Linear Discriminant Analysis vs. Kernel Fisher Discriminant Analysis

Samuel Schulter & Christoph Zechner

Graz University of Technology

November 25, 2009

Motivation

- Supervised Classification is one of the main components in Machine Learning

Motivation

- Supervised Classification is one of the main components in Machine Learning
- The goal is to find a Classification Algorithm which is able to separate the data

Motivation

- Supervised Classification is one of the main components in Machine Learning
- The goal is to find a Classification Algorithm which is able to separate the data
- while beeing as *simple* as possible

Motivation

- Supervised Classification is one of the main components in Machine Learning
- The goal is to find a Classification Algorithm which is able to separate the data
- while beeing as *simple* as possible
- Usually simple Algorithms need only a small number of training examples to work well enough

Motivation

- Supervised Classification is one of the main components in Machine Learning
- The goal is to find a Classification Algorithm which is able to separate the data
- while beeing as *simple* as possible
- Usually simple Algorithms need only a small number of training examples to work well enough
- LDA is based on quite simple assumptions

Motivation

- Supervised Classification is one of the main components in Machine Learning
- The goal is to find a Classification Algorithm which is able to separate the data
- while beeing as *simple* as possible
- Usually simple Algorithms need only a small number of training examples to work well enough
- LDA is based on quite simple assumptions
- Extension by Kernels allows solving more complex problems efficiently

Table of contents

Supervised Learning

- Given some training examples $x_i \in \mathcal{X} \quad i = 0..N-1$

Supervised Learning

- Given some training examples $x_i \in \mathcal{X} \quad i = 0..N-1$
- and given the corresponding class labels $Y_i \in \mathcal{Y} \quad i = 0..N-1$,

Supervised Learning

- Given some training examples $x_i \in \mathcal{X} \quad i = 0..N-1$
- and given the corresponding class labels $Y_i \in \mathcal{Y} \quad i = 0..N-1$,
- try to find the "best" hypothesis $H_{opt} \in \mathcal{H} : \mathcal{X} \rightarrow \mathcal{Y}$.

Supervised Learning

- Given some training examples $x_i \in \mathcal{X} \quad i = 0..N-1$
- and given the corresponding class labels $Y_i \in \mathcal{Y} \quad i = 0..N-1$,
- try to find the "best" hypothesis $H_{opt} \in \mathcal{H} : \mathcal{X} \rightarrow \mathcal{Y}$.
- So learning is a mapping $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y} \rightarrow \mathcal{H} \quad i = 0..N-1$

Supervised Learning

- Given some training examples $x_i \in \mathcal{X} \quad i = 0..N-1$
- and given the corresponding class labels $Y_i \in \mathcal{Y} \quad i = 0..N-1$,
- try to find the "best" hypothesis $H_{opt} \in \mathcal{H} : \mathcal{X} \rightarrow \mathcal{Y}$.
- So learning is a mapping $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y} \rightarrow \mathcal{H} \quad i = 0..N-1$

- But what means "best"??

Optimal Hypothesis

- We need to measure the quality of a hypothesis

Optimal Hypothesis

- We need to measure the quality of a hypothesis
- One possibility: $error(H) = E\{d(H(x), y)\}$

Optimal Hypothesis

- We need to measure the quality of a hypothesis
- One possibility: $error(H) = E\{d(H(x), y)\}$
- Typically with $d : \mathcal{Y} \times \mathcal{Y} \rightarrow \begin{cases} 0 & x = y \\ 1 & x \neq y \end{cases}$

Optimal Hypothesis

- We need to measure the quality of a hypothesis
- One possibility: $error(H) = E\{d(H(x), y)\}$
- Typically with $d : \mathcal{Y} \times \mathcal{Y} \rightarrow \begin{cases} 0 & x = y \\ 1 & x \neq y \end{cases}$

Optimality Criterion

The optimal hypothesis H_{opt} minimizes the expectation of the classification error:

$$H_{opt} = \underset{H \in \mathcal{H}}{arg\,min}\, E\{d(H(x), y)\} \tag{1}$$

Optimal Hypothesis

- We need to measure the quality of a hypothesis
- One possibility: $error(H) = E\{d(H(x), y)\}$
- Typically with $d : \mathcal{Y} \times \mathcal{Y} \rightarrow \begin{cases} 0 & x = y \\ 1 & x \neq y \end{cases}$

Optimality Criterion

The optimal hypothesis H_{opt} minimizes the expectation of the classification error:

$$H_{opt} = \underset{H \in \mathcal{H}}{arg\,min}\, E\{d(H(x), y)\} \tag{1}$$

- **Problem:** Typically the hypothesis space $\mathcal{H}$ is extremely large.

Optimal Hypothesis

- We need to measure the quality of a hypothesis
- One possibility: $error(H) = E\{d(H(x), y)\}$
- Typically with $d : \mathcal{Y} \times \mathcal{Y} \rightarrow \begin{cases} 0 & x = y \\ 1 & x \neq y \end{cases}$

Optimality Criterion

The optimal hypothesis H_{opt} minimizes the expectation of the classification error:

$$H_{opt} = \underset{H \in \mathcal{H}}{arg\,min}\, E\{d(H(x), y)\} \tag{1}$$

- **Problem:** Typically the hypothesis space $\mathcal{H}$ is extremely large.
- **Possible solution:** Restrict $\mathcal{H}$ to a special class of parameterized hypotheses and try to learn the parameters (e.g. polynomial base functions).

How to decide between classes?

- Every hypotheses produces a decision boundary ($\rightarrow$ discriminant function)

How to decide between classes?

- Every hypotheses produces a decision boundary ($\rightarrow$ discriminant function)
- Structure depends on the hypothesis class $\mathcal{H}$ (e.g. linear, RBF,...)

How to decide between classes?

- Every hypotheses produces a decision boundary ($\rightarrow$ discriminant function)
- Structure depends on the hypothesis class $\mathcal{H}$ (e.g. linear, RBF,...)
- Best hypothesis class is strongly depending on problem

How to decide between classes?

- Every hypotheses produces a decision boundary ($\rightarrow$ discriminant function)
- Structure depends on the hypothesis class $\mathcal{H}$ (e.g. linear, RBF,...)
- Best hypothesis class is strongly depending on problem
- $\rightarrow$ class seperability!

Class Seperability

- **Linear Seperability**: We call two classes linearly seperable if they can be seperated by a linear discriminant function

Class Seperability

- **Linear Seperability**: We call two classes linearly seperable if they can be seperated by a linear discriminant function
- Nice to have, but quite rare

Class Seperability

- **Linear Seperability**: We call two classes linearly seperable if they can be seperated by a linear discriminant function
- Nice to have, but quite rare
- In all other cases we will need nonlinear discriminant functions

Examples (1)

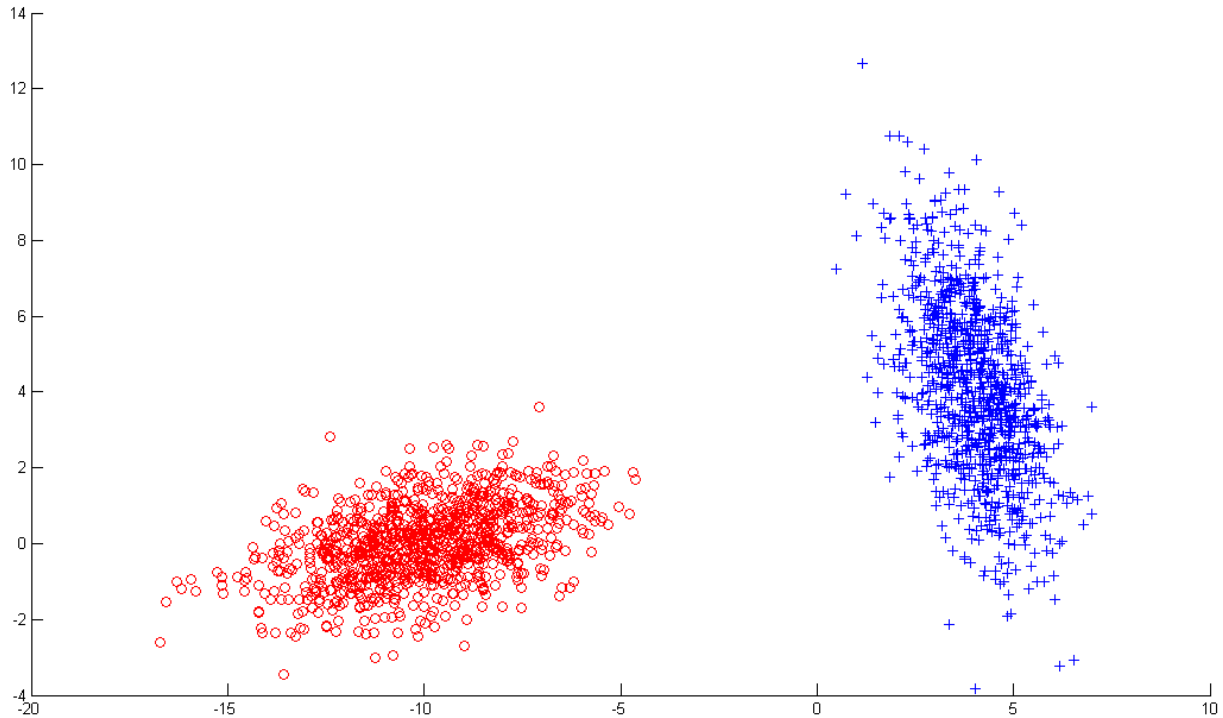


Figure: Two classes which are linearely seperable

Examples (2)

