M2 Statistics & Data Science

Advanced Statistics & Machine Learning

Faicel Chamroukhi
Professeur
https://chamroukhi.com/

Overview

1 Classification

Classification (discrimination)

1. Classification
   - K-nearest neighbors (KNN)
   - Multi-class logistic regression
   - Neural Network
   - Gaussian Discriminant Analysis
   - Mixture Discriminant Analysis

Data Classification

- Given a training data set comprising n labeled observations $((\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n))$ where $\mathbf{x}$ denotes the observation (or the input) which is assumed to be continuous-valued in $\mathcal{X} = \mathbb{R}^d$

- y denotes the target variable (or the output) representing the class label which is a discrete-valued variable in $\mathcal{Y} = \{1, \ldots, K\}$

- K being the number of classes.

- In classification, the aim is to predict the value of the class label y for a new observation $\mathbf{x}$.

K-NN

- a direct supervised classification approach

K-NN

- a direct supervised classification approach
- Does not need "learning" but only storing the data

K-NN

- a direct supervised classification approach
- Does not need "learning" but only storing the data
- It's Son very simple : its principle is as follows : the class of a new data point is the one of its nearest neighbors (the majority among the K nearest neighbors) in the sense of a chosen distance (e.g, Euclidean distance)

$$d(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_{k=1}^{d} (x_{ik} - x_{jk})^2} \quad (1)$$

K-NN

- a direct supervised classification approach
- Does not need "learning" but only storing the data
- It's Son very simple : its principle is as follows : the class of a new data point is the one of its nearest neighbors (the majority among the K nearest neighbors) in the sense of a chosen distance (e.g, Euclidean distance)

$$d(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_{k=1}^{d} (x_{ik} - x_{jk})^2} \quad (1)$$

- $\Rightarrow$ As it needs computing, for each test data point, the distances with all the data points from the labeled training set, it may be computationally expensive for large data sets.

K-NN

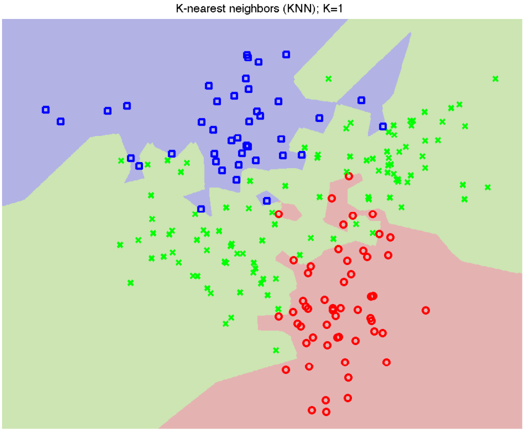
Algorithm 1 K-NN algorithm.

Inputs : Labeled data set : $\mathbf{X}^{\text{train}} = (\mathbf{x}_1^{\text{train}}, ..., \mathbf{x}_n^{\text{train}})$ and $\mathbf{y}^{\text{train}} = (y_1^{\text{train}}, ..., y_n^{\text{train}})$; Test data set $\mathbf{X}^{\text{test}} = (\mathbf{x}_1^{\text{test}}, ..., \mathbf{x}_m^{\text{test}})$; number of NN : K

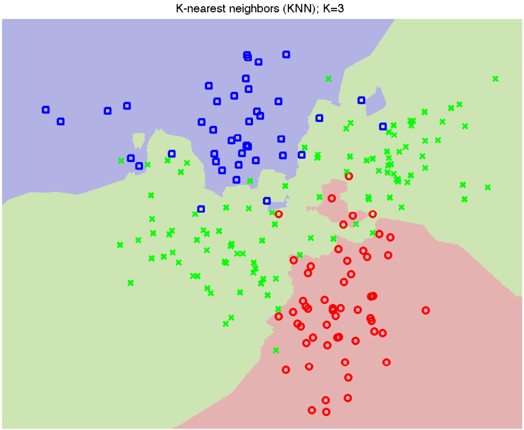
for $i = 1, \ldots, m$ **do**
  for $j = 1, \ldots, n$ **do**
    compute the Euclidean distances d_{ij} between $\mathbf{x}_i^{\text{test}}$and $\mathbf{x}_j^{\text{train}}$
    $\mathbf{d}_j \leftarrow \|\mathbf{x}_i - \mathbf{x}_j\|^2$
  end for
  The class y_i^{test} for the ith example is the one of its nearest neighbors :
  Sort the distance vector $\mathbf{d}_j$ in an increasing order for $j = 1, \ldots, n$
  Get at the same time the indexes of the elements in the new order
  Get the classes of the first K elements
  $\Rightarrow$ the class y_i^{test} is the majority class
end for

