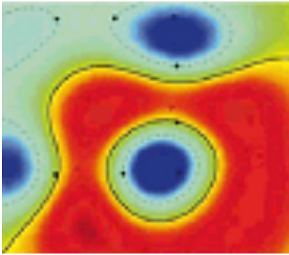
WiSe 2022/23

Machine Learning 1/1-X

Technische Universität Berlin



Lecture 3

Fisher Discriminant

Outline

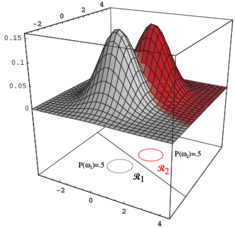
- Recap:
  - Bayes Optimal Classifier
  - Parameter Estimation
- Classification without Learning Distributions
  - Mean Separation
  - Fisher Discriminant
  - Perceptron
  - Large Margin Classifiers

Recap: Bayes Optimal Classifier

Recap:

- Assume our data is generated for each class ω_j according to the multivariate Gaussian distribution $p(\mathbf{x}|\omega_j) = \mathcal{N}(\boldsymbol{\mu}_j, \Sigma)$ and with class priors $P(\omega_j)$. The Bayes optimal classifier is derived as

$$\arg\max_j \{P(\omega_j|\mathbf{x})\}$$
$$= \arg\max_j \{\log p(\mathbf{x}|\omega_j) + \log P(\omega_j)\}$$
$$= \arg\max_j \left\{\mathbf{x}^\top \Sigma^{-1} \boldsymbol{\mu}_j^\top - \frac{1}{2}\boldsymbol{\mu}_j \Sigma^{-1} \boldsymbol{\mu}_j + \log P(\omega_j)\right\}$$



- Given our generative assumptions, there is no better classifier than the one above.
- However, in practice, we don't know these distributions and only have the data.

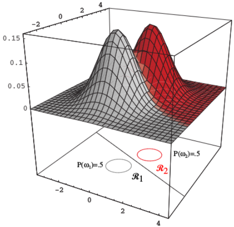
Recap: Parameter Estimation

Example of estimator:

- Maximum likelihood estimator:

$$\boldsymbol{\mu} = \frac{1}{N}\sum_{k=1}^{N} \mathbf{x}_k$$

$$\widehat{\Sigma} = \frac{1}{N}\sum_{k=1}^{N} (\mathbf{x}_k - \boldsymbol{\mu})(\mathbf{x}_k - \boldsymbol{\mu})^\top$$



Problem:

- The covariance matrix (and its inverse) may be difficult to estimate.
- We make an assumption about the data (e.g. Gaussian-distributed) which may not correspond to reality.

Distribution-Free Approaches

> "When solving a problem of interest, do not solve a more general problem as an intermediate step." (V. Vapnik)

Interpretation in our setting:

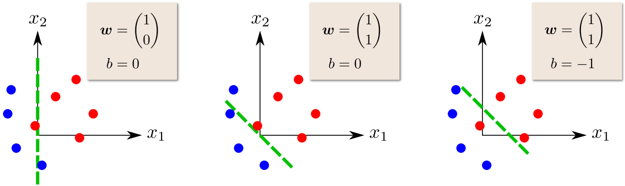
- Don't take the intermediate step of learning distributions to build the classifier. Build the classifier directly.
- That is, rather than assuming a set of distributions (e.g. Gaussian), assume a set of models (e.g. linear), and find the parameters of the model that optimize some classification objective (e.g. based on the statistics of the data in projected space).

Linear Classifiers

- Functions of the type

$$f(\mathbf{x}; \mathbf{w}, b) = \mathbf{w}^\top \mathbf{x} + b$$

  where $\mathbf{w} \in \mathbb{R}^d$ and $b \in \mathbb{R}$ are parameters to learn. We decide for class ω_1 if $f(\mathbf{x}) > 0$ and for class ω_2 if $f(\mathbf{x}) < 0$.
- Examples of linear classifiers on a simple 2d exmaple:



- **Question:** Based on what criterion do we choose the parameters $\mathbf{w}$, b?

Mean Separation Criterion

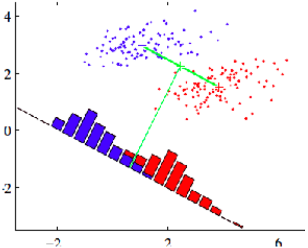
Idea:

- Build a projection the data $z = \mathbf{w}^\top \mathbf{x}$ with $\|\mathbf{w}\| = 1$ such that the means of classes in projected space are as distant as possible.