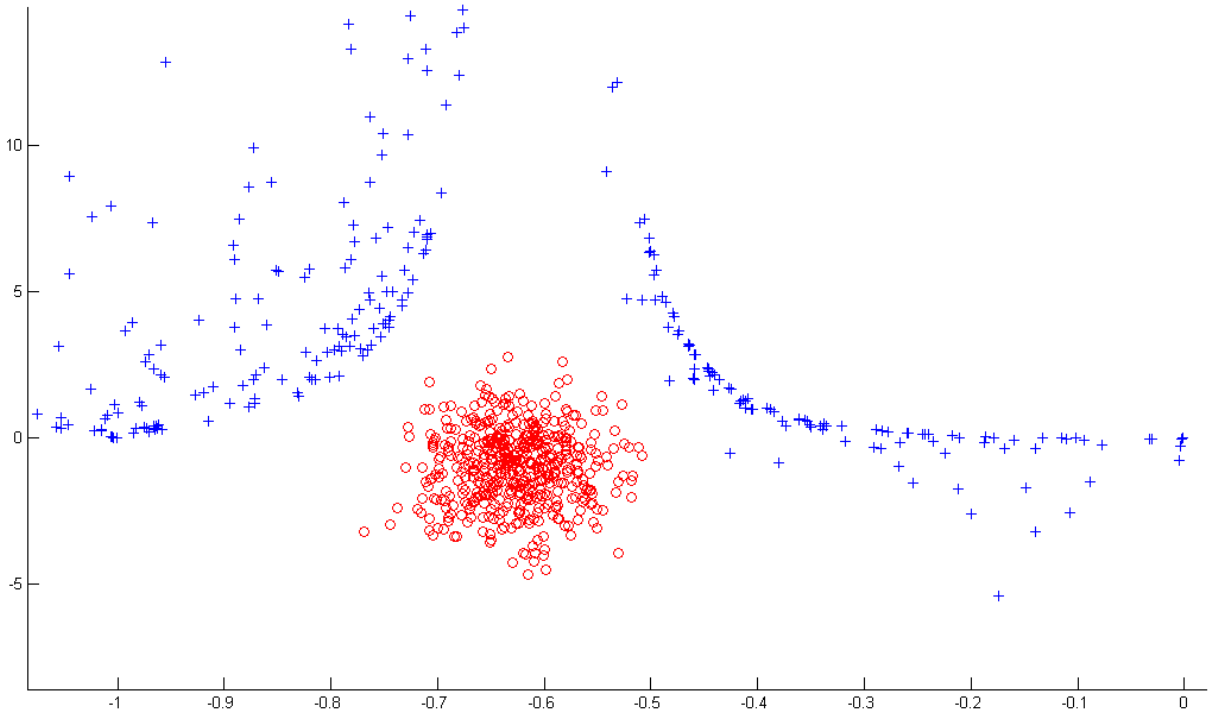


Figure: Two classes which are not linearely seperable

What it is...

- Common method in statistics / machine learning

What it is...

- Common method in statistics / machine learning
- Typically used for classification / dimensionality reduction

What it is...

- Common method in statistics / machine learning
- Typically used for classification / dimensionality reduction

Definition (binary case)

Given an N-dimensional feature space $\mathcal{X}$, examples $\mathbf{x}_1^i \in \mathcal{X}$ belonging to class 1 and examples $\mathbf{x}_2^j \in \mathcal{X}$ belonging to class 2, find a linear $(N-1)$-dimensional subspace $\hat{\mathcal{X}} \subseteq \mathcal{X}$ which best seperates classes 1 and 2.

The subspace $\hat{\mathcal{X}}$ is represented by its normal vector $\mathbf{w}^T$ ($\mathbf{w}^T\mathbf{x} = const.$).

Illustration

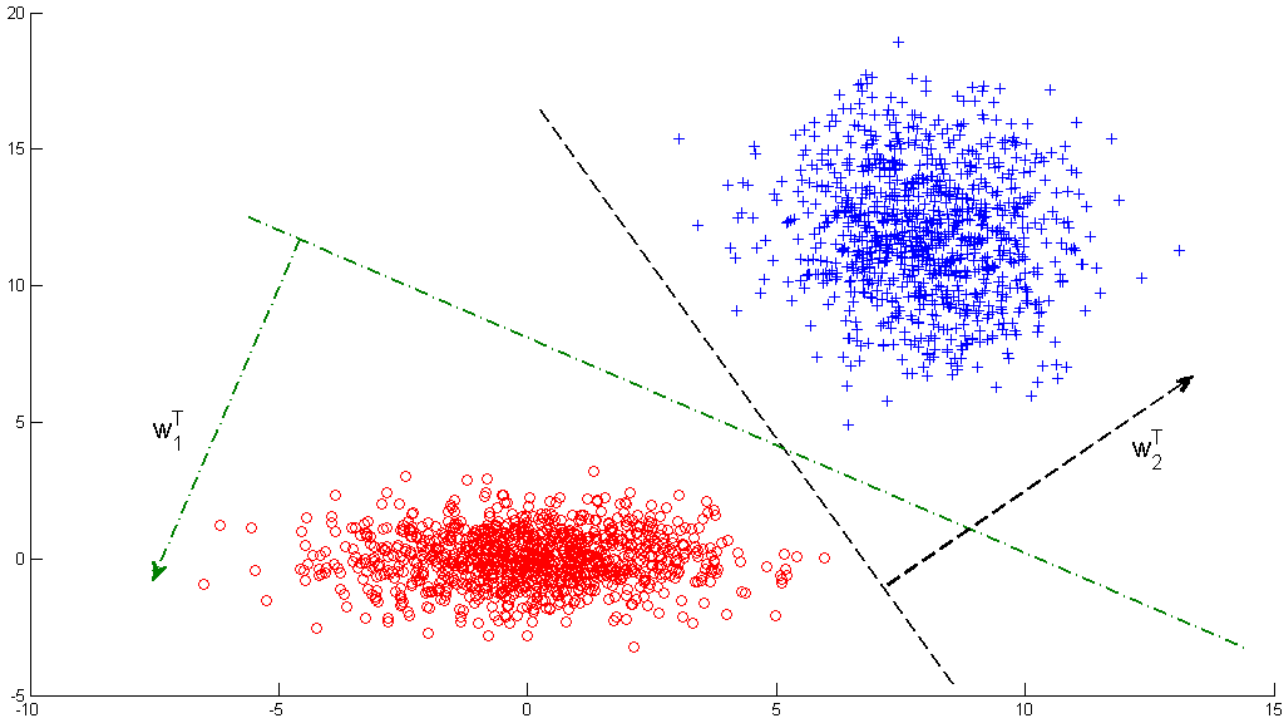


Figure: Many possibilities for seperation

Remarks

- Basically, LDA does not define how $\mathbf{w}^T$ is found

Remarks

- Basically, LDA does not define how $\mathbf{w}^T$ is found
- But typically LDA is associated with Fisher's Discriminant Analysis

Remarks

- Basically, LDA does not define how $\mathbf{w}^T$ is found
- But typically LDA is associated with Fisher's Discriminant Analysis
- In this presentation we will consider FDA

Fisher's Idea

- Definition of an optimality criterion which allows us to determine $\mathbf{w}^T$

Fisher's Idea

- Definition of an optimality criterion which allows us to determine $\mathbf{w}^T$

Fisher's Optimality Criterion

Choose a direction $\mathbf{w}^T$ that maximizes the *Within Class Variance* while minimizing the *Between Class Variance*.

Illustration of the FDA principle

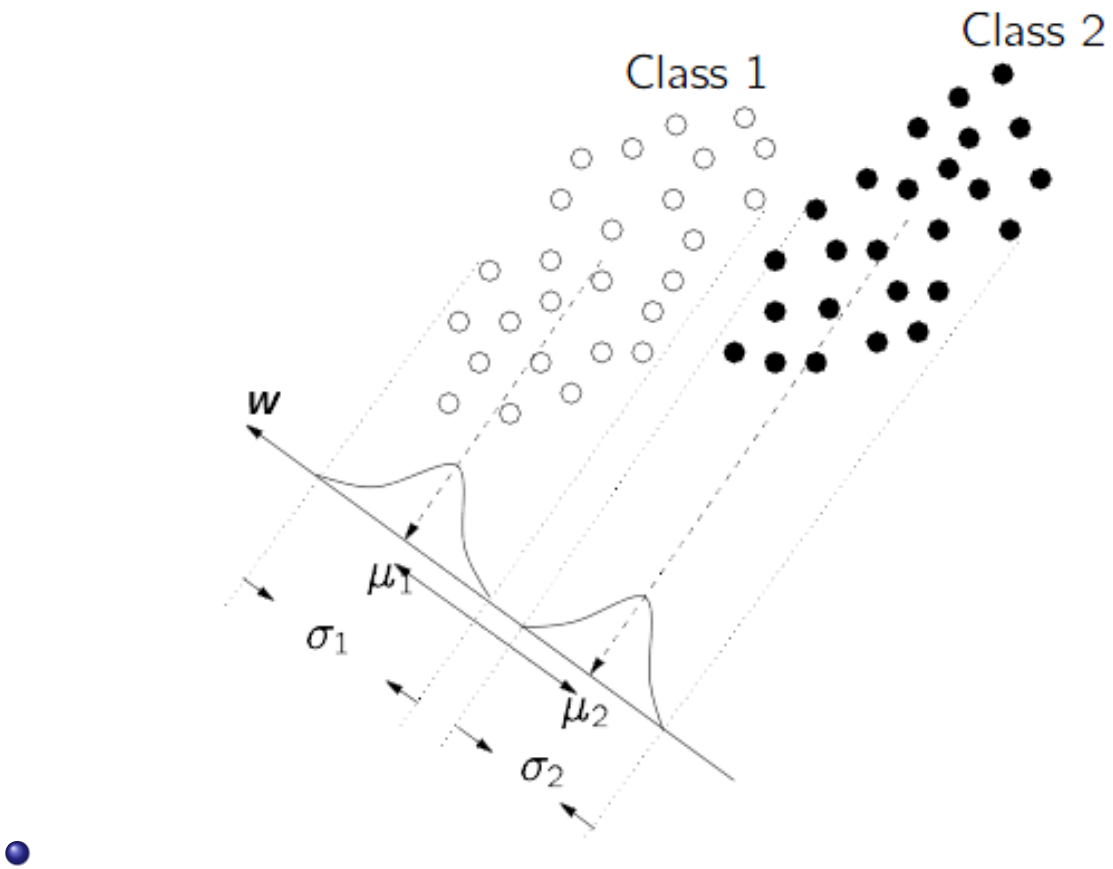


Figure: Source: [Mika, 2002]