Output : Classes of the test data $\mathbf{y}^{\text{test}} = (y_1^{\text{test}}, ..., y_m^{\text{test}})$

K-NN



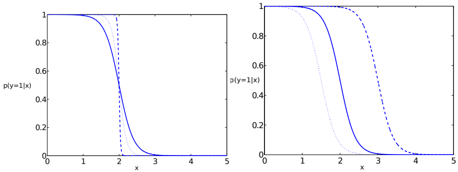
K-NN



Multi-class logistic regression

- a probabilistic supervised discriminative approach
- directly models the classes' posterior probabilities via :

$$p(y = k|\mathbf{x}) = \pi_k(\mathbf{x}; \mathbf{w}) = \frac{\exp(\boldsymbol{w}_k^T \mathbf{x})}{\sum_{h=1}^{K} \exp(\boldsymbol{w}_h^T \mathbf{x})}$$



- a logistic transformation of a linear function in $\mathbf{x}$
- ensures that the posterior probabilities are constrained to sum to one and remain in $[0, 1]$.
- The model parameter : $\mathbf{w} = (\boldsymbol{w}_1, \ldots, \boldsymbol{w}_K)^T$

Parameter estimation for Multi-class logistic regression

- The maximum likelihood is used to fit the model.
- The conditional log-likelihood of $\mathbf{w}$ for the given class labels $\mathbf{y} = (y_1, \ldots, y_n)$ conditionally on the inputs $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$:

$$\begin{aligned}\mathcal{L}(\mathbf{w}) = \mathcal{L}(\mathbf{w}; \mathbf{X}, \mathbf{y}) &= \log \prod_{i=1}^{n} p(y_i|\mathbf{x}_i; \mathbf{w}) \\ &= \log \prod_{i=1}^{n} \prod_{k=1}^{K} p(y_i = k|\mathbf{x}_i; \mathbf{w})^{y_{ik}} \\ &= \sum_{i=1}^{n} \sum_{k=1}^{K} y_{ik} \log \pi_k(\mathbf{x}_i; \mathbf{w})\end{aligned}$$

  where y_{ik} is an indicator binary variable such that $y_{ik} = 1$ if and only $y_i = k$ (i.e, $\mathbf{x}_i$ belongs to the class k).
- This log-likelihood is convex but can not be maximized in a closed form.
- The Newton-Raphson (NR) algorithm is generally used

Newton-Raphson for Multi-class logistic regression

- The Newton-Raphson algorithm is an iterative numerical optimization algorithm

- starts from an initial arbitrary solution $\mathbf{w}^{(0)}$, and updates the estimation of $\mathbf{w}$

- A single NR update is given by :

$$\mathbf{w}^{(l+1)} = \mathbf{w}^{(l)} - \left[\frac{\partial^2 \mathcal{L}(\mathbf{w})}{\partial \mathbf{w} \partial \mathbf{w}^T}\right]^{-1} \frac{\partial \mathcal{L}(\mathbf{w})}{\partial \mathbf{w}} \quad (2)$$

  where the Hessian and the gradient of $\mathcal{L}(\mathbf{w})$ (which are respectively the second and first derivative of $\mathcal{L}(\mathbf{w})$) are evaluated at $\mathbf{w} = \mathbf{w}^{(l)}$.

- NR can be stopped when the relative variation of $\mathcal{L}(\mathbf{w})$ is below a prefixed threshold.

