Introduction to Machine Learning.
CSCI-UA 9473, Lecture 4.

Augustin Cosse

Ecole Normale Supérieure, DMA & NYU
Fondation Sciences Mathématiques de Paris.

2018

What have we seen so far? (I)

- Data distribution in nature are often highly complex
- Learning = understand the distribution from a few samples
- Two possible statistical approaches :
  - Bayesian : maximizes the posterior and relies on the definition of a prior
  - Frequentist : no prior but estimation through repeated samples (sampling distribution)
- Supervised Learning (patterns = (input +output) pairs) : Two classes of models
  - Regression (labels are continuous)
  - Classification (labels (classes) are discrete/finite)

What have we seen so far? (II)

- Among all possible regression models, the simplest = linear regression
- Linear regression can also be applied after non linear transformation of the data $X' = \phi(X)$ (Ex. $\phi(X) = X^2, \log(X), \ldots$)
- Quality of a prediction depends on the Bias variance tradeoff
- Generally speaking, as the model complexity increases, the variance tends to increase and the bias tends to decrease.
- Ideally, we want to trade bias off with variance to minimize the prediction/test error

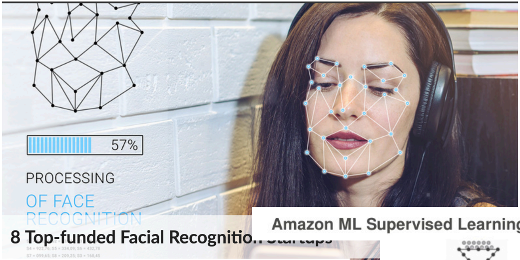
What have we seen so far? (III)

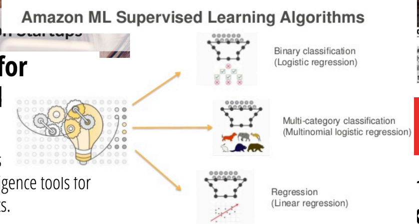
- When data is linear, linear regression has 0 bias.
- We can reduce the variance of the simple linear model by adding regularization

Formulation	Regularization
$\min_{\beta} \frac{1}{2}\|y - X\beta\|_2^2 + \lambda\|\beta\|_0$	Best subset selection
$\min_{\beta} \frac{1}{2}\|y - X\beta\|_2^2 + \lambda\|\beta\|_1$	Lasso regression
$\min_{\beta} \frac{1}{2}\|y - X\beta\|_2^2 + \lambda\|\beta\|_2^2$	Ridge regression

Today

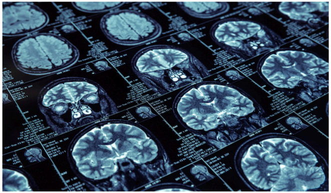
- Linear and generalized linear models for classification
- Examples in Python





Artificial Intelligence for Medical Imaging Marl Top $2B

Healthcare organizations are likely to s growing market around artificial intelligence tools for medical imaging, a new report predicts.



Source: Thinkstock

YouTube 8M

Dataset

ines

t Big Step for AI? Understanding Video

Technology Review

Last year, for example, Google released a set of eight million tagged YouTube videos called YouTube-8M. Facebook is developing an annotated data set of video actions called the Scenes, Actions, and Objects set.

- There are two main approaches at classification
  - First approach relies on the use of a discriminant function which assigns each vector x_i to a specific class $\mathcal{C}_k$
  - Second approach is to use a the conditional distribution $p(\mathcal{C}_k|\boldsymbol{x})$ in an inference stage and then use this posterior to make the decision.
- There are two ways to determine the conditional probability $p(\mathcal{C}_k|\boldsymbol{x})$
  - Either use a model for $p(\mathcal{C}_k|\boldsymbol{x})$ directly (discriminative approach)
  - Or use a model for the class conditional densities $p(\boldsymbol{x}|\mathcal{C}_k)$ together with a prior $p(\mathcal{C}_k)$ for the classes (generative approach).