Formulation

- We define:

Formulation

- We define:
  - Mean of Class 1: $\mathbf{m}_1 = E\{\mathbf{x}_1\}$
  - Mean of Class 2: $\mathbf{m}_2 = E\{\mathbf{x}_2\}$
  - Projected Mean of Class 1: $\mu_1 = \mathbf{w}^T\mathbf{m}_1$
  - Projected Mean of Class 2: $\mu_2 = \mathbf{w}^T\mathbf{m}_2$
  - Variance of projected Class 1: $\sigma_1^2 = E\left\{(\mathbf{w}^T\mathbf{x}_1 - \mu_1)^2\right\}$
  - Variance of projected Class 1: $\sigma_2^2 = E\left\{(\mathbf{w}^T\mathbf{x}_2 - \mu_1)^2\right\}$

Formulation

- We define:
  - Mean of Class 1: $\mathbf{m}_1 = E\{\mathbf{x}_1\}$
  - Mean of Class 2: $\mathbf{m}_2 = E\{\mathbf{x}_2\}$
  - Projected Mean of Class 1: $\mu_1 = \mathbf{w}^T\mathbf{m}_1$
  - Projected Mean of Class 2: $\mu_2 = \mathbf{w}^T\mathbf{m}_2$
  - Variance of projected Class 1: $\sigma_1^2 = E\{(\mathbf{w}^T\mathbf{x}_1 - \mu_1)^2\}$
  - Variance of projected Class 1: $\sigma_2^2 = E\{(\mathbf{w}^T\mathbf{x}_2 - \mu_1)^2\}$
- We consider the ratio of between class variance and within class variance

$$J(\mathbf{w}) = \frac{(\mu_1 - \mu_2)^2}{\sigma_1^2 + \sigma_2^2} \tag{2}$$

Formulation

- We define:
  - Mean of Class 1: $\mathbf{m}_1 = E\{\mathbf{x}_1\}$
  - Mean of Class 2: $\mathbf{m}_2 = E\{\mathbf{x}_2\}$
  - Projected Mean of Class 1: $\mu_1 = \mathbf{w}^T\mathbf{m}_1$
  - Projected Mean of Class 2: $\mu_2 = \mathbf{w}^T\mathbf{m}_2$
  - Variance of projected Class 1: $\sigma_1^2 = E\{(\mathbf{w}^T\mathbf{x}_1 - \mu_1)^2\}$
  - Variance of projected Class 1: $\sigma_2^2 = E\{(\mathbf{w}^T\mathbf{x}_2 - \mu_1)^2\}$
- We consider the ratio of between class variance and within class variance

$$J(\mathbf{w}) = \frac{(\mu_1 - \mu_2)^2}{\sigma_1^2 + \sigma_2^2} \tag{2}$$

- gives us the following optimization problem:

$$\mathbf{w} = \underset{\mathbf{w}\in\mathcal{X}}{arg\ max}\ J(\mathbf{w}) \tag{3}$$

Determining **w**

- It can be shown (blackboard) that

$$J(\mathbf{w}) = \frac{(\mu_1 - \mu_2)^2}{\sigma_1^2 + \sigma_2^2} = \frac{\mathbf{w}^T \mathbf{S}_b \mathbf{w}}{\mathbf{w}^T \mathbf{S}_w \mathbf{w}} \tag{4}$$

with

$$\mathbf{S}_b = (\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^T \tag{5}$$

and

$$\mathbf{S}_w = \mathbf{C}_{x_1 x_1} + \mathbf{C}_{x_2 x_2} \tag{6}$$

Determining **w**

- It can be shown (blackboard) that

$$J(\mathbf{w}) = \frac{(\mu_1 - \mu_2)^2}{\sigma_1^2 + \sigma_2^2} = \frac{\mathbf{w}^T \mathbf{S}_b \mathbf{w}}{\mathbf{w}^T \mathbf{S}_w \mathbf{w}} \tag{4}$$

  with

$$\mathbf{S}_b = (\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^T \tag{5}$$

  and

$$\mathbf{S}_w = \mathbf{C}_{x_1 x_1} + \mathbf{C}_{x_2 x_2} \tag{6}$$

- $\frac{\mathbf{w}^T \mathbf{S}_b \mathbf{w}}{\mathbf{w}^T \mathbf{S}_w \mathbf{w}}$ is known as the *rayleigh coefficient*,

Determining **w**

- It can be shown (blackboard) that

$$J(\mathbf{w}) = \frac{(\mu_1 - \mu_2)^2}{\sigma_1^2 + \sigma_2^2} = \frac{\mathbf{w}^T \mathbf{S}_b \mathbf{w}}{\mathbf{w}^T \mathbf{S}_w \mathbf{w}} \tag{4}$$

with

$$\mathbf{S}_b = (\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^T \tag{5}$$

and

$$\mathbf{S}_w = \mathbf{C}_{x_1 x_1} + \mathbf{C}_{x_2 x_2} \tag{6}$$

- $\frac{\mathbf{w}^T \mathbf{S}_b \mathbf{w}}{\mathbf{w}^T \mathbf{S}_w \mathbf{w}}$ is known as the *rayleigh coefficient*,
- which can be maximized in closed form with respect to **w**

Determining **w** (2)

Solution

Fisher's optimal direction can be determined as

$$\mathbf{w} = \mathbf{S}_w^{-1}(\mathbf{m}_1 - \mathbf{m}_2) \tag{7}$$

The discriminant function is then given by

$$\mathbf{w}^T\mathbf{x} = w_0 = \frac{\mathbf{w}^T(\mathbf{m}_1 + \mathbf{m}_2)}{2} \tag{8}$$

Classifying new Data Points

- Check if the data point $\mathbf{x}_{new}$ lies "above" or "below" the resulting hyperplane

$$H_{LDA} : \mathcal{X} \rightarrow \mathcal{Y} = \begin{cases} Class1 & \mathbf{w}^T \mathbf{x}_{new} < w_0 \\ Class2 & \mathbf{w}^T \mathbf{x}_{new} \geq w_0 \end{cases} \tag{9}$$

Classifying new Data Points

- Check if the data point $\mathbf{x}_{new}$ lies "above" or "below" the resulting hyperplane

$$H_{LDA} : \mathcal{X} \rightarrow \mathcal{Y} = \begin{cases} Class1 & \mathbf{w}^T \mathbf{x}_{new} < w_0 \\ Class2 & \mathbf{w}^T \mathbf{x}_{new} \geq w_0 \end{cases} \tag{9}$$

- Can also be interpreted as projecting the data point onto $\mathbf{w}$ and checking the decision threshhold w_0 in the reduced subspace (see MATLAB demo!)

Optimality

- In many cases, clusters follow a normal distribution

Optimality

- In many cases, clusters follow a normal distribution
- Knowing the gaussian class densities $p_1(\mathbf{x}) \sim \mathcal{N}(\mathbf{m}_1, \mathbf{\Sigma}_1)$ and $p_2(\mathbf{x}) \sim \mathcal{N}(\mathbf{m}_2, \mathbf{\Sigma}_2)$, an optimal decision boundary can be determined by setting

$$p_1(\mathbf{x}) \equiv p_2(\mathbf{x}). \tag{10}$$

Optimality

- In many cases, clusters follow a normal distribution
- Knowing the gaussian class densities $p_1(\mathbf{x}) \sim \mathcal{N}(\mathbf{m}_1, \mathbf{\Sigma}_1)$ and $p_2(\mathbf{x}) \sim \mathcal{N}(\mathbf{m}_2, \mathbf{\Sigma}_2)$, an optimal decision boundary can be determined by setting

$$p_1(\mathbf{x}) \equiv p_2(\mathbf{x}). \tag{10}$$

- It can be shown that if $\mathbf{\Sigma}_1 = \mathbf{\Sigma}_2 = \mathbf{\Sigma}$, the decision boundary takes a linear form:

$$\mathbf{w}^T \mathbf{x} = const. \tag{11}$$

Optimality

- In many cases, clusters follow a normal distribution
- Knowing the gaussian class densities $p_1(\mathbf{x}) \sim \mathcal{N}(\mathbf{m}_1, \mathbf{\Sigma}_1)$ and $p_2(\mathbf{x}) \sim \mathcal{N}(\mathbf{m}_2, \mathbf{\Sigma}_2)$, an optimal decision boundary can be determined by setting

$$p_1(\mathbf{x}) \equiv p_2(\mathbf{x}). \tag{10}$$

- It can be shown that if $\mathbf{\Sigma}_1 = \mathbf{\Sigma}_2 = \mathbf{\Sigma}$, the decision boundary takes a linear form:

$$\mathbf{w}^T \mathbf{x} = const. \tag{11}$$

- Does LDA produce the optimal gaussian decision boundary? (blackboard)

Optimality

- In many cases, clusters follow a normal distribution
- Knowing the gaussian class densities $p_1(\mathbf{x}) \sim \mathcal{N}(\mathbf{m}_1, \mathbf{\Sigma}_1)$ and $p_2(\mathbf{x}) \sim \mathcal{N}(\mathbf{m}_2, \mathbf{\Sigma}_2)$, an optimal decision boundary can be determined by setting

$$p_1(\mathbf{x}) \equiv p_2(\mathbf{x}). \tag{10}$$

- It can be shown that if $\mathbf{\Sigma}_1 = \mathbf{\Sigma}_2 = \mathbf{\Sigma}$, the decision boundary takes a linear form:

$$\mathbf{w}^T\mathbf{x} = const. \tag{11}$$

- Does LDA produce the optimal gaussian decision boundary? (blackboard)
- Yes it does!

