $p(\boldsymbol{x}, \boldsymbol{y})$
- Number of parameters: Discriminative models have fewer parameters to be learned
  - Just the weight vector/matrix $\boldsymbol{w}/\boldsymbol{W}$ in case of logistic/softmax classification
- Ease of parameter estimation: Debatable as to which one is easier
  - For “simple” class-conditionals, easier for gen. classifn model (often closed-form solution)
  - Parameter estimation for discriminative models (logistic/softmax) usually requires iterative methods (although objective functions usually have global optima)
- Dealing with missing features: Generative models can handle this easily
  - E.g., by integrating out the missing features while estimating the parameters)
- Inputs with features having mixed types: Generative model can handle this
  - Appropriate $p(x_d|y)$ for each type of feature in the input. Difficult for discriminative models

Discriminative vs Generative (Contd)

- Leveraging unlabeled data: Generative models can handle this easily by treating the missing labels are latent variables and are ideal for Semi-supervised Learning. Discriminative models can't do it easily
- Adding data from new classes: Discriminative model will need to be re-trained on all classes all over again. Generative model will just require estimating the class-cond of newly added classes
- Have lots of labeled training data? Discriminative models usually work very well
- Final Verdict? Despite generative classification having some clear advantages, both methods can be quite powerful (the actual choice may be dictated by the problem)
  - Important to be aware of their strengths/weaknesses, and also the connections between these
- Possibility of a Hybrid Design? Yes, Generative and Disc. models can be combined, e.g.,
  - "Principled Hybrids of Generative and Discriminative Models" (Lassere et al, 2006)
  - "Deep Hybrid Models: Bridging Discriminative & Generative Approaches" (Kuleshov & Ermon, 2017)

Coming up next

- Large-margin hyperplane based classifiers (support vector machines)
- Kernel methods for learning nonlinear models