$$p(\mathcal{C}_k|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\mathcal{C}_k)p(\mathcal{C}_k)}{p(\boldsymbol{x})}$$

Discriminant functions

- Linear classifiers = linear decision boundaries (possibly in augmented space)
- Simplest representation for a linear discriminant function is to take a linear function of the input

$$y(\boldsymbol{x}) = \boldsymbol{\beta}^T \boldsymbol{x} + \beta_0$$

- Recall that just as in regression, every algorithm we will cover is also applicable if we first apply a fixed non linear transformation of the input variables $\phi(X)$.

From two classes to multiple classes

- In the two classes cases, the simplest way to discriminate between the classes for a new pattern X_μ is to compute $y(\boldsymbol{x}) = \boldsymbol{\beta}^T \boldsymbol{x} + \beta_0$ and then set

$$\begin{array}{ll} \boldsymbol{x} \in \mathcal{C}_1 & y(x) \geq 0 \text{ (sometimes } y(x) \geq 1/2) \\ \boldsymbol{x} \in \mathcal{C}_2 & \text{otherwise.} \end{array}$$

- What do we do when there are multiple classes?
  - One possibility would be to define $K - 1$ classifiers each separating class $\mathcal{C}_k$ from the rest of the dataset (One vs rest)
  - Another approach could be to introduce $K(K-1)/2$ classifiers, discriminating between each pair of classes. A point would then be classified through a majority vote. (One vs One)

From two classes to multiple classes

From Bishop, Pattern recognition and ML

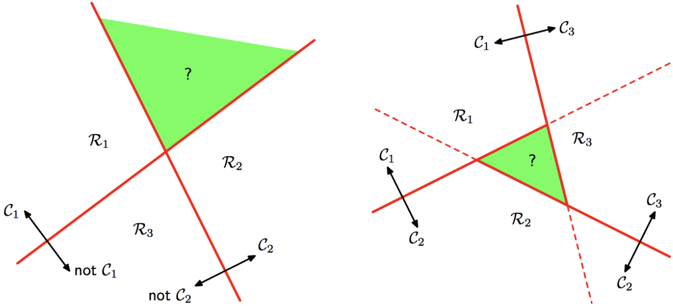


Figure 4.2 Attempting to construct a K class discriminant from a set of two class discriminants leads to ambiguous regions, shown in green. On the left is an example involving the use of two discriminants designed to distinguish points in class $\mathcal{C}_k$ from points not in class $\mathcal{C}_k$. On the right is an example involving three discriminant functions each of which is used to separate a pair of classes $\mathcal{C}_k$ and $\mathcal{C}_j$.

An alternative: Multiclass RSS

- Consider a set of patterns $X_1, X_2, \ldots, X_n$ that are grouped as rows $[1, X_k]$ in the matrix $\boldsymbol{X}$
- The class of each pattern X_k is described by a binary vector $y_k = (0, 1, 0, \ldots, 0)$
- We know from regression that (under some conditions) the model minimizing the RSS criterion can read as

$$\hat{\boldsymbol{Y}} = \boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{Y} = \boldsymbol{X}\hat{\boldsymbol{B}}$$

- $\hat{\boldsymbol{Y}}$ is called the indicator response matrix and $\hat{\boldsymbol{B}}$ is called the coefficient matrix

An alternative: Multiclass RSS

- For a new input X, we compute the output as

$$\hat{f}(X) = [1, X]^T \hat{\boldsymbol{B}}$$

- Thus getting values y_k from each of the classifiers β_k, $\hat{\boldsymbol{B}} = [\hat{\boldsymbol{\beta}}_1, \hat{\boldsymbol{\beta}}_2, \ldots, \hat{\boldsymbol{\beta}}_K]$ for the K classes

- To determine the class of X, we simply take class that outputs the largest label

$$\hat{G}(X) = \underset{k \in \mathcal{C}}{\text{argmax}}\ \hat{f}_k(X)$$

where $\hat{f}_k(X) = [1, X]^T \hat{\boldsymbol{\beta}}_k$ and $\hat{\boldsymbol{\beta}}_k$ is the k^{th} column of $\hat{\boldsymbol{B}}$.