- We get similar results for Bayesian models

- We get similar results for Bayesian models
- **BUT:** Class priors must be equal!

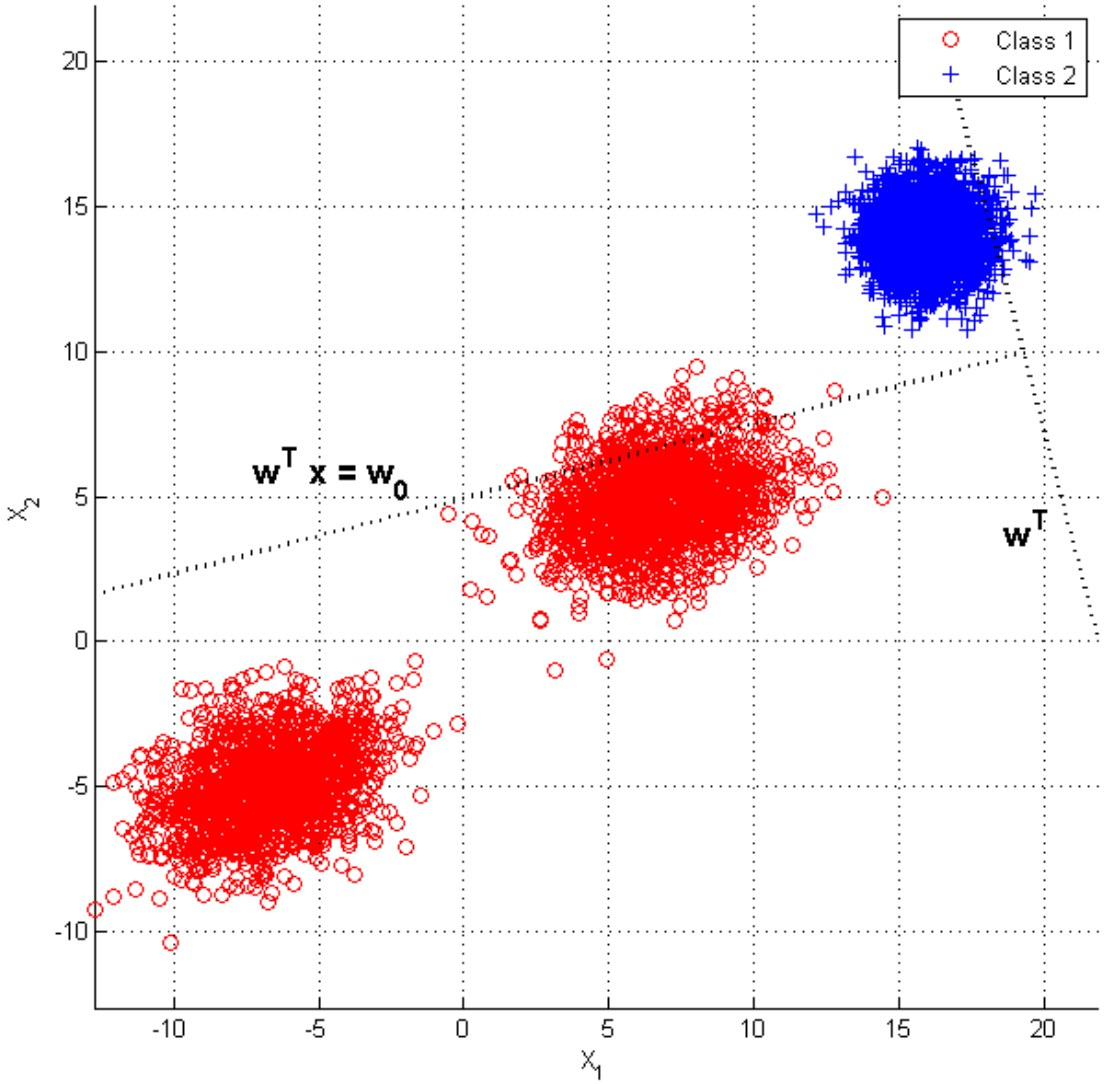
- In some cases LDA yields bad results

- In some cases LDA yields bad results
- Trivial: If data is not linearly separable

- In some cases LDA yields bad results
- Trivial: If data is not linearly separable
- Can be really bad, even if data is linearly separable

- In some cases LDA yields bad results
- Trivial: If data is not linearly separable
- Can be really bad, even if data is linearly separable
- Often if underlying densities are *complex* (e.g. multi-modal)

Example where LDA fails



Demo

MATLAB Demo...

Shortcomings of Fisher Discriminant

- Limited on linear discriminant functions...

Shortcomings of Fisher Discriminant

- Limited on linear discriminant functions...
- ...limited in complexity of problems which can be solved

Shortcomings of Fisher Discriminant

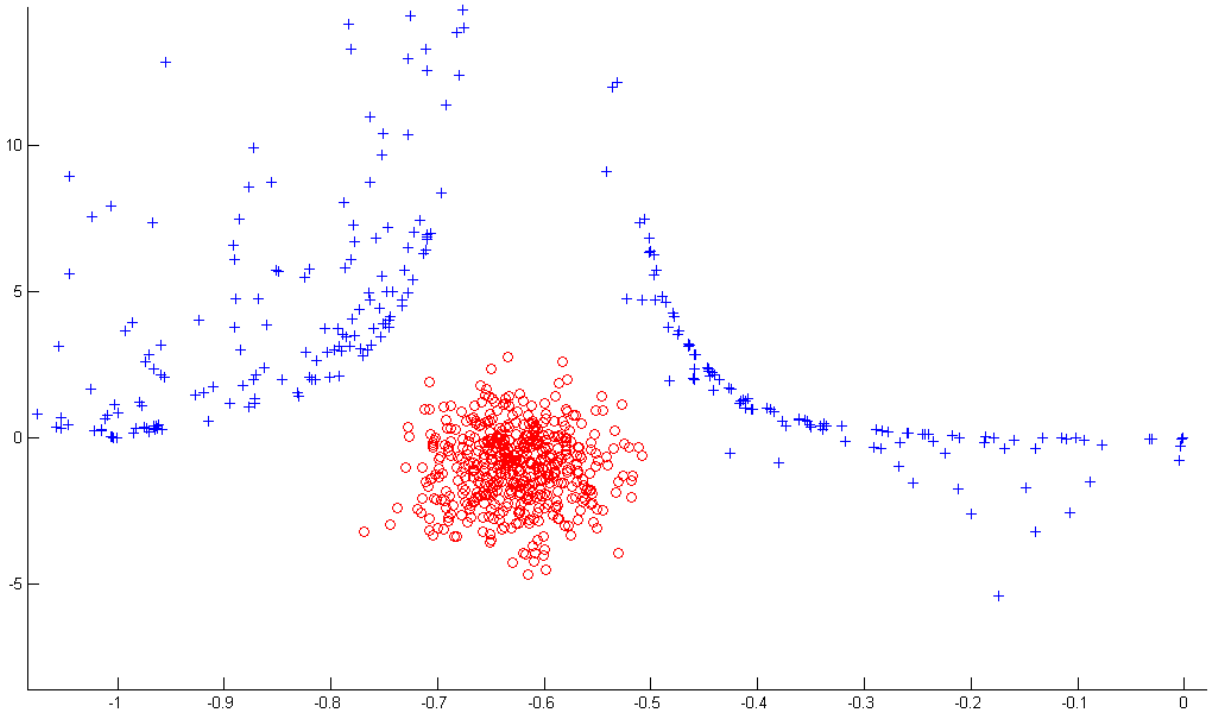


Figure: Two classes which are not linearly separable

Advantages of Kernels

- Proper solution to more complex, nonlinear problems

Advantages of Kernels

- Proper solution to more complex, nonlinear problems
- Linear discriminant functions in higher (even infinitely high) dimensional feature space...

Advantages of Kernels

- Proper solution to more complex, nonlinear problems
- Linear discriminant functions in higher (even infinitely high) dimensional feature space...
- ...complex, nonlinear discriminant function in input space

Advantages of Kernels

- Proper solution to more complex, nonlinear problems
- Linear discriminant functions in higher (even infinitely high) dimensional feature space...
- ...complex, nonlinear discriminant function in input space
- Kernel trick

Advantages of Kernels

- Proper solution to more complex, nonlinear problems
- Linear discriminant functions in higher (even infinitely high) dimensional feature space...
- ...complex, nonlinear discriminant function in input space
- Kernel trick
- Computational benefits

Advantages of Kernels

- Proper solution to more complex, nonlinear problems
- Linear discriminant functions in higher (even infinitely high) dimensional feature space...
- ...complex, nonlinear discriminant function in input space
- Kernel trick
- Computational benefits
- Kernels are not restricted to vectorial data - e.g. Strings ([Blaz, 2004], [Lodhi et al., 2002]) or Graphs ([Zhou et al., 2009])

Idea of how to use a kernel

- As linear discriminants are not enough for real world data...