Approach:

- First, we compute the means in projected space for the two classes

$$\mu_1 = \frac{1}{N_1} \sum_{k \in \mathcal{C}_1} z_k \qquad \mu_2 = \frac{1}{N_2} \sum_{k \in \mathcal{C}_2} z_k$$



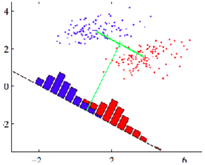
- Then we would like to find $\mathbf{w}$ that maximizes the difference of means, i.e. we express the means as a function of $\mathbf{w}$ and pose the optimization problem:

$$\arg\max_{\mathbf{w}} |\mu_2(\mathbf{w}) - \mu_1(\mathbf{w})| \qquad \text{with} \quad \|\mathbf{w}\| = 1$$

Derivation of Mean Separation

- The constrained optimization problem (subject to the constraint $\|\mathbf{w}\| = 1$) can be developed as:

$$
\begin{aligned}
&\arg\max_{\mathrm{w}} |\mu_2(\mathbf{w}) - \mu_1(\mathbf{w})| \\
&\quad = \arg\max_{\mathrm{w}} \left| \frac{1}{N_2} \sum_{k \in \mathcal{C}_2} z_k(\mathbf{w}) - \frac{1}{N_1} \sum_{k \in \mathcal{C}_1} z_k(\mathbf{w}) \right| \\
&\quad = \arg\max_{\mathrm{w}} \left| \frac{1}{N_2} \sum_{k \in \mathcal{C}_2} \mathbf{w}^\top \mathbf{x}_k - \frac{1}{N_1} \sum_{k \in \mathcal{C}_1} \mathbf{w}^\top \mathbf{x}_k \right| \\
&\quad = \arg\max_{\mathrm{w}} \left| \mathbf{w}^\top \left(\frac{1}{N_2} \sum_{k \in \mathcal{C}_2} \mathbf{x}_k \right) - \mathbf{w}^\top \left(\frac{1}{N_1} \sum_{k \in \mathcal{C}_1} \mathbf{x}_k \right) \right| \\
&\quad = \arg\max_{\mathrm{w}} \left| \mathbf{w}^\top (\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1) \right|
\end{aligned}
$$

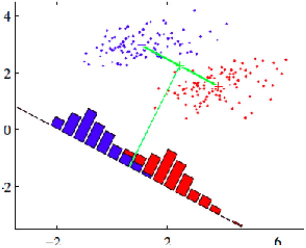


where $\boldsymbol{\mu}_2$ and $\boldsymbol{\mu}_1$ are the means in *input space*.

- The best vector $\mathbf{w}$ is the one that aligns with $(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)$, i.e. $\mathbf{w} = (\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)/\|\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1\|$.

Limitations of Mean Separation

- There is a significant class overlap in projected space.
- A better classifier seems achievable if we rotate the projection a few degrees clockwise.
- Making means distant may not be sufficient to induce class separability in projected space.



Fisher Discriminant

R.A. Fisher (1890 - 1962)

Idea:

- In addition to maximizing the separation between class means in projected space, also consider to reduce the within-class variance.

$$\mu_1 = \frac{1}{|\mathcal{C}_1|} \sum_{k \in \mathcal{C}_1} z_k \qquad \mu_2 = \frac{1}{|\mathcal{C}_2|} \sum_{k \in \mathcal{C}_2} z_k$$

$$s_1 = \sum_{k \in \mathcal{C}_1} (z_k - \mu_1)^2 \qquad s_2 = \sum_{k \in \mathcal{C}_2} (z_k - \mu_2)^2$$

- Maximizing distance between means while minimizing within-class variance can be formulated as:

$$\arg\max_{w} \frac{(\mu_2(\mathbf{w}) - \mu_1(\mathbf{w}))^2}{s_1(\mathbf{w}) + s_2(\mathbf{w})}$$

Deriving the Fisher Discriminant (1)

The within-class variance can be expanded as:

$$
\begin{aligned}
s_j(\mathbf{w}) &= \sum_{k \in \mathcal{C}_j} (z_k - \mu_j)^2 \\
&= \sum_{k \in \mathcal{C}_j} (\mathbf{w}^\top \mathbf{x}_k - \mathbf{w}^\top \boldsymbol{\mu}_j)^2 \\
&= \mathbf{w}^\top \underbrace{\sum_{k \in \mathcal{C}_j} (\mathbf{x}_k - \boldsymbol{\mu}_j)(\mathbf{x}_k - \boldsymbol{\mu}_j)^\top}_{S_j} \mathbf{w}
\end{aligned}
$$

where S_j is a scatter matrix (unnormalized covariance matrix) for the data of class j.