RSS is not always a good idea (I)

- The discriminant RSS solution $\hat{\boldsymbol{Y}} = \boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{Y} = \boldsymbol{X}\hat{\boldsymbol{B}}$ suffers from some severe problems
  - First, The RSS solution penalizes solution that are "too" correct (lie a long way on the correct side of the decision boundary)
  - Second, the RSS solution corresponds to assuming a Gaussian distribution for the conditional density which is clearly not true (target vector t_k are far from Gaussian)
- An alternative is given by logistic regression which we will discuss below

RSS is not always a good idea (II)

C.M. Bishop, Pettern recognition and ML

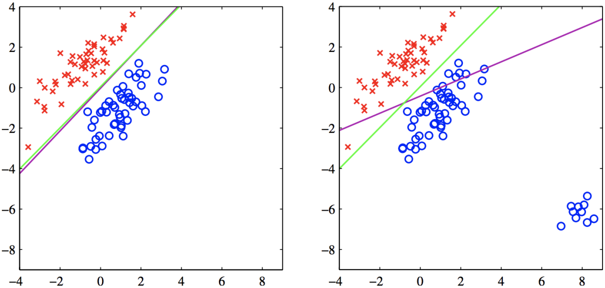


Figure 4.4 The left plot shows data from two classes, denoted by red crosses and blue circles, together with the decision boundary found by least squares (magenta curve) and also by the logistic regression model (green curve), which is discussed later in Section 4.3.2. The right-hand plot shows the corresponding results obtained when extra data points are added at the bottom left of the diagram, showing that least squares is highly sensitive to outliers, unlike logistic regression.

RSS is not always a good idea (II)

C.M. Bishop, Pettern recognition and ML

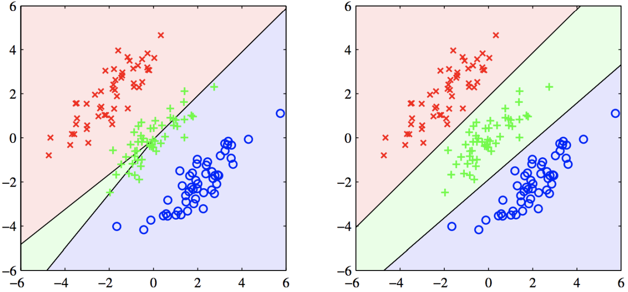


Figure 4.5 Example of a synthetic data set comprising three classes, with training data points denoted in red ($\times$), green ($+$), and blue ($\circ$). Lines denote the decision boundaries, and the background colours denote the respective classes of the decision regions. On the left is the result of using a least-squares discriminant. We see that the region of input space assigned to the green class is too small and so most of the points from this class are misclassified. On the right is the result of using logistic regressions as described in Section 4.3.2 showing correct classification of the training data.

Fisher's linear discriminant (I)

- Classification models can be though of as applying a dimenionality reduction step where we project the data points $\boldsymbol{x}$ onto the normal to the separating hyperplane $\boldsymbol{w}$, as $y = \boldsymbol{w}^T \boldsymbol{x}$
- When projecting high dimensional data on a one dimensional vector, we lose a lot of information
- By choosing $\boldsymbol{w}$ appropriately, one can select a projection that maximizes the class separation

Fisher's linear discriminant (II)

- Let μ_1 and μ_2 denote the class means

$$\boldsymbol{\mu}_1 = \frac{1}{N_1} \sum_{k \in \mathcal{C}_1} \boldsymbol{x}_k, \quad \boldsymbol{\mu}_2 = \frac{1}{N_2} \sum_{k \in \mathcal{C}_2} \boldsymbol{x}_k,$$

- One way to maximize separation could be to take $\boldsymbol{w}$ to maximize the separation of the projected class means

$$m_1 - m_2 = \boldsymbol{w}^T (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$$

- Simply maximizing the projected mean difference would lead to $\boldsymbol{w} = \infty$. An alternative would be to search only among normalized vectors (as this does not change orientation) $\|\boldsymbol{w}\|^2 = 1$.