Idea of how to use a kernel

- As linear discriminants are not enough for real world data...
- ...look for non-linear discriminants

Idea of how to use a kernel

- As linear discriminants are not enough for real world data...
- ...look for non-linear discriminants

- Map the data non-linearly to a feature space $\mathcal{F}$...

Idea of how to use a kernel

- As linear discriminants are not enough for real world data...
- ...look for non-linear discriminants

- Map the data non-linearly to a feature space $\mathcal{F}$...
- ...compute Fisher Discriminant there, yielding a non-linear discriminant in input space

Recap

- For the Fisher Discriminant we had to optimize:

$$J(\mathbf{w}) = \frac{(\mu_1 - \mu_2)^2}{\sigma_1^2 + \sigma_2^2} = \frac{\mathbf{w}^T \mathbf{S}_b \mathbf{w}}{\mathbf{w}^T \mathbf{S}_w \mathbf{w}} \tag{12}$$

with

$$\mathbf{S}_b = (\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^T \tag{13}$$

and

$$\mathbf{S}_w = \mathbf{C}_{x_1 x_1} + \mathbf{C}_{x_2 x_2} \tag{14}$$

and

$$\mathbf{m}_i = E\{\mathbf{x}_i\} \tag{15}$$

Formulation of the non-linear mapping Φ

- $J(\mathbf{w}) = \frac{\mathbf{w}^T \mathbf{S}_b^\Phi \mathbf{w}}{\mathbf{w}^T \mathbf{S}_w^\Phi \mathbf{w}}$

Formulation of the non-linear mapping Φ

- $J(\mathbf{w}) = \frac{\mathbf{w}^T \mathbf{S}_b^\Phi \mathbf{w}}{\mathbf{w}^T \mathbf{S}_w^\Phi \mathbf{w}}$
- with: $\mathbf{S}_b^\Phi = (\mathbf{m}_1^\Phi - \mathbf{m}_2^\Phi)(\mathbf{m}_1^\Phi - \mathbf{m}_2^\Phi)^T$

Formulation of the non-linear mapping Φ

- $J(\mathbf{w}) = \frac{\mathbf{w}^T \mathbf{S}_b^\Phi \mathbf{w}}{\mathbf{w}^T \mathbf{S}_w^\Phi \mathbf{w}}$
- with: $\mathbf{S}_b^\Phi = (\mathbf{m}_1^\Phi - \mathbf{m}_2^\Phi)(\mathbf{m}_1^\Phi - \mathbf{m}_2^\Phi)^T$
- $\mathbf{m}_i^\Phi = E\{\Phi(\mathbf{x}_i)\}$

Formulation of the non-linear mapping Φ

- $J(\mathbf{w}) = \frac{\mathbf{w}^T \mathbf{S}_b^\Phi \mathbf{w}}{\mathbf{w}^T \mathbf{S}_w^\Phi \mathbf{w}}$
- with: $\mathbf{S}_b^\Phi = (\mathbf{m}_1^\Phi - \mathbf{m}_2^\Phi)(\mathbf{m}_1^\Phi - \mathbf{m}_2^\Phi)^T$
- $\mathbf{m}_i^\Phi = E\{\Phi(\mathbf{x}_i)\}$
- $\mathbf{S}_w^\Phi = \mathbf{C}_{x_1 x_1}^\Phi + \mathbf{C}_{x_2 x_2}^\Phi$

Formulation of the non-linear mapping Φ

- $J(\mathbf{w}) = \frac{\mathbf{w}^T \mathbf{S}_b^\Phi \mathbf{w}}{\mathbf{w}^T \mathbf{S}_w^\Phi \mathbf{w}}$
- with: $\mathbf{S}_b^\Phi = (\mathbf{m}_1^\Phi - \mathbf{m}_2^\Phi)(\mathbf{m}_1^\Phi - \mathbf{m}_2^\Phi)^T$
- $\mathbf{m}_i^\Phi = E\{\Phi(\mathbf{x}_i)\}$
- $\mathbf{S}_w^\Phi = \mathbf{C}_{x_1 x_1}^\Phi + \mathbf{C}_{x_2 x_2}^\Phi$
- $\mathbf{C}_{x_i x_i}^\Phi = E\left\{(\Phi(\mathbf{x}_i) - \mathbf{m}_i^\Phi)(\Phi(\mathbf{x}_i) - \mathbf{m}_i^\Phi)^T\right\}$

Formulation of the non-linear mapping Φ

- $J(\mathbf{w}) = \frac{\mathbf{w}^T \mathbf{S}_b^\Phi \mathbf{w}}{\mathbf{w}^T \mathbf{S}_w^\Phi \mathbf{w}}$
- with: $\mathbf{S}_b^\Phi = (\mathbf{m}_1^\Phi - \mathbf{m}_2^\Phi)(\mathbf{m}_1^\Phi - \mathbf{m}_2^\Phi)^T$
- $\mathbf{m}_i^\Phi = E\{\Phi(\mathbf{x}_i)\}$
- $\mathbf{S}_w^\Phi = \mathbf{C}_{x_1 x_1}^\Phi + \mathbf{C}_{x_2 x_2}^\Phi$
- $\mathbf{C}_{x_i x_i}^\Phi = E\left\{(\Phi(\mathbf{x}_i) - \mathbf{m}_i^\Phi)(\Phi(\mathbf{x}_i) - \mathbf{m}_i^\Phi)^T\right\}$

- BUT NOW: $\mathbf{w} \in \mathcal{F}$

Kernel trick

- Kernel trick!!!

Kernel trick

- Kernel trick!!!
- Need a formulation of $\frac{\mathbf{w}^T \mathbf{S}_b^\Phi \mathbf{w}}{\mathbf{w}^T \mathbf{S}_w^\Phi \mathbf{w}}$ with only dot-products $\langle \Phi(\mathbf{x}), \Phi(\mathbf{y}) \rangle \rightarrow k(\mathbf{x}, \mathbf{y})$

Kernel trick

- Kernel trick!!!
- Need a formulation of $\frac{\mathbf{w}^T \mathbf{S}_b^\Phi \mathbf{w}}{\mathbf{w}^T \mathbf{S}_w^\Phi \mathbf{w}}$ with only dot-products $\langle \Phi(\mathbf{x}), \Phi(\mathbf{y}) \rangle \rightarrow k(\mathbf{x}, \mathbf{y})$
- Then we can use different kernels (RBF, Polynomial, ...) to do the computation efficiently, without explicitly map to the feature space $\mathcal{F}$

Kernel trick

- Kernel trick!!!
- Need a formulation of $\frac{\mathbf{w}^T \mathbf{S}_b^\Phi \mathbf{w}}{\mathbf{w}^T \mathbf{S}_w^\Phi \mathbf{w}}$ with only dot-products $\langle \Phi(\mathbf{x}), \Phi(\mathbf{y}) \rangle \rightarrow k(\mathbf{x}, \mathbf{y})$
- Then we can use different kernels (RBF, Polynomial, ...) to do the computation efficiently, without explicitly map to the feature space $\mathcal{F}$
- As numerator and denominator are both scalars, reformulation can be done independently

Reformulation of numerator

- Problem: $J(\mathbf{w}) = \frac{\mathbf{w}^T \mathbf{S}_b^\Phi \mathbf{w}}{\mathbf{w}^T \mathbf{S}_w^\Phi \mathbf{w}}$

Reformulation of numerator

- Problem: $J(\mathbf{w}) = \frac{\mathbf{w}^T \mathbf{S}_b^\Phi \mathbf{w}}{\mathbf{w}^T \mathbf{S}_w^\Phi \mathbf{w}}$
- Numerator: $\mathbf{w}^T \mathbf{S}_b^\Phi \mathbf{w} = \mathbf{w}^T (\mathbf{m}_1^\Phi - \mathbf{m}_2^\Phi)(\mathbf{m}_1^\Phi - \mathbf{m}_2^\Phi)^T \mathbf{w}$

Reformulation of numerator

- Problem: $J(\mathbf{w}) = \frac{\mathbf{w}^T \mathbf{S}_b^\Phi \mathbf{w}}{\mathbf{w}^T \mathbf{S}_w^\Phi \mathbf{w}}$
- Numerator: $\mathbf{w}^T \mathbf{S}_b^\Phi \mathbf{w} = \mathbf{w}^T (\mathbf{m}_1^\Phi - \mathbf{m}_2^\Phi)(\mathbf{m}_1^\Phi - \mathbf{m}_2^\Phi)^T \mathbf{w}$
- Theory of RKHS: $\mathbf{w} = \sum_{\mathbf{x} \in \mathcal{Z}}^M \alpha_x \Phi(\mathbf{x})$

