Linear Models for Classification

TTT4185 Machine Learning for Signal Processing

Giampiero Salvi

Department of Electronic Systems
NTNU

HT2021

Outline

1. Discriminant Functions
2. Learning Parameters
   - Least-squares
   - Fisher's discriminant
   - Perceptron
3. Probabilistic Models
   - Generative Approach
   - Discriminative Approach

Outline

Discriminant Functions

Classification

$$\mathbf{x} \in \mathbb{R}^M, \quad t \in \{\mathcal{C}_1, \dots, \mathcal{C}_K\}$$

Then

$$y(\mathbf{x}) : \mathbb{R}^M \to \{\mathcal{C}_1, \dots, \mathcal{C}_K\}$$

Decision regions $\mathcal{R}_1, \dots, \mathcal{R}_K \in \mathbb{R}^M$

Decision Boundaries:

- linear models: $\mathbf{x} \in \mathbb{R}^{M-1}$ ($M-1$ hyperplanes)

Discriminant Functions

Linear in parameters and inputs

$$y(\mathbf{x}, \mathbf{w}) = f(\mathbf{w}^T x + w_0)$$

$f(.)$ is called

- activation function (machine learning)
- inverse link function (statistics)

Decision surfaces:

$$y(\mathbf{x}, \mathbf{w}) = \text{constant} \Leftrightarrow \mathbf{w}^T \mathbf{x} + w_0 = \text{constant}$$

Even if $f(.)$ is non-linear.

Example: Two Classes

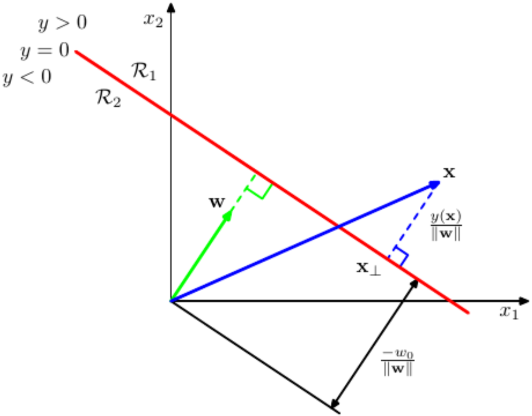
$$y(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + w_0$$
$$y(\mathbf{x}) \geq 0 \rightarrow \mathbf{x} \in \mathcal{R}_1 \quad (\mathcal{C}_1)$$
$$y(\mathbf{x}) < 0 \rightarrow \mathbf{x} \in \mathcal{R}_2 \quad (\mathcal{C}_2)$$

(To be precise: $y(\mathbf{x}) = \mathrm{sign}(\mathbf{w}^T\mathbf{x} + w_0)$)

Equal values perpendicular to $\mathbf{w}$:

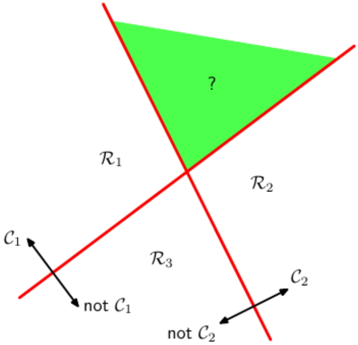
$$\begin{aligned} y(\mathbf{x}_A) = y(\mathbf{x}_B) \Rightarrow &\quad \mathbf{w}^T(\mathbf{x}_A - \mathbf{x}_B) = 0 \\ \Leftrightarrow &\quad \mathbf{w} \perp \text{boundary} \end{aligned}$$

Decision boundary for $\mathbf{w}^T\mathbf{x} = -w_0$

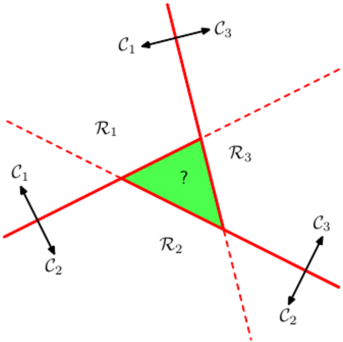


Multiple Classes

Problem combining 2-class decision functions:



one-versus-the-rest



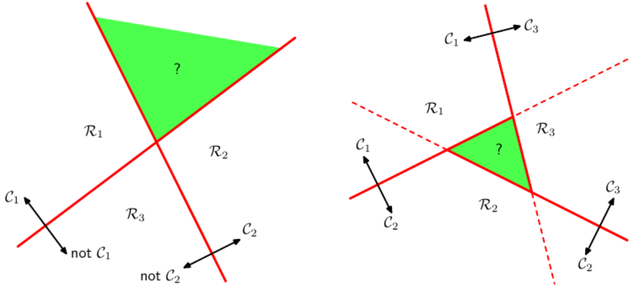
one-versus-one

k-class discriminant

$$y_k(\mathbf{x}) = \mathbf{w}_k^T \mathbf{x} + w_{k0}$$
$$\mathbf{x} \in C_k \Leftrightarrow y_k(\mathbf{x}) > y_j(\mathbf{x}) \forall j \neq k$$