Fisher's linear discriminant (III)

- The result is then a projection on the vector $\boldsymbol{w} = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)/\|(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)\|$ joining the two means.

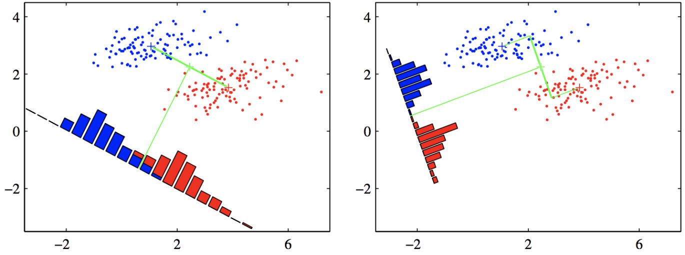


Figure 4.6 The left plot shows samples from two classes (depicted in red and blue) along with the histograms resulting from projection onto the line joining the class means. Note that there is considerable class overlap in the projected space. The right plot shows the corresponding projection based on the Fisher linear discriminant, showing the greatly improved class separation.

Fisher's linear discriminant (IV)

- An alternative (due to Fisher) tries to maintain a large separation of the projected class means while at the same time keeping a small variance within each class (minimize class overlap)

- The Fisher criterion maximizes the ratio of the separation of (projected) class means to the total (projected) within-class variance

$$J(\boldsymbol{w}) = \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2}$$

where

$$(Proj.Mean) \quad m_i = \boldsymbol{w}^T \boldsymbol{\mu}_i = (1/N_i) \sum_{k \in \mathcal{C}_i} \boldsymbol{w}^T \boldsymbol{x}_k,$$

$$(Proj.Variance) \quad s_1^2 = \sum_{k \in \mathcal{C}_1} (y_k - m_k)^2, \quad s_2^2 = \sum_{k \in \mathcal{C}_2} (y_k - m_k)^2$$

Fisher's linear discriminant (V)

- The Fisher criterion can read as a function of the unknown weight vector $\boldsymbol{w}$ as

$$J(\boldsymbol{w}) = \frac{\boldsymbol{w}^T \boldsymbol{B} \boldsymbol{w}}{\boldsymbol{w}^T \boldsymbol{H} \boldsymbol{w}}$$

with

$$\boldsymbol{B} = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T$$
$$\boldsymbol{H} = \sum_{k \in \mathcal{C}_1} (\boldsymbol{x}_k - \boldsymbol{\mu}_1)(\boldsymbol{x}_k - \boldsymbol{\mu}_1)^T + \sum_{k \in \mathcal{C}_2} (\boldsymbol{x}_k - \boldsymbol{\mu}_2)(\boldsymbol{x}_k - \boldsymbol{\mu}_2)^T$$

- Setting the derivative of $J(\boldsymbol{w})$ to zero gives

$$(\boldsymbol{w}^T \boldsymbol{B} \boldsymbol{w}) \boldsymbol{H} \boldsymbol{w} = (\boldsymbol{w}^T \boldsymbol{H} \boldsymbol{w}) \boldsymbol{B} \boldsymbol{w}$$

Fisher's linear discriminant (VI)

- Setting the derivative of $J(\boldsymbol{w})$ to zero gives

$$(\boldsymbol{w}^T \boldsymbol{B} \boldsymbol{w}) \boldsymbol{H} \boldsymbol{w} = (\boldsymbol{w}^T \boldsymbol{H} \boldsymbol{w}) \boldsymbol{B} \boldsymbol{w}$$

