# Lecture 6:

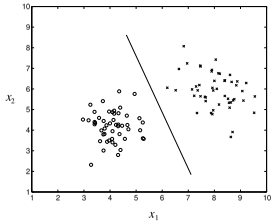
# Classification Models

Sam Roweis

January 23, 2002

## Classification

- Consider a discrete *class label* $y$ and some *features* $\mathbf{x}$.
- Goal: compute $p(y|\mathbf{x})$.
- Two approaches:
  *Generative*: model $p(\mathbf{x}, y) = p(y)p(\mathbf{x}|y)$ and use Bayes' rule to infer the conditional $p(y|\mathbf{x})$.
  *Discriminative*: model $p(y|\mathbf{x})$ directy.
- Generative approach is related to conditional *density estimation* while discriminative approach is closer to *regression*.

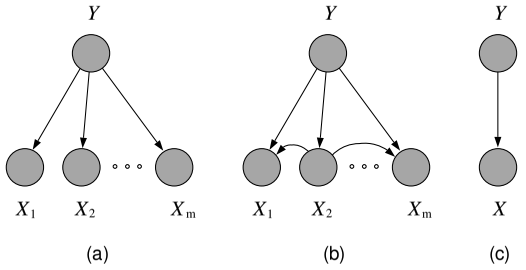


## Outline

- Generative vs. Discriminative Classification
- Gaussian-per-class
- Softmax/Linear result
- Naive Bayes
- Logistic/Softmax Regression
- Noisy-OR
- Others

## Generative Classification Models

- Model: $p(\mathbf{x}, y) = p(y)p(\mathbf{x}|y)$.
  $p(y)$ are called class *priors*.
  $p(\mathbf{x}|y)$ are called *class conditional feature distributions*.
- For the prior we use a Bernoulli or multinomial:
  $p(y = k|\pi) = \pi_k$ with $\sum_k \pi_k = 1$.
- In the class conditional distributions we can assume the features are independent, partly dependent, or fully dependent.

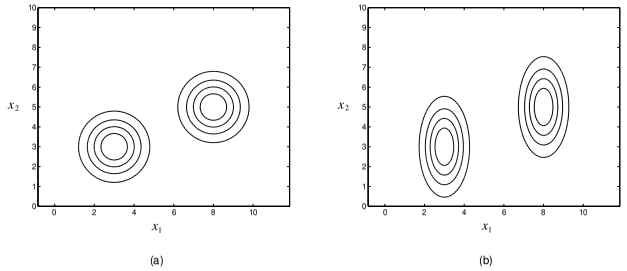


## Gaussian Class-Conditional Distributions

- If all features are continuous, a popular choice is a Gaussian class-conditional.

$$p(\mathbf{x}|y=k,\theta) = |2\pi\Sigma|^{-1/2}\exp\left\{-\frac{1}{2}(\mathbf{x}-\mu_k)\Sigma^{-1}(\mathbf{x}-\mu_k)\right\}$$

- Generally we assume all the covariances are the same. (Depending on the independence assumptions we make, the inverse covariance matrix is either full, sparse, or diagonal.)




## Softmax/Logistic

- The squashing function is known as the *softmax* or *logistic*:

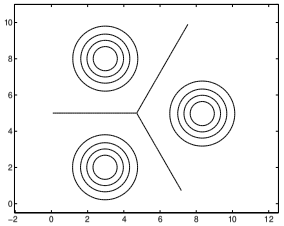
$$\phi_k(\mathbf{z}) \equiv \frac{e^{z_k}}{\sum_j e^{z_j}} \qquad g(\eta) = \frac{1}{1+e^{-\eta}}$$

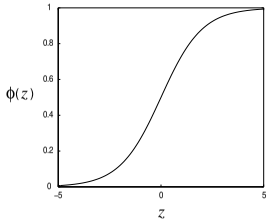
- It is invertible (up to a constant):

$$z_k = \log\phi_k + c \qquad \eta = \log(g/1-g)$$

- Derivative is easy:

$$\frac{\partial\phi_k}{\partial z_j} = \phi_k(\delta_{kj}-\phi_j) \qquad \frac{dg}{d\eta} = g(1-g)$$






## Gaussian Class-Conditional Distributions

- Compute the posterior via Bayes' rule:

$$\begin{aligned} p(y=k|\mathbf{x},\theta) &= \frac{p(\mathbf{x}|y=k,\theta)p(y=k|\pi)}{\sum_j p(\mathbf{x}|y=j,\theta)p(y=j|\pi)} \\ &= \frac{\exp\{\mu_k^\top\Sigma^{-1}\mathbf{x} - \mu_k^\top\Sigma^{-1}\mu_k/2 + \log\pi_k\}}{\sum_j \exp\{\mu_j^\top\Sigma^{-1}\mathbf{x} - \mu_j^\top\Sigma^{-1}\mu_j/2 + \log\pi_j\}} \\ &= \frac{e^{\beta_k^\top\mathbf{x}}}{\sum_j e^{\beta_j^\top\mathbf{x}}} \end{aligned}$$

  where $\beta_k = [\Sigma^{-1}\mu_k\,;\,(\mu_k^\top\Sigma^{-1}\mu_k + \log\pi_k)]$ and we have augmented $\mathbf{x}$ with a constant component always equal to 1 (bias term).