Decision boundaries:

$$y_k(\mathbf{x}) = y_j(\mathbf{x}) \Rightarrow$$
$$(\mathbf{w}_k - \mathbf{w}_j)^T(\mathbf{x}) + (w_{k0} - w_{j0}) = 0$$



Simplified Notation and Basis Functions

Simplified notation

$$y(\mathbf{x}, \mathbf{w}) = \mathbf{w}^T \mathbf{x}$$

where

$$\mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_M \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} 1 \\ x_1 \\ \vdots \\ x_M \end{bmatrix}$$

All results are equivalent if we use basis functions:

$$y(\mathbf{x}, \mathbf{w}) = \mathbf{w}^T \phi(\mathbf{x})$$

Outline

1. Discriminant Functions
2. Learning Parameters
   - Least-squares
   - Fisher's discriminant
   - Perceptron
3. Probabilistic Models
   - Generative Approach
   - Discriminative Approach

Least Squares

Data $\mathcal{D} = \{(\mathbf{x}_1, t_1), \dots, (\mathbf{x}_N, t_n)\}$

Minimize:

$$E_{\mathcal{D}} = \frac{1}{2} \sum_{n=1}^{N} (y(\mathbf{x}_n, \mathbf{w}) - t_n)^2$$

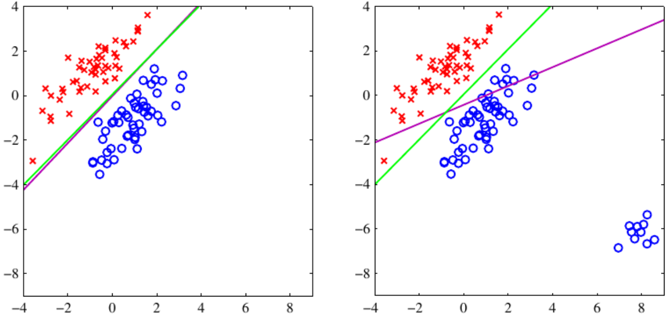
Notation:
1-of-K binary encoding for t:

$$t_n = (0, 0, 0, 1, 0, \dots, 0)$$

Least Squares

Problems

- least-squares suffers from outliers (figure)
- corresponds to ML in the assumption of Gaussian conditional distributions



Fisher's Linear Discriminant

- linear classification viewed as dimensionality reduction
- select projection that maximizes class separation

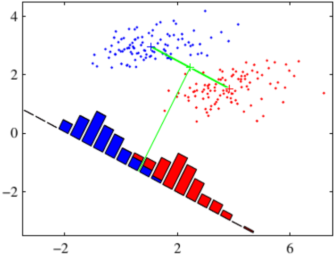
Example: 2 classes, C_1, N_1, C_2, N_2

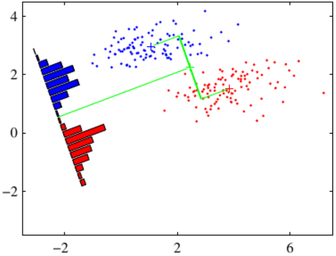
$$\mathbf{y} = \mathbf{w}^T\mathbf{x}$$

Measure of separation (to be maximized)

$$J(\mathbf{w}) = \frac{\mathbf{w}^T S_B \mathbf{w}}{\mathbf{w}^T S_W \mathbf{w}}$$

- S_B: between-class covariance
- S_W: within-class covariance





Fisher's Linear Discriminant

- linear classification viewed as dimensionality reduction
- select projection that maximizes class separation

Example: 2 classes, C_1, N_1, C_2, N_2

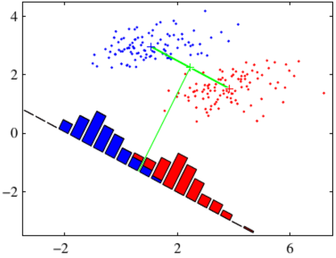
$$\mathbf{y} = \mathbf{w}^T \mathbf{x}$$

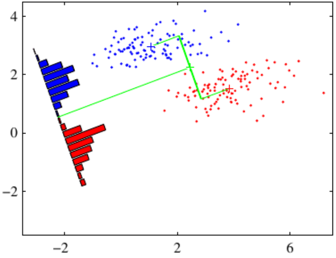
Measure of separation (to be maximized)

$$J(\mathbf{w}) = \frac{\mathbf{w}^T S_B \mathbf{w}}{\mathbf{w}^T S_W \mathbf{w}}$$