- If you solve this equation for the direction ($(\boldsymbol{w}^T \boldsymbol{B} \boldsymbol{w})$ and $(\boldsymbol{w}^T \boldsymbol{H} \boldsymbol{w})$ are scalars so we neglect them when trying to understand the direction of the separating plane), you get

$$\boldsymbol{w} \propto \boldsymbol{H}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$$

- This result is known as Fisher discriminant (although it is more a specific projection choice then a discriminant function as we will see in LDA)

- A similar result holds when solving the RSS criterion (exercice)

Fisher's linear discriminant (Multiple classes)

- When we have $K > 2$ classes, we need to introduce multiple features, $\boldsymbol{y} = (y_1, y_2, \ldots, y_K)$ (Think of a binary pattern for example)
- We then want to learn a separating hyperplane for each feature. Those planes are stacked in a matrix $\boldsymbol{W} = [\boldsymbol{w}_1, \ldots, \boldsymbol{w}_K]$ so that $\boldsymbol{y} = \boldsymbol{W}^T \boldsymbol{x}$

Fisher's linear discriminant (Multiple classes)

- One way to extend Fisher's criterion to multiple classes is to introduce the between class and within class covariances (after projection)

$$\boldsymbol{s}_W = \sum_{k=1}^{K} \sum_{i \in \mathcal{C}_k} (\boldsymbol{y}_i - \boldsymbol{m}_k)(\boldsymbol{y}_i - \boldsymbol{m}_k)^T$$

$$\boldsymbol{s}_B = \sum_{k=1}^{K} N_k (\boldsymbol{m}_k - \boldsymbol{m})(\boldsymbol{m}_k - \boldsymbol{m})^T$$

where $\boldsymbol{m} = (1/N) \sum_{k=1}^{K} N_k \boldsymbol{m}_k$

- And find a criterion that maximizes the ratio of the between class covariance to the within class covariance
- One example (Fukunaga): $J(\boldsymbol{w}) = \mathrm{Tr}(\boldsymbol{s}_W^{-1} \boldsymbol{s}_B)$

Linear Discriminant Analysis (I)

- Recall that Bayes gives (for class conditional densities f_k and priors π_k)

$$P(\mathcal{C}_k|X) = \frac{f_k(X)\pi_k}{\sum_{\ell=1}^{K} f_\ell(X)\pi_\ell}$$

- Then suppose we model the conditional class densities $f_k(X)$ ($\neq$ conditional densities $P(\mathcal{C}_k|X)$) using a multivariate Gaussian

$$f_k(x) = \frac{1}{(2\pi)^{p/2}|\boldsymbol{\Sigma}|^{1/2}} \exp(-\frac{1}{2}(\boldsymbol{x} - \mu_k)^T \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_k))$$

- LDA arises when we assume that the classes have the same covariance matrix $\boldsymbol{\Sigma}_k = \boldsymbol{\Sigma}\ \forall k$.

Linear Discriminant Analysis (II)

- To discriminate between classes, we can just look at the log ratio

$$\log\left(\frac{P(\mathcal{C}_k|X)}{P(\mathcal{C}_\ell|X)}\right) = \log\left(\frac{f_k(X)}{f_\ell(X)}\right) + \log(\frac{\pi_k}{\pi_\ell})$$
$$= \log\left(\frac{\pi_k}{\pi_\ell}\right) - \frac{1}{2}(\mu_k + \mu_\ell)^T \boldsymbol{\Sigma}^{-1}(\mu_k - \mu_\ell)$$
$$+ \boldsymbol{x}^T \boldsymbol{\Sigma}^{-1}(\mu_k - \mu_\ell)$$

- Equality between covariance matrices causes the quadratic terms and normalizing factors to cancel

- The decision boundary (set of points $\boldsymbol{x}$ for which $P(\mathcal{C}_k|X) = P(\mathcal{C}_\ell|X)$) is linear in $\boldsymbol{x}$.