IRLS I

The gradient component $\frac{\partial \mathcal{L}(\mathbf{w})}{\partial \boldsymbol{w}_h}$ $(h = 1, \ldots, K-1)$ is given by

$$\frac{\partial \mathcal{L}(\mathbf{w})}{\partial \boldsymbol{w}_h} = \sum_{i=1}^{n} \big(y_{ih} - \pi_h(\mathbf{x}_i; \mathbf{w})\big)\mathbf{x}_i$$

which can be formulated in a matrix form as

$$\frac{\partial \mathcal{L}(\mathbf{w})}{\partial \boldsymbol{w}_h} = \mathbf{X}^T(\mathbf{y}_h - \mathbf{p}_h)$$

where $\mathbf{X}$ is the $n \times (d+1)$ matrix whose rows are the input vectors $\mathbf{x}_i$, $\mathbf{y}_h$ is the $n \times 1$ column vector whose elements are the indicator variables y_{ih} for the hth logistic component :

$$\mathbf{y}_h = (y_{1h}, \ldots, y_{nh})^T$$

IRLS II

and $\mathbf{p}_h$ is the $n \times 1$ column vector of logistic probabilities corresponding to the ith input

$$\mathbf{p}_h = (\pi_h(\mathbf{x}_1; \mathbf{w}), \ldots, \pi_h(\mathbf{x}_n; \mathbf{w}))^T.$$

Thus, the matrix formulation of the gradient of $\mathcal{L}(\mathbf{w})$ for all the logistic components is

$$\frac{\partial \mathcal{L}(\mathbf{w})}{\partial \mathbf{w}} = \mathbf{X}^{*T}(\mathbf{Y} - \mathbf{P}) \tag{3}$$

where $\mathbf{Y} = (\mathbf{y}_1^T, \ldots, \mathbf{y}_{K-1}^T)^T$ and $\mathbf{P} = (\mathbf{p}_1^T, \ldots, \mathbf{p}_{K-1}^T)^T$ are $n \times (K-1)$ column vectors and $\mathbf{X}^*$ is the $(n \times (K-1))$ by $(d+1)$ matrix of $K-1$ copies of $\mathbf{X}$ such that $\mathbf{X}^* = (\mathbf{X}^T, \ldots, \mathbf{X}^T)^T$.
The Hessian matrix is composed of $(K-1) \times (K-1)$ block matrices where each block matrix is of dimension $(d+1) \times (d+1)$ and is given by :

$$\frac{\partial^2 \mathcal{L}(\mathbf{w})}{\partial \boldsymbol{w}_h \partial \boldsymbol{w}_k^T} = -\sum_{i=1}^{n} \pi_h(\mathbf{x}_i; \mathbf{w}) \left(\delta_{hk} - \pi_k(\mathbf{x}_i; \mathbf{w})\right) \mathbf{x}_i \mathbf{x}_i^T$$

which can be formulated in a matrix form as

$$\frac{\partial^2 \mathcal{L}(\mathbf{w})}{\partial \boldsymbol{w}_h \partial \boldsymbol{w}_k^T} = -\mathbf{X}^T \mathbf{W}_{hk} \mathbf{X}$$

where $\mathbf{W}_{hk}$ is the $n \times n$ diagonal matrix whose diagonal elements are $\pi_h(\mathbf{x}_i; \mathbf{w}) \left(\delta_{hk} - \pi_k(\mathbf{x}_i; \mathbf{w})\right)$ for $i = 1, \ldots, n$. For all the logistic components $(h, k = 1, \ldots, K-1)$, the Hessian takes the following form :

$$\frac{\partial^2 \mathcal{L}(\mathbf{w})}{\partial \mathbf{w} \partial \mathbf{w}^T} = -\mathbf{X}^{*T} \mathbf{W} \mathbf{X}^* \qquad (4)$$

where $\mathbf{W}$ is the $(n \times (K-1))$ by $(n \times (K-1))$ matrix composed of $(K-1)) \times (K-1))$ block matrices, each block is $\mathbf{W}_{hk}$ $(h, k = 1, \ldots, K-1)$. It can be shown that the Hessian matrix for the multi-class logistic regression model is positive semi definite and therefore the optimized log-likelihood is concave.
The NR algorithm (2) in this case can therefore be reformulated from the Equations (3) and (4) as