Differentiating $J(\mathbf{w})$:

$$\mathbf{w} \propto S_W^{-1}(\mathbf{m}_2 - \mathbf{m}_1)$$





Perceptron (Rosenblatt 1962)

$$y(\mathbf{x}) = f(\mathbf{w}^T \mathbf{x}) \qquad\qquad f(a) = \begin{cases} +1 & \text{if } a \geq 0 \\ -1 & \text{if } a < 0 \end{cases}$$

- in the book $\phi(\mathbf{x})$ instead of $\mathbf{x}$
- encode t with $+1$ for $\mathcal{C}_1$ and -1 for $\mathcal{C}_2$
- classify $\mathbf{x}$ according to sign of $\mathbf{w}^T \mathbf{x}$
- ideally for the training data $(\mathbf{w}^T \mathbf{x}_n) t_n > 0, \quad \forall n$

Error function

$$E_p(\mathbf{w}) = -\sum_{n \in \mathcal{M}} \mathbf{w}^T \mathbf{x}_n t_n \qquad\qquad \mathcal{M} = \text{miss-classified examples}$$

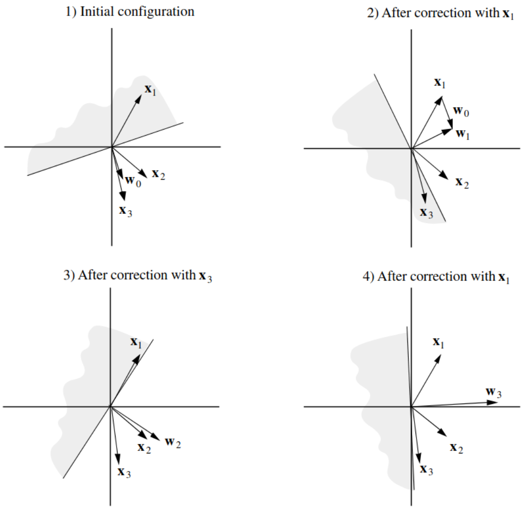
Stochastic Gradient descent

Parameter update for every miss-classified data point $(x_n, t_n) \in \mathcal{M}$

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta \nabla E_p(\mathbf{w}) = \mathbf{w}^{(t)} + \eta \mathbf{x}_n t$$

$$\eta = \text{learning rate} = 1$$

Not guaranteed to reduce error at every update.



Perception: Convergence Theorem

Convergence Theorem

If there exists a solution (linearly separable data)
then the perception will find it in a finite number of steps.

Problems:

- the number of steps could be substantial
- no way to know if the problem is linearly separable before convergence
- there usually are several solutions
- if not linearly separable, it never converges

Outline

Probabilistic Models

2 classes:

$$p(\mathcal{C}_1|\mathbf{x}) = \frac{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1)}{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1) + p(\mathbf{x}|\mathcal{C}_2)p(\mathcal{C}_2)} = \frac{1}{1+\exp(-a)} \quad \text{(sigmoid)}$$

where

$$a = \ln \frac{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1)}{p(\mathbf{x}|\mathcal{C}_2)p(\mathcal{C}_2)} \quad \text{log odds}$$

K classes:

$$p(\mathcal{C}_k|\mathbf{x}) = \frac{p(\mathbf{x}|\mathcal{C}_k)p(\mathcal{C}_k)}{\sum_j p(\mathbf{x}|\mathcal{C}_j)p(\mathcal{C}_j)} = \frac{\exp(a_k)}{\sum_j \exp(a_j)} \quad \text{(softmax)}$$

where

$$a_k = \ln p(\mathbf{x}|\mathcal{C}_k)p(\mathcal{C}_k)$$

Generative Approach

Given data $\mathcal{D} = \{(\mathbf{x}_1, t_1), \ldots, (\mathbf{x}_N, t_N)\}$ maximize log likelihood:

$$\ln \mathcal{L} = \sum_{n=1}^{N} p(\mathbf{x}_n, t_n|\theta) = \sum_{k=1}^{K} \sum_{n:\{t_n=k\}} \Big[\ln p(\mathbf{x}|\mathcal{C}_k, \theta_k) + \ln p(\mathcal{C}_k|\theta_k) \Big]$$

with $\theta = \{\theta_1, \ldots, \theta_K\}$, and θ_k are the class dependent model parameters.