Observations:

- Similar structure as the PCA objective (but for each class separately)
- Unlike PCA, we want to *minimize* the variance rather than maximize it.

Deriving the Fisher Discriminant (2)

Making use of the results of Slide 7 and 10, we can rewrite the Fisher objective

$$J(\mathbf{w}) = \frac{(\mu_2(\mathbf{w}) - \mu_1(\mathbf{w}))^2}{s_1(\mathbf{w}) + s_2(\mathbf{w})}$$

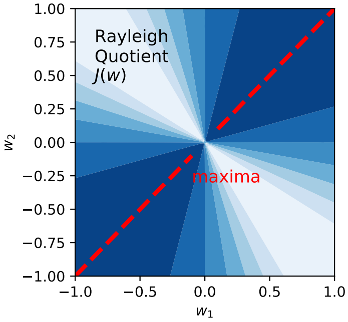
as

$$\boxed{J(\mathbf{w}) = \frac{\mathbf{w}^\top \mathbf{S}_B \, \mathbf{w}}{\mathbf{w}^\top \mathbf{S}_W \, \mathbf{w}}} \qquad (1)$$

where

$$\mathbf{S}_B = (\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)^\top$$
$$\mathbf{S}_W = S_1 + S_2$$

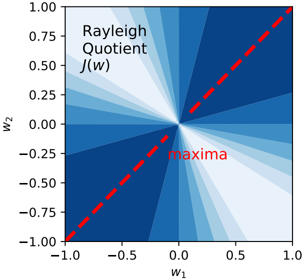
are the '**B**etween-Class' and '**W**ithin-Class' scatter matrices respectively. The form of Eq. (1) is known as *Rayleigh Coefficient*.



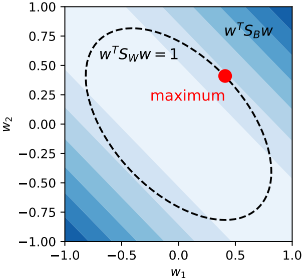
Deriving the Fisher Discriminant (3)

- A solution that maximizes the Rayleigh quotient $J(\mathbf{w})$ can be obtained by first observing that $\forall_{\alpha \neq 0}: \; J(\alpha\mathbf{w}) = J(\mathbf{w})$ and searching for the particular solution for which the denominator is exactly one. This can be stated as the constrained optimization problem

$$\boxed{\arg\max_{\mathbf{w}} \mathbf{w}^\top \mathbf{S}_B \mathbf{w} \quad \text{s.t.} \quad \mathbf{w}^\top \mathbf{S}_W \mathbf{w} = 1}$$



Constrained formulation $\longrightarrow$



Deriving the Fisher Discriminant (4)

We start with the constrained optimization problem:

$$\arg\max_{\mathbf{w}} \mathbf{w}^\top \mathbf{S}_B \mathbf{w} \quad \text{s.t.} \quad \mathbf{w}^\top \mathbf{S}_W \mathbf{w} = 1$$

Method of Lagrange Multipliers

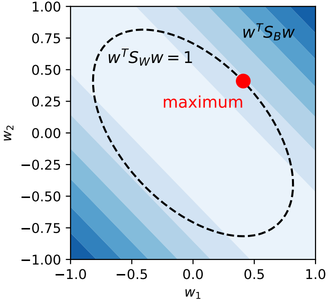
Step 1: We build the Lagrangian

$$\mathcal{L}(\mathbf{w}, \lambda) = \mathbf{w}^\top \mathbf{S}_B \mathbf{w} + \lambda \cdot (1 - \mathbf{w}^\top \mathbf{S}_W \mathbf{w})$$

Step 2: We look for potential solutions by posing $\nabla \mathcal{L}(\mathbf{w}, \lambda) = \mathbf{0}$, which leads to the equation:

$$\boxed{\mathbf{S}_B \mathbf{w} = \lambda \mathbf{S}_W \mathbf{w}}$$

This is a generalized eigenvalue problem.