$$\begin{aligned}
\mathbf{w}^{(l+1)} &= \mathbf{w}^{(l)} + (\mathbf{X}^{*T}\mathbf{W}^{(l)}\mathbf{X}^{*})^{-1}\mathbf{X}^{*T}(\mathbf{Y}-\mathbf{P}^{(l)}) \\
&= (\mathbf{X}^{*T}\mathbf{W}^{(l)}\mathbf{X}^{*})^{-1}\left[\mathbf{X}^{*T}\mathbf{W}^{(l)}\mathbf{X}^{*}\mathbf{w}^{(l)} + \mathbf{X}^{*T}(\mathbf{Y}-\mathbf{P}^{(l)})\right] \\
&= (\mathbf{X}^{*T}\mathbf{W}^{(l)}\mathbf{X})^{-1}\mathbf{X}^{*T}\left[\mathbf{W}^{(l)}\mathbf{X}^{*}\mathbf{w}^{(l)} + (\mathbf{Y}-\mathbf{P}^{(l)})\right] \\
&= (\mathbf{X}^{*T}\mathbf{W}^{(l)}\mathbf{X}^{*})^{-1}\mathbf{X}^{*T}\mathbf{W}^{(l)}\mathbf{Y}^{*}
\end{aligned}$$

where $\mathbf{Y}^{*} = \mathbf{X}^{*}\mathbf{w}^{(l)} + (\mathbf{W}^{(l)})^{-1}(\mathbf{Y}-\mathbf{P}^{(l)})$ which yields in the Iteratively Reweighted Least Squares (IRLS) algorithm.

Neural Network

notes vues en cours

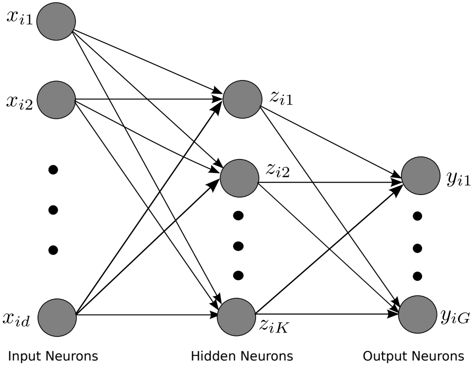


Figure – Graphical representation of Multi-Layer Perceptron (MLP).

Linear Discriminant Analysis

- generative approach that consists in modeling each conditional-class density by a multivariate Gaussian :

$$p(\mathbf{x}|y = k; \boldsymbol{\Psi}_k) = \frac{1}{(2\pi)^{\frac{d}{2}}|\boldsymbol{\Sigma}_k|^{\frac{1}{2}}} \exp\Big(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^T\boldsymbol{\Sigma}_k^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)\Big)$$

Linear Discriminant Analysis

- generative approach that consists in modeling each conditional-class density by a multivariate Gaussian :

$$p(\mathbf{x}|y=k;\boldsymbol{\Psi}_k) = \frac{1}{(2\pi)^{\frac{d}{2}}|\boldsymbol{\Sigma}_k|^{\frac{1}{2}}}\exp\Big(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^T\boldsymbol{\Sigma}_k^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)\Big)$$

- $\boldsymbol{\mu}_k \in \mathbb{R}^d$ is the mean vector
- $\boldsymbol{\Sigma}_k \in \mathbb{R}^{d\times d}$ is the covariance matrix
- $\boldsymbol{\Psi}_k = (\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ for $k = 1, \ldots, K$.

Linear Discriminant Analysis

- generative approach that consists in modeling each conditional-class density by a multivariate Gaussian :

$$p(\mathbf{x}|y = k; \boldsymbol{\Psi}_k) = \frac{1}{(2\pi)^{\frac{d}{2}}|\boldsymbol{\Sigma}_k|^{\frac{1}{2}}} \exp\Big(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1}(\mathbf{x} - \boldsymbol{\mu}_k)\Big)$$

- $\boldsymbol{\mu}_k \in \mathbb{R}^d$ is the mean vector
- $\boldsymbol{\Sigma}_k \in \mathbb{R}^{d\times d}$ is the covariance matrix
- $\boldsymbol{\Psi}_k = (\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ for $k = 1, \ldots, K$.

- Linear Discriminant Analysis (LDA) arises when we assume that all the classes have a common covariance matrix $\boldsymbol{\Sigma}_k = \boldsymbol{\Sigma}\ \forall k = 1, \ldots, K$.

Linear Discriminant Analysis

- The term “linear" in LDA is due to the fact that the decision boundaries between each pair of classes k and h are linear.