Linear Discriminant Analysis (III)

- In particular we can now discriminate between classes using the linear discriminant functions δ_k,

$$\delta_k(\boldsymbol{x}) = \boldsymbol{x}^T \boldsymbol{\Sigma}^{-1} \mu_k - \frac{1}{2} \mu_k^T \boldsymbol{\Sigma}^{-1} \mu_k + \log(\pi_k)$$

- In practice we do not have access to the parameters of the Gaussian distributions and we have to estimate them empirically.
  - $\hat{\pi}_k = N_k / N$ ($N_k =$ number of observations in class $\mathcal{C}_k$)
  - $\hat{\mu}_k = \sum_{i \in \mathcal{C}_k} x_i / N_k$
  - $\hat{\boldsymbol{\Sigma}} = \sum_{k=1}^{K} \sum_{i \in \mathcal{C}_k} (x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^T / (N - K)$

Linear Discriminant Analysis (IV)

- From this, the class of a new point $\boldsymbol{x}$ is set by choosing the index k such that

$$\boldsymbol{x}^T \hat{\boldsymbol{\Sigma}}^{-1}(\hat{\mu}_k - \hat{\mu}_\ell) > \frac{1}{2}\hat{\mu}_k^T \hat{\boldsymbol{\Sigma}}^{-1}\hat{\mu}_k - \frac{1}{2}\hat{\mu}_\ell^T \hat{\boldsymbol{\Sigma}}^{-1}\hat{\mu}_\ell - \log(N_k/N_\ell)$$

- Note that if we choose to keep distinct covariance matrices, we end up with quadratic discriminant functions

$$\delta_k(\boldsymbol{x}) = -\frac{1}{2}\log(|\boldsymbol{\Sigma}_k|) - \frac{1}{2}(\boldsymbol{x} - \mu_k)^T \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x} - \mu_k) + \log(\pi_k)$$

Probabilistic classifiers

- For K classes we use a 1 of K coding scheme where

$$\boldsymbol{t} = \underbrace{(0, 1, 0, \ldots, 0)}_{K \text{ times}}$$

whenever the pattern $\boldsymbol{x}_\mu$ belongs to the class $\mathcal{C}_2$.

- t_k can be interpreted as the probability that the pattern belongs to the class $\mathcal{C}_k$

- For non probabilitic classifiers, other choices of target variables are possible $\{+1, -1\}$ for example

Generalized linear models

- When defining a probabilistic classifier, we will want to make sure that the posterior probabilities fall within the interval $[0, 1]$.
- In regression we used a model of the form

$$y(\boldsymbol{x}) = \beta_0 + \boldsymbol{\beta}^T \boldsymbol{x}$$

- Probabilistic classifiers generalize this model to a function of the form

$$y(\boldsymbol{x}) = f(\beta_0 + \boldsymbol{\beta}^T \boldsymbol{w})$$

- Here $f(\boldsymbol{x})$ is known as the activation function and maps the output of linear classifier to the $[0, 1]$ interval. The inverse of $f(\boldsymbol{x})$ is called the link function.

Generalized linear models

- Because of the non linear activation function, models such sa the one below are called generalized linear models

$$y(\boldsymbol{x}) = f(\beta_0 + \boldsymbol{\beta}^T \boldsymbol{w})$$

- An example of such a model is the perceptron classifier from Rosenblatt
- Another example is the logistic regression classifier

Logistic regression (I)

- The idea behind logistic regression is to model posterior probabilities $P(\mathcal{C}_k|X)$ as linear functions in x
- To ensure that the posterior probabilities sum to one and that they remain in the interval $[0, 1]$, we define the model as

$$\log\left(\frac{P(\mathcal{C}_1|X)}{P(\mathcal{C}_K|X)}\right) = \beta_{10} + \beta_1^T X$$

$$\log\left(\frac{P(\mathcal{C}_2|X)}{P(\mathcal{C}_K|X)}\right) = \beta_{20} + \beta_2^T X$$

$$\vdots$$

$$\log\left(\frac{P(\mathcal{C}_{K-1}|X)}{P(\mathcal{C}_K|X)}\right) = \beta_{(K-1)0} + \beta_K^T X$$