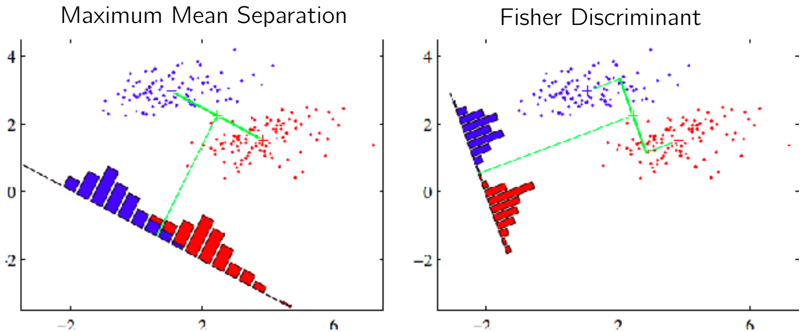
Deriving the Fisher Discriminant (5)

Further steps:

- If $\mathbf{S}_W$ is invertible, it can be restated as the standard eigenvalue problem
$$\left(\mathbf{S}_W^{-1}\mathbf{S}_B\right)\mathbf{w} = \lambda\mathbf{w}$$
- Expanding the term $\mathbf{S}_B$, we get:
$$\mathbf{S}_W^{-1}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)\underbrace{(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)^\top\mathbf{w}}_{\text{scalar!}} = \lambda\mathbf{w}$$
- Therefore, one possible solution for **w** is given by
$$\boxed{\mathbf{w} = \mathbf{S}_W^{-1}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)}$$

(Reminder, in our original Rayleigh quotient formulation, $J(\alpha\mathbf{w}) = J(\mathbf{w})$*, i.e. the optimum is defined up to a scaling factor.)*

Mean Separation vs. Fisher Discriminant



- Fisher Discriminant leads (in general) to better class separability, and therefore, better classification accuracy.
- Fisher Discriminant requires inversion of a covariance matrix (only tractable for low-dimensional data).

Decision Theory vs. Fisher Discriminant

Bayes decision theory (Lecture 1)

Discriminant has the form

$$\mathbf{w} = \Sigma^{-1}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)$$

- ...under the assumption that the data-generating distributions are *Gaussian* (with means $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2$ for each class, and *same* covariance Σ).
- Bias is also given by the analysis (cf. Slide 2).

Fisher discriminant (today)

Discriminant has the form

$$\mathbf{w} = \mathbf{S}_W^{-1}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)$$

- No generative assumption. Classifier only derived from a criterion on mean and dispersion of data in projected space.
- Bias is not provided by the analysis but it can be fitted in a second step.