Reformulation of numerator

- Problem: $J(\mathbf{w}) = \frac{\mathbf{w}^T \mathbf{S}_b^\Phi \mathbf{w}}{\mathbf{w}^T \mathbf{S}_w^\Phi \mathbf{w}}$
- Numerator: $\mathbf{w}^T \mathbf{S}_b^\Phi \mathbf{w} = \mathbf{w}^T (\mathbf{m}_1^\Phi - \mathbf{m}_2^\Phi)(\mathbf{m}_1^\Phi - \mathbf{m}_2^\Phi)^T \mathbf{w}$
- Theory of RKHS: $\mathbf{w} = \sum_{\mathbf{x} \in \mathcal{Z}}^{M} \alpha_x \Phi(\mathbf{x})$
- $\mathbf{m}_i^\Phi = E\{\Phi(\mathbf{x}_i)\}$

Reformulation of numerator

- Problem: $J(\mathbf{w}) = \frac{\mathbf{w}^T \mathbf{S}_b^\Phi \mathbf{w}}{\mathbf{w}^T \mathbf{S}_w^\Phi \mathbf{w}}$
- Numerator: $\mathbf{w}^T \mathbf{S}_b^\Phi \mathbf{w} = \mathbf{w}^T (\mathbf{m}_1^\Phi - \mathbf{m}_2^\Phi)(\mathbf{m}_1^\Phi - \mathbf{m}_2^\Phi)^T \mathbf{w}$
- Theory of RKHS: $\mathbf{w} = \sum_{\mathbf{x} \in \mathcal{Z}}^{M} \alpha_x \Phi(\mathbf{x})$
- $\mathbf{m}_i^\Phi = E\{\Phi(\mathbf{x}_i)\}$
- **Solution:** $\alpha^T \mathbf{M} \alpha$

Reformulation of numerator

- Problem: $J(\mathbf{w}) = \frac{\mathbf{w}^T \mathbf{S}_b^\Phi \mathbf{w}}{\mathbf{w}^T \mathbf{S}_w^\Phi \mathbf{w}}$
- Numerator: $\mathbf{w}^T \mathbf{S}_b^\Phi \mathbf{w} = \mathbf{w}^T (\mathbf{m}_1^\Phi - \mathbf{m}_2^\Phi)(\mathbf{m}_1^\Phi - \mathbf{m}_2^\Phi)^T \mathbf{w}$
- Theory of RKHS: $\mathbf{w} = \sum_{\mathbf{x} \in \mathcal{Z}}^{M} \alpha_x \Phi(\mathbf{x})$
- $\mathbf{m}_i^\Phi = E\{\Phi(\mathbf{x}_i)\}$
- **Solution:** $\alpha^T \mathbf{M} \alpha$
- where $\mathbf{M} = (\mathbf{K}_{\bar{x}_1} - \mathbf{K}_{\bar{x}_2})(\mathbf{K}_{\bar{x}_1} - \mathbf{K}_{\bar{x}_2})^T$

Reformulation of denominator

- Problem: $J(\mathbf{w}) = \frac{\mathbf{w}^T \mathbf{S}_b^\Phi \mathbf{w}}{\mathbf{w}^T \mathbf{S}_w^\Phi \mathbf{w}}$

Reformulation of denominator

- Problem: $J(\mathbf{w}) = \frac{\mathbf{w}^T \mathbf{S}_b^\Phi \mathbf{w}}{\mathbf{w}^T \mathbf{S}_w^\Phi \mathbf{w}}$
- Denominator: $\mathbf{w}^T \mathbf{S}_w^\Phi \mathbf{w} = \mathbf{C}_{x_1 x_1}^\Phi + \mathbf{C}_{x_2 x_2}^\Phi$

Reformulation of denominator

- Problem: $J(\mathbf{w}) = \frac{\mathbf{w}^T \mathbf{S}_b^\Phi \mathbf{w}}{\mathbf{w}^T \mathbf{S}_w^\Phi \mathbf{w}}$
- Denominator: $\mathbf{w}^T \mathbf{S}_w^\Phi \mathbf{w} = \mathbf{C}_{x_1 x_1}^\Phi + \mathbf{C}_{x_2 x_2}^\Phi$
- Theory of RKHS: $\mathbf{w} = \sum_{\mathbf{x} \in \mathcal{Z}}^M \alpha_x \Phi(\mathbf{x})$

Reformulation of denominator

- Problem: $J(\mathbf{w}) = \frac{\mathbf{w}^T \mathbf{S}_b^\Phi \mathbf{w}}{\mathbf{w}^T \mathbf{S}_w^\Phi \mathbf{w}}$
- Denominator: $\mathbf{w}^T \mathbf{S}_w^\Phi \mathbf{w} = \mathbf{C}_{x_1 x_1}^\Phi + \mathbf{C}_{x_2 x_2}^\Phi$
- Theory of RKHS: $\mathbf{w} = \sum_{\mathbf{x} \in \mathcal{Z}}^M \alpha_x \Phi(\mathbf{x})$
- $\mathbf{m}_i^\Phi = E\{\Phi(\mathbf{x}_i)\}$

Reformulation of denominator

- Problem: $J(\mathbf{w}) = \frac{\mathbf{w}^T \mathbf{S}_b^\Phi \mathbf{w}}{\mathbf{w}^T \mathbf{S}_w^\Phi \mathbf{w}}$
- Denominator: $\mathbf{w}^T \mathbf{S}_w^\Phi \mathbf{w} = \mathbf{C}_{x_1 x_1}^\Phi + \mathbf{C}_{x_2 x_2}^\Phi$
- Theory of RKHS: $\mathbf{w} = \sum_{\mathbf{x} \in \mathcal{Z}}^M \alpha_x \Phi(\mathbf{x})$
- $\mathbf{m}_i^\Phi = E\{\Phi(\mathbf{x}_i)\}$
- **Solution:** $\alpha^T \mathbf{N} \alpha$

Reformulation of denominator

- Problem: $J(\mathbf{w}) = \frac{\mathbf{w}^T \mathbf{S}_b^\Phi \mathbf{w}}{\mathbf{w}^T \mathbf{S}_w^\Phi \mathbf{w}}$
- Denominator: $\mathbf{w}^T \mathbf{S}_w^\Phi \mathbf{w} = \mathbf{C}_{x_1 x_1}^\Phi + \mathbf{C}_{x_2 x_2}^\Phi$
- Theory of RKHS: $\mathbf{w} = \sum_{\mathbf{x} \in \mathcal{Z}}^M \alpha_x \Phi(\mathbf{x})$
- $\mathbf{m}_i^\Phi = E\{\Phi(\mathbf{x}_i)\}$
- **Solution:** $\alpha^T \mathbf{N} \alpha$
- where
  $\mathbf{N} = E\left\{(\mathbf{K}_{x_1} - \mathbf{K}_{\bar{x}_1})(\mathbf{K}_{x_1} - \mathbf{K}_{\bar{x}_1})^T\right\} + E\left\{(\mathbf{K}_{x_2} - \mathbf{K}_{\bar{x}_2})(\mathbf{K}_{x_2} - \mathbf{K}_{\bar{x}_2})^T\right\}$

Kernel Fisher Discriminant Solution

- Solving the Kernel Fisher Discriminant by:

Kernel Fisher Discriminant Solution

- Solving the Kernel Fisher Discriminant by:

Optimization Problem

$$J(\alpha) = \frac{\alpha^T M \alpha}{\alpha^T N \alpha} \tag{16}$$

Kernel Fisher Discriminant Solution

- Solving the Kernel Fisher Discriminant by:

Optimization Problem

$$J(\alpha) = \frac{\alpha^T M \alpha}{\alpha^T N \alpha} \tag{16}$$

- solve like the Fisher Discriminant, Eigenvector problem

Kernel Fisher Discriminant Solution

- Solving the Kernel Fisher Discriminant by:

Optimization Problem

$$J(\alpha) = \frac{\alpha^T M \alpha}{\alpha^T N \alpha} \tag{16}$$

- solve like the Fisher Discriminant, Eigenvector problem
- Projection of new example $\mathbf{x}_{new}$ onto $\mathbf{w}$:

Kernel Fisher Discriminant Solution

- Solving the Kernel Fisher Discriminant by:

Optimization Problem

$$J(\alpha) = \frac{\alpha^T M \alpha}{\alpha^T N \alpha} \tag{16}$$

- solve like the Fisher Discriminant, Eigenvector problem
- Projection of new example $\mathbf{x}_{new}$ onto $\mathbf{w}$:

Projection of new example $\mathbf{x}_{new}$ onto $\mathbf{w}$

$$\mathbf{w}^T \Phi(\mathbf{x}_{new}) = \sum_{\mathbf{x} \in \mathcal{Z}}^{M} \alpha_x k(\mathbf{x}, \mathbf{x}_{new}) \tag{17}$$

Some useful facts

- Regularization ([Mika, 2002])

Some useful facts

- Regularization ([Mika, 2002])
- Choice of an optimal kernel ([Kim et al., 2006])

Some useful facts

- Regularization ([Mika, 2002])
- Choice of an optimal kernel ([Kim et al., 2006])
- Computational efficiency - Kernel Matrix

Some useful facts

- Regularization ([Mika, 2002])
- Choice of an optimal kernel ([Kim et al., 2006])
- Computational efficiency - Kernel Matrix
- Approximation algorithms ([Mika, 2002])

Example to KFD from ([Knaf, 2007])