Logistic regression (II)

- You can check that

$$P(\mathcal{C}_k|X) = \frac{\exp(\beta_{k0} + \beta_k^T X)}{1 + \sum_{\ell=1}^{K-1} \exp(\beta_{\ell 0} + \beta_\ell^T X)}, \quad k = 1, \ldots, K-1.$$

$$P(\mathcal{C}_K|X) = \frac{1}{1 + \sum_{\ell=1}^{K-1} \exp(\beta_{\ell 0} + \beta_\ell^T X)}$$

- Logistic regression models are fit by Maximum likelihood

- Example: In the two classes framework, we let $y_i \in \{0,1\}$ denote the class ($\mathcal{C}_0$ or $\mathcal{C}_1$) of each point. I.e $y_i = 1$ is point x_i is classified in $\mathcal{C}_1$. The probability that a point X has the particular class $\mathcal{C} = y$ is thus given by

$$P(\mathcal{C} = y|X) = P(\mathcal{C}_1|X)^y P(\mathcal{C}_0|X)^{(1-y)}$$

Logistic regression (III)

- Now taking the log, and assuming the samples are independent, we get

$$\begin{aligned}\ell(\beta) &= \sum_{i=1}^{N} \{ y_i \log(P(\mathcal{C}_1|X)) + (1-y_i)\log(P(\mathcal{C}_0|X)) \} \\ &= \sum_{i=1}^{N} \left\{ y_i \beta^T x_i - \log(1 + e^{\beta^T x_i}) \right\}\end{aligned}$$

- To solve this with respect to β, apply the following Newton Raphson scheme

$$\beta_{k+1} \leftarrow \beta_k - \left(\frac{\partial^2 \ell(\beta)}{\partial\beta\partial\beta^T} \right)^{-1} \frac{\partial \ell(\beta)}{\partial \beta}$$

Discriminative vs Generative

- Remember the distinction between generative and discriminative classifier ?
- In which class does logistic regression fall ?

Another generative classifier: Naive Bayes (I)

- let $\boldsymbol{x} = (X_1, X_2, \ldots, X_d)$ denote a vector from the training set with features $X_1, \ldots, X_d$.
- The Naive Bayes classifier assumes that the features are independent
- Recall that using Bayes theorem, one can write the class posterior from given models for the class conditional densities $P(\boldsymbol{x}|\mathcal{C}_k) = f_k(\boldsymbol{x})$ and priors for the probability of each class $P(\mathcal{C}_k) = \pi_k$
- When features are independent, we can write $f_k(\boldsymbol{x})$ as

$$f_k(\boldsymbol{x}) = \prod_{\ell=1}^{d} f_{k\ell}(X_\ell) \tag{1}$$

Another generative classifier: Naive Bayes (II)

- From this, just as in LDA, we can write the log ratios

$$
\begin{aligned}
\log\left(\frac{P(\mathcal{C}_1|X)}{P(\mathcal{C}_K|X)}\right) &= \log\left(\frac{\pi_1 f_1(X)}{\pi_2 f_2(X)}\right) \\
&= \log\left(\frac{\pi_1}{\pi_2}\frac{\prod_{\ell=1}^{d} f_{\ell 1}(X_\ell)}{\prod_{\ell=1}^{d} f_{\ell,2}(X_\ell)}\right) \\
&= \log\left(\frac{\pi_1}{\pi_2}\right) + \sum_{\ell=1}^{d} \log\left(\frac{f_{\ell,1}(X_\ell)}{f_{\ell,2}(X_\ell)}\right) \\
&= \alpha_1 + \sum_{\ell=1}^{d} g_{1,\ell}(X_\ell)
\end{aligned}
$$