Application: P300 BCI Speller

BCI with ML: Calibration and Feedback

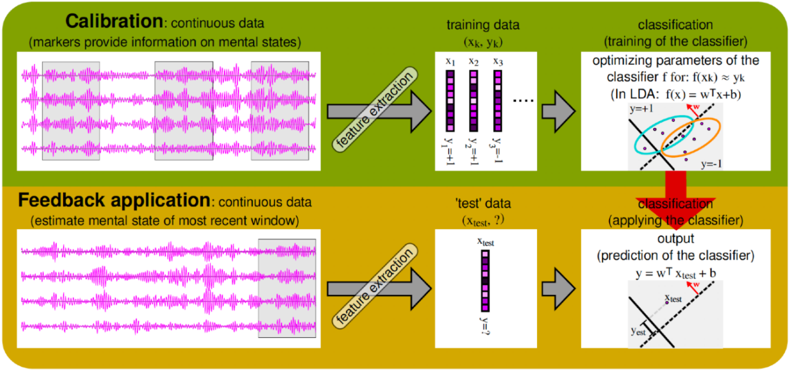
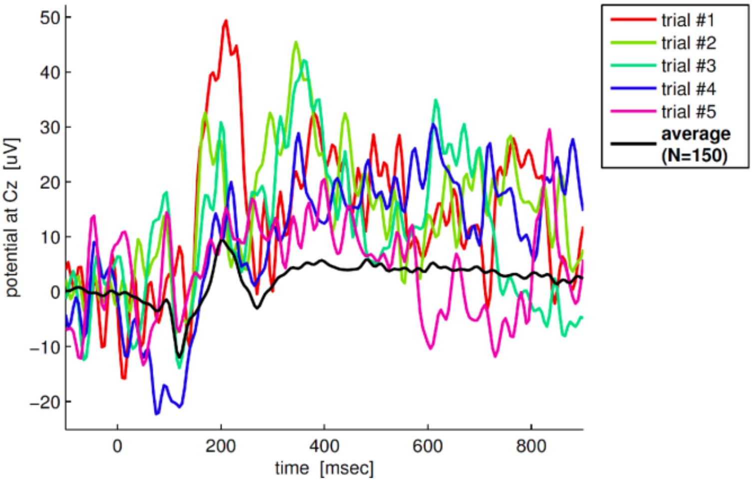
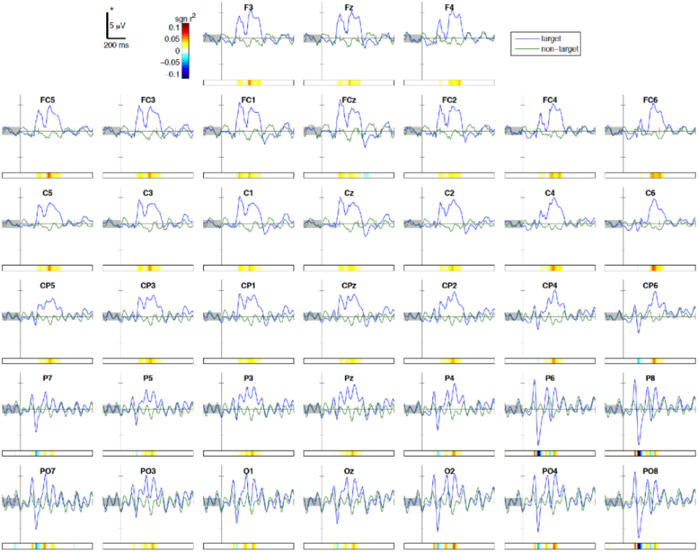


Illustration: Single-Trials and ERPs

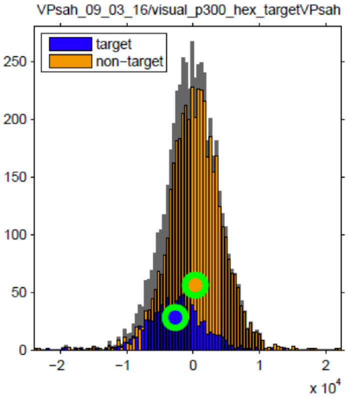


Scalp Potentials In Response to (Non-)Targets

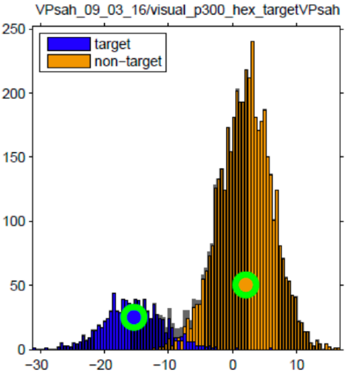


A BCI example: P300 speller

Mean Separation



Fisher Discrminant



Fisher Discriminant (Strengths/Limitations)

Strengths:

- Accurate when the means/covariances describe well the data, and close to optimal when the data is Gaussian of fixed covariance.
- The Fisher discriminant is given in closed form and is fast to compute.

Limitations:

- Although applicable to non-Gaussian distributions, the resulting decision boundary can become in that case strongly suboptimal.
- In particular, like principal component analysis, Fisher Discriminant is not robust to outliers.

Idea:

- To overcome these limitations, we will discuss other learning algorithms that more specifically focus on modeling the decision boundary between the two classes.

The Perceptron

F. Rosenblatt (1928–1971)

- Proposed by F. Rosenblatt in 1958.
- Classifier that prefectly separates training data (if the data is linearly separable).
- Trained using an simple and cheap iterative procedure.
- The perceptron gave rise to artificial neural networks.

The Perceptron Algorithm

- Consider our linear model

$$z_k = \mathbf{w}^\top \mathbf{x}_k + b \qquad y_k = \text{sign}(z_k)$$

and let t_k be 1 and -1 when the true class of $\mathbf{x}_k$ is ω_1 and ω_2 respectively.