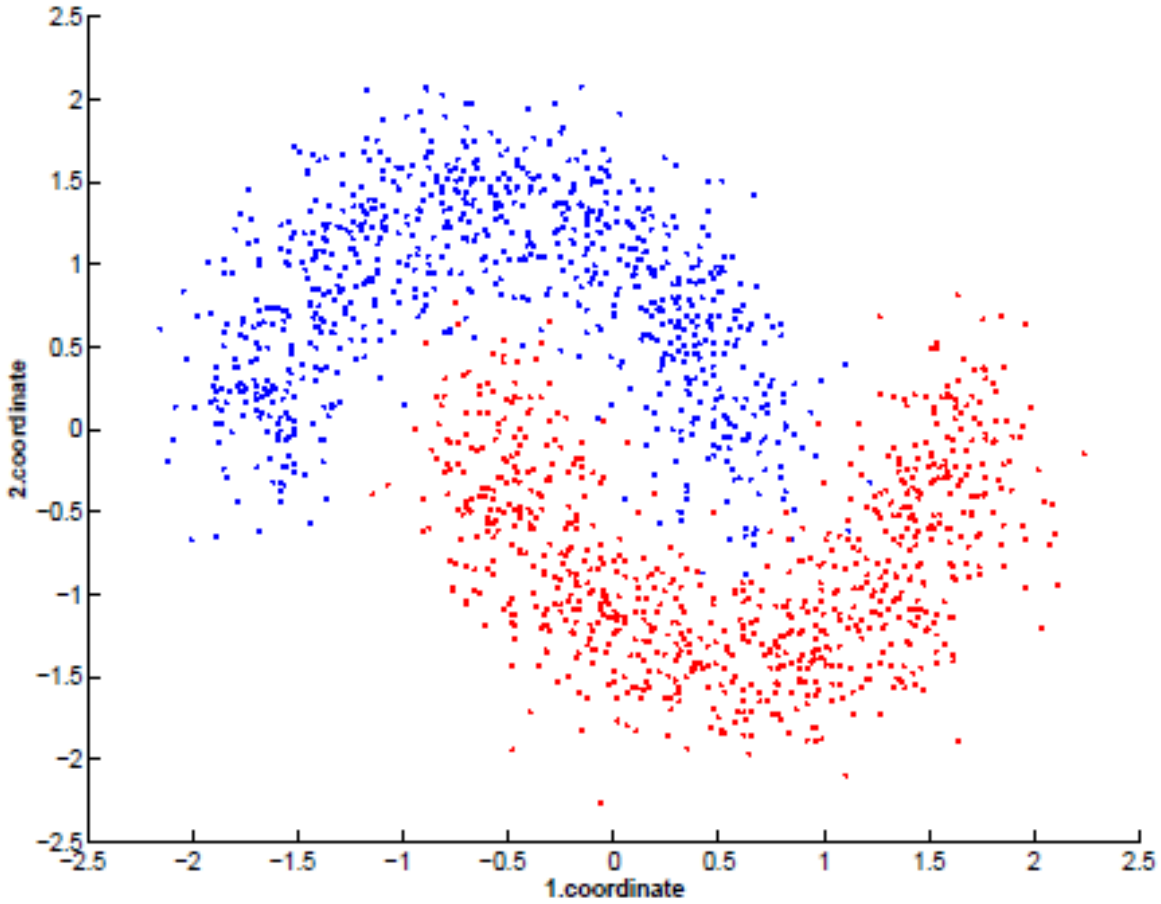


Figure: Input data $\mathcal{X} = \mathcal{X}_1 \cup \mathcal{X}_2$, Source: [Knaf, 2007]

Approach

- Separate the data by:

Approach

- Separate the data by:
- Kernel Fisher Discriminant with different Kernels

Approach

- Separate the data by:
- Kernel Fisher Discriminant with different Kernels
- Gaussian Kernel: $K(\mathbf{x}_1, \mathbf{x}_2) = e^{-\frac{\|\mathbf{x}_1 - \mathbf{x}_2\|^2}{h^2}}$

Approach

- Separate the data by:
- Kernel Fisher Discriminant with different Kernels
- Gaussian Kernel: $K(\mathbf{x}_1, \mathbf{x}_2) = e^{-\frac{\|\mathbf{x}_1 - \mathbf{x}_2\|^2}{h^2}}$
- Polynomial Kernel: $K(\mathbf{x}_1, \mathbf{x}_2) = (\langle \mathbf{x}_1, \mathbf{x}_2 \rangle + c)^3$

Approach

- Separate the data by:
- Kernel Fisher Discriminant with different Kernels
- Gaussian Kernel: $K(\mathbf{x}_1, \mathbf{x}_2) = e^{-\frac{\|\mathbf{x}_1 - \mathbf{x}_2\|^2}{h^2}}$
- Polynomial Kernel: $K(\mathbf{x}_1, \mathbf{x}_2) = (\langle \mathbf{x}_1, \mathbf{x}_2 \rangle + c)^3$
- Comparison to simple Fisher Discriminant

Discriminant function of the KFD with Gauss-Kernel

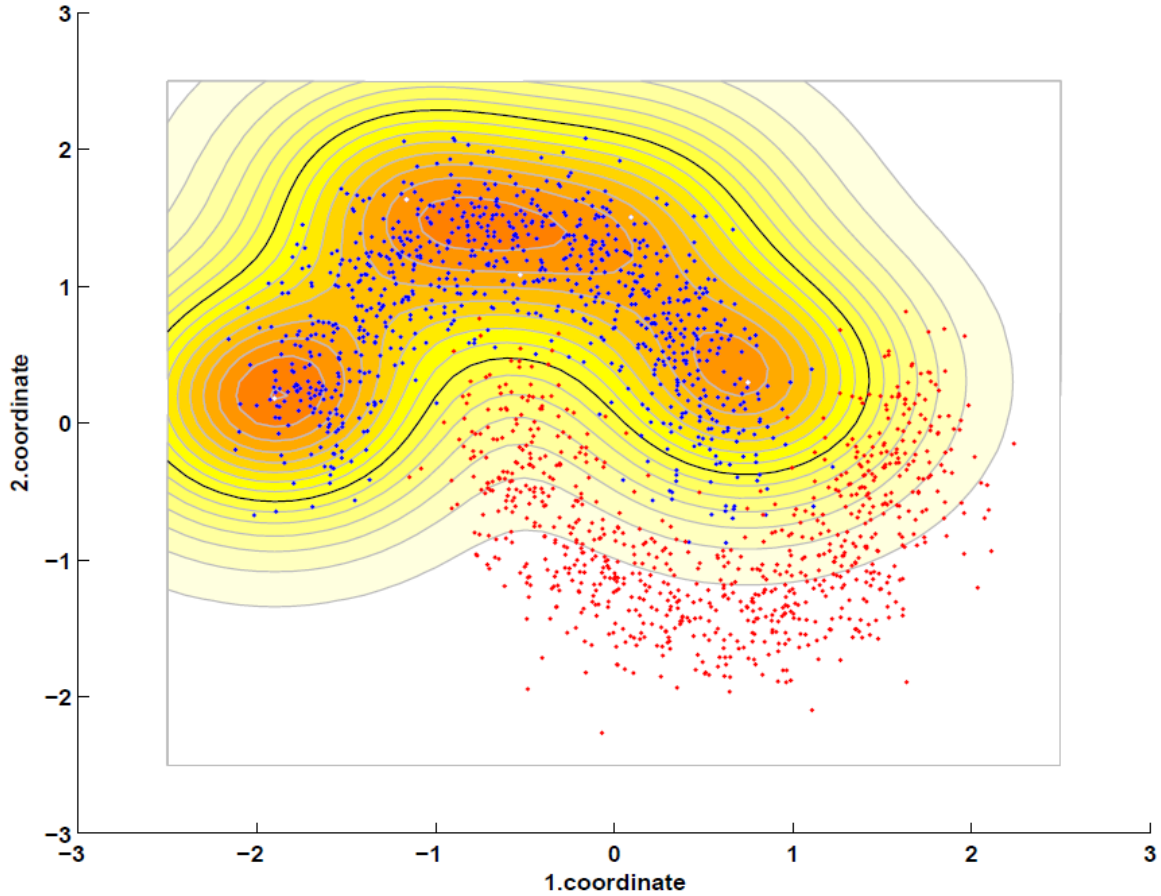


Figure: Source: [Knaf, 2007]