Linear Discriminant Analysis

- The term “linear" in LDA is due to the fact that the decision boundaries between each pair of classes k and h are linear.
- The decision boundary between classes k and h, which is the set of inputs $\mathbf{x}$ verifying $p(y=k|\mathbf{x}) = p(y=h|\mathbf{x})$, or by equivalence :

$$\log \frac{p(y=g|\mathbf{x};\boldsymbol{\Psi}_k)}{p(y=h|\mathbf{x};\boldsymbol{\Psi}_h)} = 0 \Leftrightarrow \log \frac{\pi_k}{\pi_h} + \log \frac{\mathcal{N}(\mathbf{x};\boldsymbol{\mu}_k,\boldsymbol{\Sigma})}{\mathcal{N}(\mathbf{x};\boldsymbol{\mu}_h,\boldsymbol{\Sigma})} =$$

Linear Discriminant Analysis

- The term "linear" in LDA is due to the fact that the decision boundaries between each pair of classes k and h are linear.

- The decision boundary between classes k and h, which is the set of inputs $\mathbf{x}$ verifying $p(y = k|\mathbf{x}) = p(y = h|\mathbf{x})$, or by equivalence :

$$\log \frac{p(y = g|\mathbf{x}; \boldsymbol{\Psi}_k)}{p(y = h|\mathbf{x}; \boldsymbol{\Psi}_h)} = 0 \Leftrightarrow \log \frac{\pi_k}{\pi_h} + \log \frac{\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma})}{\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_h, \boldsymbol{\Sigma})} =$$

$$\Leftrightarrow \log \tfrac{\pi_k}{\pi_h} - \tfrac{1}{2}(\boldsymbol{\mu}_k + \boldsymbol{\mu}_h)^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_k - \boldsymbol{\mu}_h) + \mathbf{x}^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_k - \boldsymbol{\mu}_h) = 0,$$

$\Rightarrow$ a linear function in $\mathbf{x}$ and therefore the classes will be separated by hyperplanes in the input space.

Linear Discriminant Analysis : Parameter Estimation

- Each of the class prior probabilities π_k is calculated with the proportion of the class g in the training data set :

$$\pi_k = \frac{\sum_{i|y_i=k}}{n} = \frac{n_k}{n}.$$

Linear Discriminant Analysis : Parameter Estimation

- Each of the class prior probabilities π_k is calculated with the proportion of the class g in the training data set :

$$\pi_k = \frac{\sum_{i|y_i=k}}{n} = \frac{n_k}{n}.$$

- The parameters $\boldsymbol{\Psi}_k$ are estimated by maximum likelihood

Linear Discriminant Analysis : Parameter Estimation

- Each of the class prior probabilities π_k is calculated with the proportion of the class g in the training data set :

$$\pi_k = \frac{\sum_{i|y_i=k}}{n} = \frac{n_k}{n}.$$

- The parameters $\boldsymbol{\Psi}_k$ are estimated by maximum likelihood
- the log-likelihood of $\boldsymbol{\Psi}_k$ given an i.i.d sample :

$$\mathcal{L}(\boldsymbol{\Psi}_k) = \log \prod_{i|y_i=k} \mathcal{N}(\mathbf{x}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}) = \sum_{i|y_i=k} \log \mathcal{N}(\mathbf{x}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}).$$

- $\Rightarrow$ The problem is solved in a closed form

$$\hat{\boldsymbol{\mu}}_k = \frac{1}{n_k} \sum_{i|y_i=k} \mathbf{x}_i,$$

$$\hat{\boldsymbol{\Sigma}} = \frac{1}{n-K} \sum_{k=1}^{K} \sum_{i|y_i=k} (\mathbf{x}_i - \hat{\boldsymbol{\mu}}_k)(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_k)^T,$$

Illustration

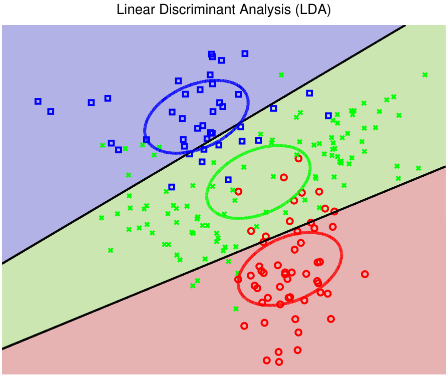


Figure – A three-classes classification example of a synthetic data set in which one of the classes occurs into two sub-classes, with training data points denoted in blue (□), green (×), and red (○). Ellipses denote the contours of the class probability density functions, lines denote the decision boundaries, and the background colors denote the respective classes of the decision regions. We see that LDA provides linear separation.