- Maximum likelihood estimates for parameters:
  priors $\pi_k$: use observed frequencies of classes
  means $\mu_k$: use class means
  covariance $\Sigma$: use pooled data $(\mathbf{x}^m - \mu_{y^m})$ to estimate covariance

## Linear Geometry

- Taking the ratio of any two posteriors (the "odds") shows that the contours of equal pairwise probability are linear surfaces in the feature space:

$$\frac{p(y=k|\mathbf{x},\theta)}{p(y=j|\mathbf{x},\theta)} = \exp\left\{(\beta_k-\beta_j)^\top\mathbf{x}\right\}$$

- The pairwise discrimination contours $p(y=k) = p(y=j)$ are orthogonal to the differences of the means in feature space when $\Sigma = \sigma I$. For general $\Sigma$ the same is true in the transformed feature space $\mathbf{w} = \Sigma^{-1}\mathbf{x}$.
- The priors do not change the geometry, they only shift the operating point on the logistic by the log-odds $\log(\pi_k/\pi_j)$.
- Thus, for equal class-covariances, we obtain a *linear classifier*. (If we use difference covariances, the arguments to the softmax are quadratic forms in $\mathbf{x}$, the decision surfaces are conic sections and we have a quadratic classifier.)

## Exponential Family Class-Conditionals

- The softmax form results whenever the class-conditional densities are *any* exponential family density:

$$
\begin{aligned}
p(\mathbf{x}|y=k,\eta_k) &= h(\mathbf{x})\exp\{\eta_k^\top \mathbf{x} - a(\eta_k)\} \\
p(y=k|\mathbf{x},\eta) &= \frac{p(\mathbf{x}|y=k,\eta_k)p(y=k|\pi)}{\sum_j p(\mathbf{x}|y=j,\eta_j)p(y=j|\pi)} \\
&= \frac{\exp\{\eta_k^\top \mathbf{x} - a(\eta_k)\}}{\sum_j \exp\{\eta_j^\top \mathbf{x} - a(\eta_j)\}} \\
&= \frac{e^{\beta_k^\top \mathbf{x}}}{\sum_j e^{\beta_j^\top \mathbf{x}}}
\end{aligned}
$$

  where $\beta_k = [\eta_k\,;\, -a(\eta_k)]$ and we have augmented $\mathbf{x}$ with a constant component always equal to 1 (bias term).

- Resulting classifier is linear in the sufficient statistics.

## Naive Bayes Likelihood

- $\ell(\theta;\mathcal{D}) = \sum_m \log p(y^m|\pi) + \sum_{mi} \log p(\mathbf{x}_i{}^m|y^m,\eta)$
- First term is easy.
  Second term, enforcing constraint with Lagrange multipliers:

$$
\begin{aligned}
\ell(\eta;\mathcal{D}) &= \sum_m \sum_{ijk} [\mathbf{x}_i{}^m = j][y^m = k]\log\eta_{ijk} + \sum_{ik}\lambda_{ik}(1-\textstyle\sum_j \eta_{ijk}) \\
\frac{\partial \ell}{\partial \eta_{ijk}} &= \frac{\sum_m [\mathbf{x}_i{}^m = j][y^m = k]}{\eta_{ijk}} - \lambda_{ik} \\
\lambda_{ik} &= \sum_m [y^m = k] \\
\Rightarrow \eta^*_{ijk} &= \frac{\sum_m [\mathbf{x}_i{}^m = j][y^m = k]}{\sum_m [y^m = k]}
\end{aligned}
$$

- ML parameters are class-conditional frequency counts.