Discriminant function of the KFD with Polynomial-Kernel

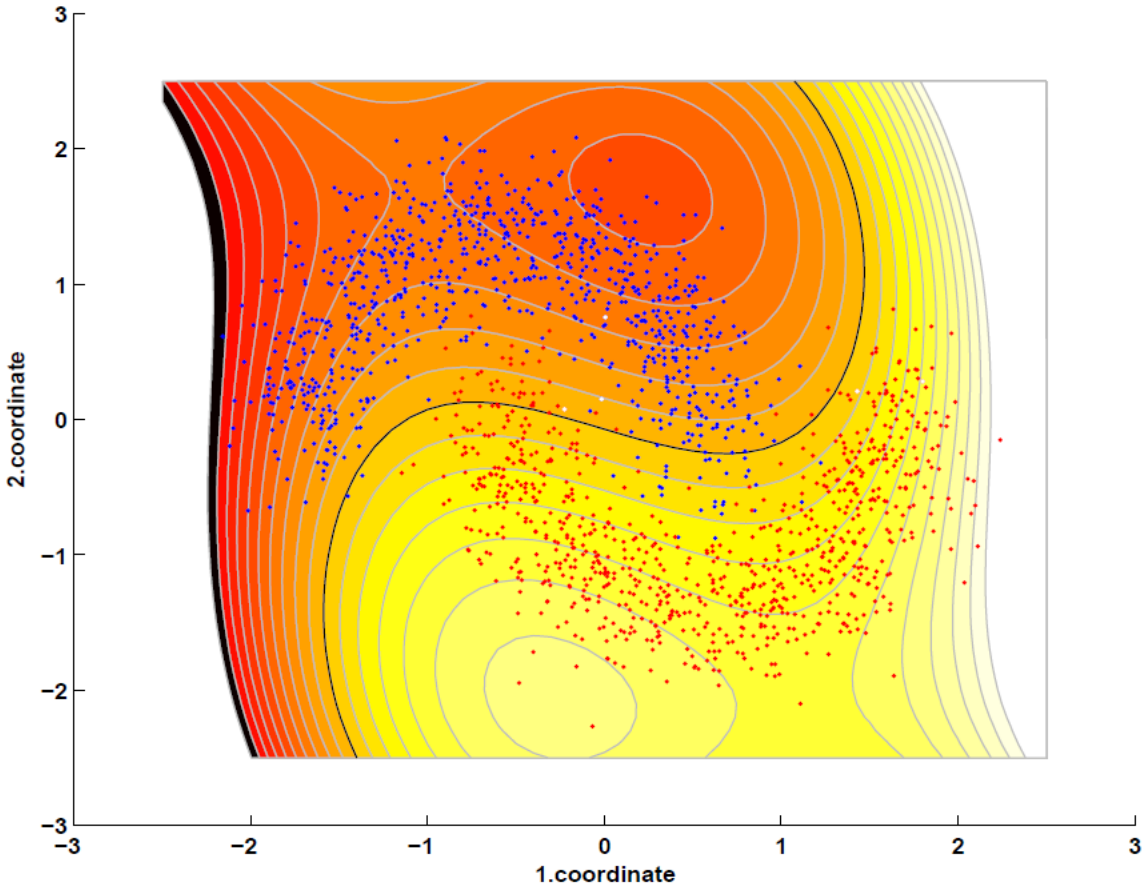


Figure: Source: [Knaf, 2007]

Distributions of the data points projected on the discriminant functions

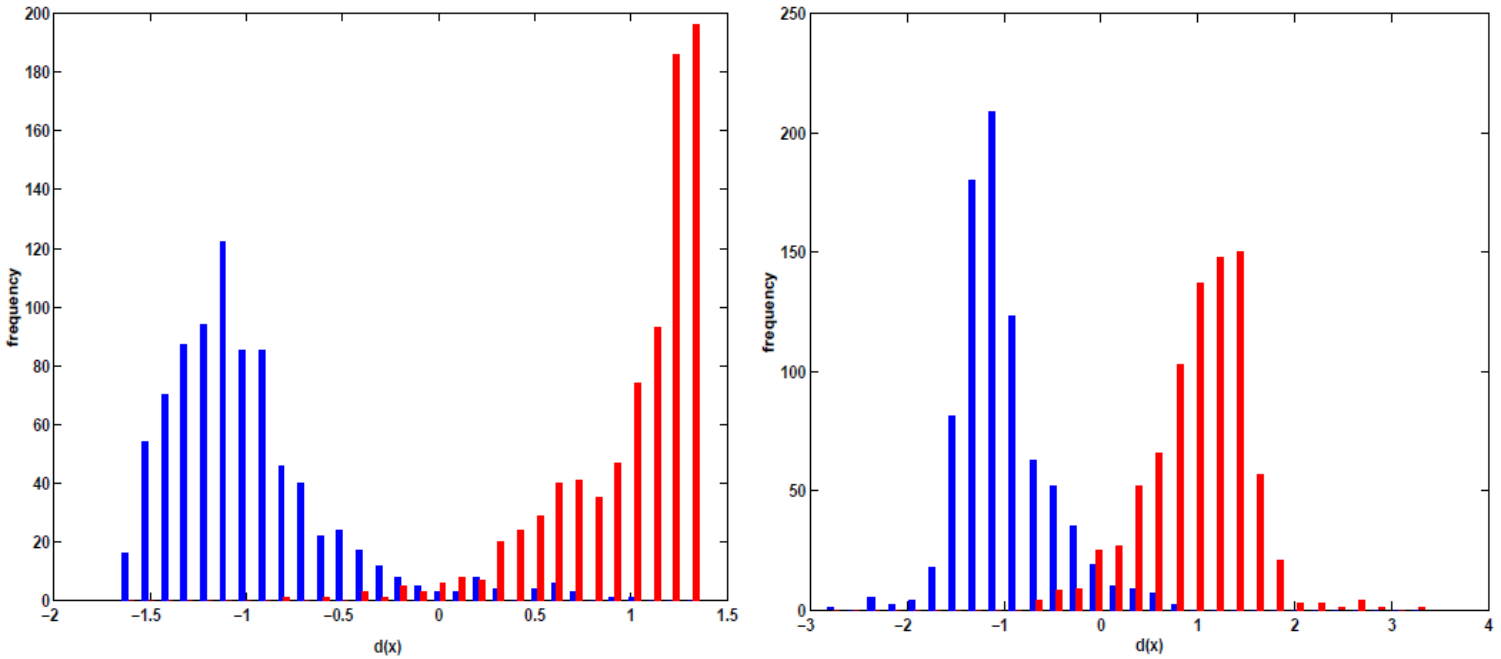


Figure: Source: [Knaf, 2007]

Solution of the simple Fisher discriminant function

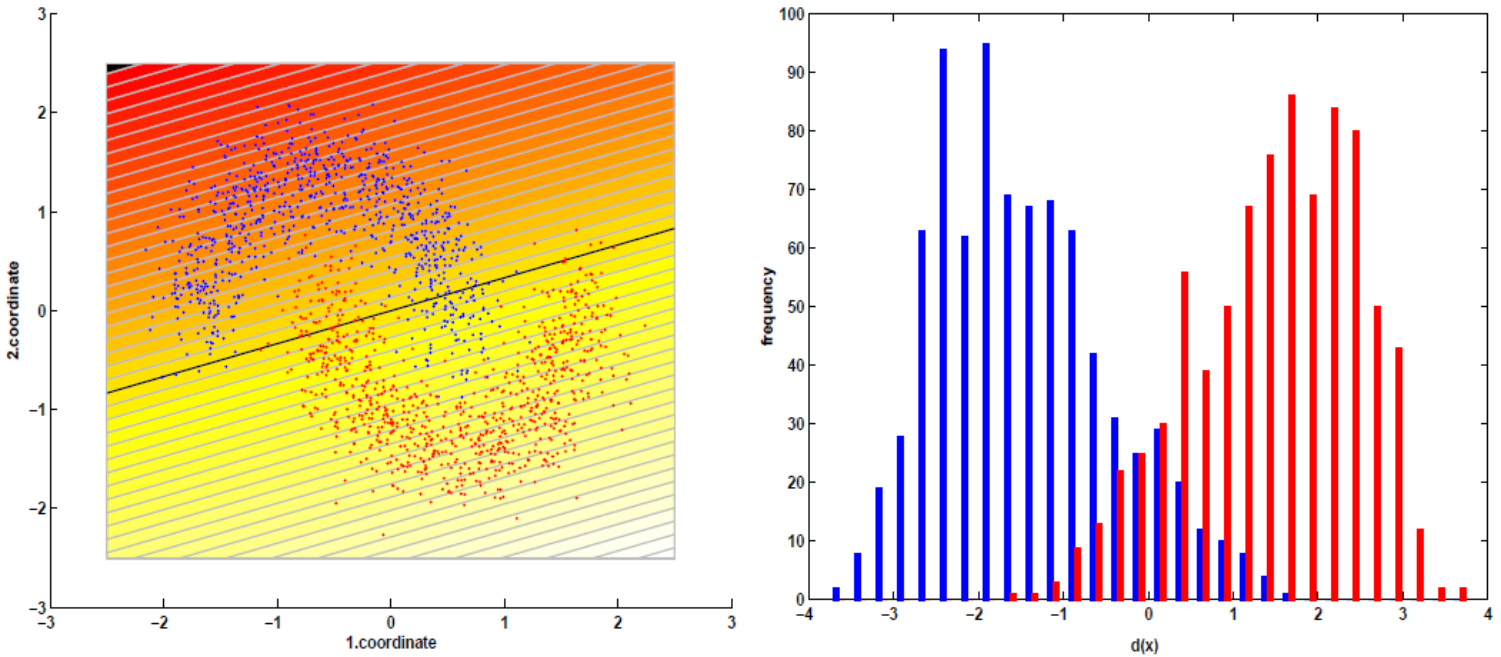


Figure: Source:[Knaf, 2007]

Demo

MATLAB Demo...

Summary

- Supervised learning

Summary

- Supervised learning
- Linear Discriminant

Summary

- Supervised learning
- Linear Discriminant
- Fisher Discriminant

Summary

- Supervised learning
- Linear Discriminant
- Fisher Discriminant
- Kernel Fisher Discriminant

Blaz, F. (2004).
String kernels.

Kim, S.-J., Magnani, A., and Boyd, S. (2006).
Optimal kernel selection in kernel fisher discriminant analysis.
Proceedings of the 23rd International Conference on Machine Learning (ICML), pages 465–472.

Knaf, H. (2007).
Kernel fisher discriminant functions - a concise and rigorous introduction.
Berichte des Fraunhofer ITWM 117, Fraunhofer Institut, Techno- und Wirtschaftsmathematik.

Lodhi, H., Saunders, C., Shawe-Taylor, J., Cristianini, N., and Watkins, C. (2002).
Text classification using string kernels.
Journal of Machine Learning Resarch 2, pages 419–444.

Mika, S. (2002).
Kernel Fisher Discriminants.

PhD thesis, University of technology, Berlin.

Zhou, Z.-H., Sun, Y.-Y., and Li, Y.-F. (2009).
Multi-instance learning by treating instances as non-.i.i.d samples.