Quadratic Discriminant Analysis

- Quadratic Discriminant Analysis (QDA) is an extension of LDA that considers a different covariance matrix for each class.

Quadratic Discriminant Analysis

- Quadratic Discriminant Analysis (QDA) is an extension of LDA that considers a different covariance matrix for each class.
- The decision functions are quadratic :

$$\log \frac{p(y=k|\mathbf{x})}{p(y=h|\mathbf{x})} = \log \frac{\pi_k}{\pi_h} - \frac{1}{2}\log \frac{|\boldsymbol{\Sigma}_k|}{|\boldsymbol{\Sigma}_h|} - \frac{1}{2}\{(\mathbf{x}-\boldsymbol{\mu}_k)^T\boldsymbol{\Sigma}_k^{-1}(\mathbf{x}-\boldsymbol{\mu}_k) - (\mathbf{x}-\boldsymbol{\mu}_h)^T\boldsymbol{\Sigma}_h^{-1}(\mathbf{x}-\boldsymbol{\mu}_h)\} = 0.$$

$\Rightarrow$ This function is quadratic in $\mathbf{x}$, we then get quadratic discriminant functions in the input space.

Quadratic Discriminant Analysis

- Quadratic Discriminant Analysis (QDA) is an extension of LDA that considers a different covariance matrix for each class.
- The decision functions are quadratic :

$$\log \frac{p(y=k|\mathbf{x})}{p(y=h|\mathbf{x})} = \log \frac{\pi_k}{\pi_h} - \frac{1}{2} \log \frac{|\boldsymbol{\Sigma}_k|}{|\boldsymbol{\Sigma}_h|} - \frac{1}{2}\{(\mathbf{x}-\boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} (\mathbf{x}-\boldsymbol{\mu}_k) - (\mathbf{x}-\boldsymbol{\mu}_h)^T \boldsymbol{\Sigma}_h^{-1} (\mathbf{x}-\boldsymbol{\mu}_h)\} = 0.$$

$\Rightarrow$ This function is quadratic in $\mathbf{x}$, we then get quadratic discriminant functions in the input space.

- The parameters $\boldsymbol{\Psi}_k$ for QDA are estimated similarly as for LDA, except that separate covariance matrix must be estimated for each class :

$$\hat{\boldsymbol{\mu}}_k = \frac{1}{n_k} \sum_{i|y_i=k} \mathbf{x}_i$$

$$\hat{\boldsymbol{\Sigma}}_k = \frac{1}{n_k} \sum_{i|y_i=k} (\mathbf{x}_i - \hat{\boldsymbol{\mu}}_k)(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_k)^T.$$

Illustration

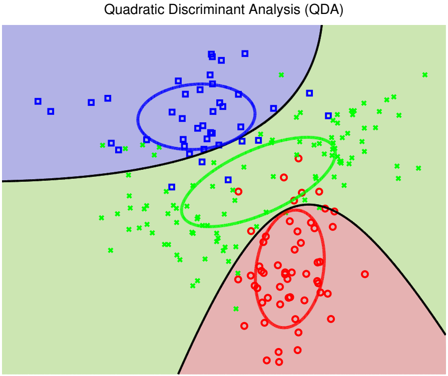


Figure – A three-classes classification example of a synthetic data set in which one of the classes occurs into two sub-classes, with training data points denoted in blue (□), green (×), and red (○). Ellipses denote the contours of the class probability density functions, lines denote the decision boundaries, and the background colors denote the respective classes of the decision regions. We see that QDA provides quadratic boundaries in the plan.

Mixture Discriminant Analysis

- for Gaussian discriminant analysis, in both LDA and QDA, each class density is modeled by a single Gaussian.

- This may be limited for modeling non homogeneous classes where the classes are dispersed.