## Naive Bayes Classifier

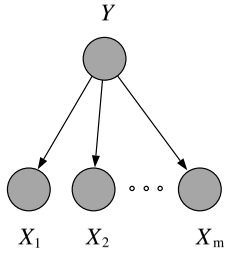
- Discrete features $\mathbf{x}_i$, assumed independent given the class label $y$.

$$
\begin{aligned}
p(\mathbf{x}_i = j|y=k) &= \eta_{ijk} \\
p(\mathbf{x}|y=k,\eta) &= \prod_i \prod_j \eta_{ijk}^{[\mathbf{x}_i=j]}
\end{aligned}
$$

Classification rule:

$$
\begin{aligned}
p(y=k|\mathbf{x},\eta) &= \frac{\pi_k \prod_i \prod_j \eta_{ijk}^{[\mathbf{x}_i=j]}}{\sum_q \pi_q \prod_i \prod_j \eta_{ijq}^{[\mathbf{x}_i=j]}} \\
&= \frac{e^{\beta_k^\top \mathbf{x}}}{\sum_q e^{\beta_q^\top \mathbf{x}}}
\end{aligned}
$$

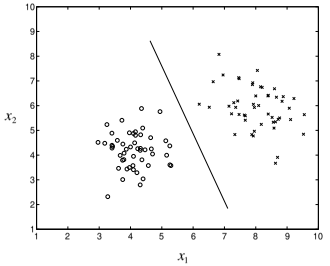
$\beta_k = \log[\eta_{11k} \ldots \eta_{1jk} \ldots \eta_{ijk} \ldots \log\pi_k]$
$\mathbf{x} = [\mathbf{x}_1{=}1; \mathbf{x}_1{=}2; \ldots; \mathbf{x}_i{=}j; \ldots; 1]$



(a)

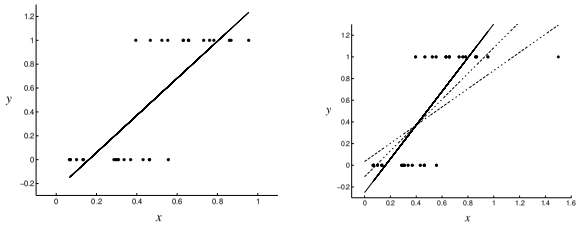
## Discriminative Models

- Parametrize $p(y|\mathbf{x})$ directly.
- Don't need to model the density of the features.
  Some density models have lots of parameters.
  Many densities give same linear classifier.
  But we cannot generate new labelled data.
- Optimize the same cost function we use at test time.



## Classification via Regression

- Binary case: $p(y = 1|\mathbf{x})$ is also the conditional expectation.
- So we could forget that $y$ was a discrete (categorical) random variable and just attempt to model $p(y|\mathbf{x})$ using regression.
- Very bad idea! Noise modes (e.g. Gaussian) for regression are totally inappropriate, and fits are oversensitive to outliers.




## Noisy-OR

- Many probabilistic models can be obtained as noisy versions of formulas from propositional logic.
- Noisy-OR: each input $\mathbf{x}_i$ activates output $y$ w/some probability.

$$p(y = 0|\mathbf{x}, \alpha) = \prod_i \alpha_i^{\mathbf{x}_i} = \exp\left\{\sum_i \mathbf{x}_i \log \alpha_i\right\}$$

- Letting $\theta_i = -\log \alpha_i$ we get another linear classifier:

$$p(y = 1|\mathbf{x}, \theta) = 1 - e^{-\theta^\top \mathbf{x}}$$

## Logistic/Softmax Regression

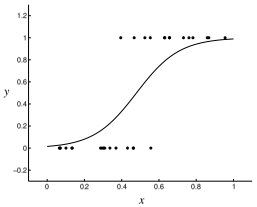
- Model: $y$ is a multinomial random variable whose posterior is the softmax of linear functions of the features.

$$p(y = k|\mathbf{x}, \theta) = \frac{e^{\theta_k^\top \mathbf{x}}}{\sum_j e^{\theta_j^\top \mathbf{x}}}$$

- Likelihood:

$$\ell(\theta; \mathcal{D}) = \sum_{mk} [y^m = k] \log p(y = k|\mathbf{x}^m, \theta) = \sum_{mk} y_k^m \log p_k^m$$

$$\begin{aligned}
\frac{\partial \ell}{\partial \theta_i} &= \sum_{mk} \frac{\partial \ell_k^m}{\partial p_k^m} \frac{\partial p_k^m}{\partial z_i^m} \frac{\partial z_i^m}{\partial \theta_i} \\
&= \sum_{mk} \frac{y_k^m}{p_k^m} p_k^m (\delta_{ik} - p_i^m) \mathbf{x}^m \\
&= \sum_m (y_k^m - p_k^m) \mathbf{x}^m
\end{aligned}$$



## Other Models

- Probit regression.
- Complementary log-log.
- Generalized linear models.
- Two-layer neural networks.
- Semiparametric
  (e.g. kernel classifiers, support vector machines, Gaussian processes).
- Non-parametric (e.g. K-nearest-neighbour).
- Some return a value for $y$ without a distribution.