Example:

- Categorical priors: $p(\mathcal{C}_k|\theta_k) = \pi_k$, with $\sum_k \pi_k = 1$
- Gaussian class conditional likelihoods $p(\mathbf{x}|\mathcal{C}_k, \theta_k) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$
- $\theta = \{\underbrace{\pi_1, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1}_{\theta_1}, \ldots, \underbrace{\pi_K, \boldsymbol{\mu}_K, \boldsymbol{\Sigma}_K}_{\theta_K}\}$

Class-conditional optimization

Given data $\mathcal{D} = \{(\mathbf{x}_1, t_1), \ldots, (\mathbf{x}_N, t_N)\}$ maximize log likelihood:

$$\ln \mathcal{L} = \sum_{n=1}^{N} p(\mathbf{x}_n, t_n|\theta) = \sum_{k=1}^{K} \sum_{n:\{t_n=k\}} \Big[\ln p(\mathbf{x}|\mathcal{C}_k, \theta_k) + \ln p(\mathcal{C}_k|\theta_k) \Big]$$

Differentiating w.r.t. θ_k, only contributions from $(\mathbf{x}_n, t_n)$ for which $t_n = k$!

1. split data into K subset according to class label t_n
2. optimize $p(\mathbf{x}|\mathcal{C}_k, \theta_k)$ and $p(\mathcal{C}_k|\theta_k)$ independently using ML

Example: Gaussian class-conditional likelihoods

Categorical priors

$$p(\mathcal{C}_k|\theta_k) = \pi_k, \text{ with } \sum_k \pi_k = 1$$

Maximum likelihood: $\pi_k = \frac{1}{N} \sum_{n:\{t_n=k\}} 1 = \frac{N_k}{N}$

Gaussian class conditional likelihoods

$$p(\mathbf{x}|\mathcal{C}_k, \theta_k) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

Maximum likelihood:

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{n:\{t_n=k\}} \mathbf{x}_n$$

$$\boldsymbol{\Sigma}_k = \frac{1}{N_k} \sum_{n:\{t_n=k\}} (\mathbf{x}_n - \boldsymbol{\mu}_k)^T (\mathbf{x}_n - \boldsymbol{\mu}_k)$$

Special Case: Gaussians with Equal Covariance

2-class problem with $\mathbf{\Sigma}_1 = \mathbf{\Sigma}_2 = \mathbf{\Sigma}$

$$p(\mathcal{C}_1|\mathbf{x}) = \frac{1}{1+\exp(-a)} = \sigma(a) \quad \text{with } a = \ln \frac{\pi \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_1, \mathbf{\Sigma})}{(1-\pi)\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_2, \mathbf{\Sigma})} \quad \text{log odds}$$

Equal normalization term in $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_1, \mathbf{\Sigma})$ and $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_2, \mathbf{\Sigma})$, expanding and simplifying:

$$\begin{aligned} a &= \mathbf{w}^T\mathbf{x} + w_0, \quad \text{with} \\ \mathbf{w} &= \mathbf{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2), \quad \text{and} \\ w_0 &= -\frac{1}{2}\boldsymbol{\mu}_1^T\mathbf{\Sigma}^{-1}\boldsymbol{\mu}_1 + \frac{1}{2}\boldsymbol{\mu}_2^T\mathbf{\Sigma}^{-1}\boldsymbol{\mu}_2 + \ln\frac{\pi}{1-\pi} \end{aligned}$$

Linear classifier!!

Discriminative Approach: Logistic Regression

- 2-class problem: posterior as sigmoid
- use ML to maximize $P(\mathcal{C}_k|\mathbf{x})$ directly.

$$P(\mathcal{C}_k|\mathbf{x}) = \sigma(\mathbf{w}^T\mathbf{x}) = \frac{1}{1+\exp(-\mathbf{w}^T\mathbf{x})}$$

$$E(\mathbf{w}) = -\ln p(\mathbf{t}|\mathbf{w}) = -\sum_{n=1}^{N}\{t_n \ln y_n + (1-t_n)\ln(1-y_n)\}$$

$$\nabla E(\mathbf{w}) = \sum_{n=1}^{N}(y_n - t_n)x_n$$

- non-linear in $\mathbf{w}$ $\Rightarrow$ no closed form solution
- but $E(\mathbf{w})$ is convex $\Rightarrow$ single global minimum
- can be optimized with iterative algorithm

Number of Parameters, 2-class problem, M dimensional $\mathbf{x}$

Generative model
(equal covariance matrices):

Parameter	Degrees of freedom
$\boldsymbol{\mu}_1$	M
$\boldsymbol{\mu}_2$	M
$\boldsymbol{\Sigma}$	$\frac{M(M+1)}{2}$
$P(C_1)$	1
Total	$\frac{M(M+5)}{2} + 1$

Discriminative model
(logistic regression)

Parameter	Degrees of freedom
$\mathbf{w}$	$M + 1$
Total	$M + 1$