  $\Rightarrow$ In Mixture Discriminant Analysis (MDA) each class density is modeled by a Gaussian mixture density

- with MDA, we can therefore capture many specific properties of real data such as multimodality, unobserved heterogeneity, heteroskedasticity, etc.

Mixture Discriminant Analysis (MDA)

- Each class g is modeled by a Gaussian mixture density :

$$p(\mathbf{x}|y = k; \boldsymbol{\Psi}_k) = \sum_{r=1}^{R_k} \pi_{kr} \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_{kr}, \boldsymbol{\Sigma}_{kr})$$

where R_k is the number of mixture components for class k

Mixture Discriminant Analysis (MDA)

- Each class g is modeled by a Gaussian mixture density :

$$p(\mathbf{x}|y = k; \boldsymbol{\Psi}_k) = \sum_{r=1}^{R_k} \pi_{kr} \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_{kr}, \boldsymbol{\Sigma}_{kr})$$

  where R_k is the number of mixture components for class k

- $$\boldsymbol{\Psi}_k = (\pi_{k1}, \ldots, \pi_{kR_k}, \boldsymbol{\mu}_{k1}, \ldots, \boldsymbol{\mu}_{kR_k}, \ldots, \boldsymbol{\Sigma}_{k1}, \ldots, \boldsymbol{\Sigma}_{kR_k})$$

  is the parameter vector of the mixture density of class k

Mixture Discriminant Analysis (MDA)

- Each class g is modeled by a Gaussian mixture density :

$$p(\mathbf{x}|y=k;\boldsymbol{\Psi}_k)=\sum_{r=1}^{R_k}\pi_{kr}\mathcal{N}(\mathbf{x};\boldsymbol{\mu}_{kr},\boldsymbol{\Sigma}_{kr})$$

  where R_k is the number of mixture components for class k

- $$\boldsymbol{\Psi}_k=(\pi_{k1},\ldots,\pi_{kR_k},\boldsymbol{\mu}_{k1},\ldots,\boldsymbol{\mu}_{kR_k},\ldots,\boldsymbol{\Sigma}_{k1},\ldots,\boldsymbol{\Sigma}_{kR_k})$$

  is the parameter vector of the mixture density of class k

- the π_{kr}'s ($r=1,\ldots,R_k$) are the non-negative mixing proportions satisfying $\sum_{r=1}^{R_k}\pi_{kr}=1\ \forall k$.

Mixture Discriminant Analysis (MDA)

- Each class g is modeled by a Gaussian mixture density :

$$p(\mathbf{x}|y = k; \boldsymbol{\Psi}_k) = \sum_{r=1}^{R_k} \pi_{kr} \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_{kr}, \boldsymbol{\Sigma}_{kr})$$

  where R_k is the number of mixture components for class k

- $$\boldsymbol{\Psi}_k = (\pi_{k1}, \ldots, \pi_{kR_k}, \boldsymbol{\mu}_{k1}, \ldots, \boldsymbol{\mu}_{kR_k}, \ldots, \boldsymbol{\Sigma}_{k1}, \ldots, \boldsymbol{\Sigma}_{kR_k})$$

  is the parameter vector of the mixture density of class k
- the π_{kr}'s $(r = 1, \ldots, R_k)$ are the non-negative mixing proportions satisfying $\sum_{r=1}^{R_k} \pi_{kr} = 1 \; \forall k$.
- we can allow a different covariance matrix for each mixture component as well as a common covariance matrix

Illustration

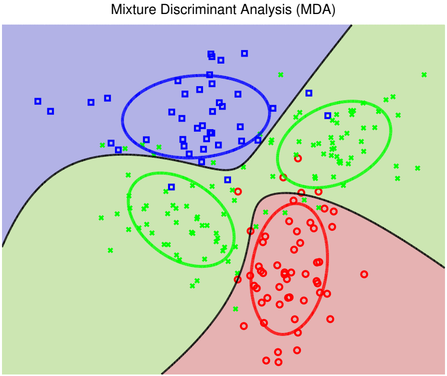


Figure – A three-classes classification example of a synthetic data set in which one of the classes occurs into two sub-classes, with training data points denoted in blue (□), green (×), and red (○). Ellipses denote the contours of the class probability density functions, lines denote the decision boundaries, and the background colors denote the respective classes of the decision regions. We see that MDA provides more non-linear decision boundaries.

Illustrations of Logistic Regression, LDA, QDA and MDA

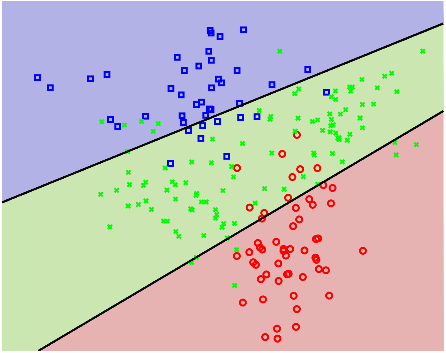


Figure – A three-classes classification example of a synthetic data set in which one of the classes occurs into two sub-classes, with training data points denoted in blue (□), green (×), and red (○). Ellipses denote the contours of the class probability density functions, lines denote the decision boundaries, and the background colors denote the respective classes of the decision regions. We see that both LDA and Logistic regression provide linear separation, while QDA and MDA provide non linear separation. MDA can further deal the problem of heterogeneous classes.

Illustrations of Logistic Regression, LDA, QDA and MDA

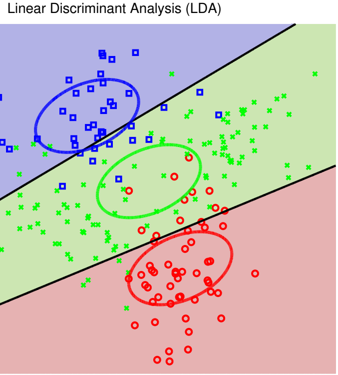


Figure – A three-classes classification example of a synthetic data set in which one of the classes occurs into two sub-classes, with training data points denoted in blue (□), green (×), and red (○). Ellipses denote the contours of the class probability density functions, lines denote the decision boundaries, and the background colors denote the respective classes of the decision regions. We see that both LDA and Logistic regression provide linear separation, while QDA and MDA provide non linear separation. MDA can further deal the problem of heterogeneous classes.

Illustrations of Logistic Regression, LDA, QDA and MDA

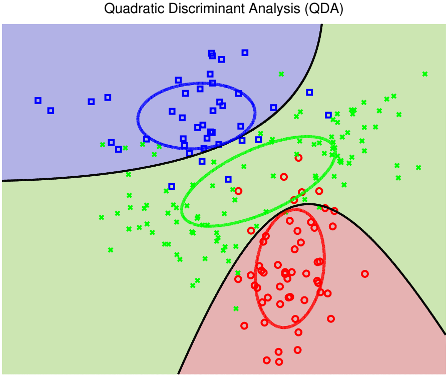


Figure – A three-classes classification example of a synthetic data set in which one of the classes occurs into two sub-classes, with training data points denoted in blue (□), green (×), and red (○). Ellipses denote the contours of the class probability density functions, lines denote the decision boundaries, and the background colors denote the respective classes of the decision regions. We see that both LDA and Logistic regression provide linear separation, while QDA and MDA provide non linear separation. MDA can further deal the problem of heterogeneous classes.

Illustrations of Logistic Regression, LDA, QDA and MDA

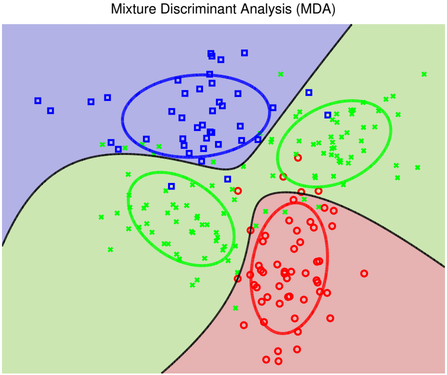


Figure – A three-classes classification example of a synthetic data set in which one of the classes occurs into two sub-classes, with training data points denoted in blue (□), green (×), and red (○). Ellipses denote the contours of the class probability density functions, lines denote the decision boundaries, and the background colors denote the respective classes of the decision regions. We see that both LDA and Logistic regression provide linear separation, while QDA and MDA provide non linear separation. MDA can further deal the problem of heterogeneous classes.