Algorithm

- Iterate over all examples $k = 1 \dots, N$ (multiple times).
  - If example $\mathbf{x}_k$ is correctly classified($y_k = t_k$), continue.
  - If example $\mathbf{x}_k$ is wrongly classified ($y_k \neq t_k$), apply:

$$\mathbf{w} \leftarrow \mathbf{w} + \eta \cdot \mathbf{x}_k t_k \tag{2}$$

$$b \leftarrow b + \eta \cdot t_k \tag{3}$$

  where η is a learning rate.
- The algorithm stops once all examples are correctly classified.

The Perceptron: Optimization View

- The perceptron can be seen as the minimization of the error function

$$\mathcal{E}(\mathbf{w}, b) = \frac{1}{N} \sum_{k=1}^{N} \underbrace{\max(0, -z_k t_k)}_{\mathcal{E}_k(\mathbf{w}, b)}$$

- *Proof:* Computing the gradient of $\mathcal{E}_k$ gives

$$\begin{aligned}
\nabla_{\mathrm{w}} \mathcal{E}_k(\mathbf{w}, b) &= 1_{-z_k t_k > 0} \cdot (-\mathbf{x}_k t_k) \\
&= 1_{y_k \neq t_k} \cdot (-\mathbf{x}_k t_k) \\
&= \begin{cases} 0 & y_k = t_k \\ -\mathbf{x}_k t_k & y_k \neq t_k \end{cases}
\end{aligned}$$

  And we observe that the update rule $\mathbf{w} \leftarrow \mathbf{w} - \eta \cdot \nabla_{\mathrm{w}} \mathcal{E}_k(\mathbf{w}, b)$ is equivalent to that of Eq. (2)

- Similar result can be obtained for the bias.

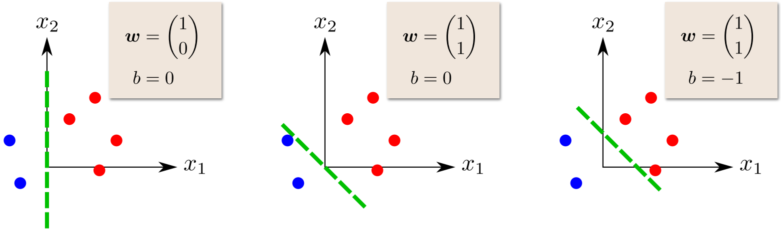
The Perceptron: Optimization View

- Recall: The objective function corresponding to the perceptron algorithm is given by:

$$\mathcal{E}(\mathbf{w}, b) = \frac{1}{N} \sum_{k=1}^{N} \underbrace{\max(0, -z_k t_k)}_{\mathcal{E}_k(\mathbf{w}, b)}$$

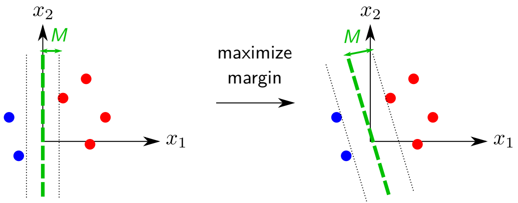
- This objective can be interpreted as measuring for each wrongly classified data point how far the data point is from the decision boundary, and penalizing accordingly.
- In practice various strategies can be implemented to optimize this objective (e.g. gradient descent on $\mathcal{E}(\mathbf{w}, b)$ directly, or adding momentum to the gradient descent).
- Optimization of the objective also works for data that is not linearly separable.

A Problem of the Perceptron



- All these solutions have an error $\mathcal{E}(\mathbf{w}, b) = 0$.
- Some solutions are obviously better than other, e.g. those where the decision boundary is separated from the data with a large margin.

Large Margin Classifiers

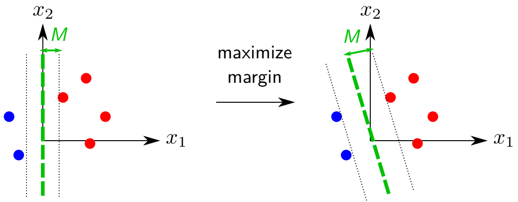


Idea: Induce large margin M by redefining the optimization problem as:

$$\underbrace{\arg\min_{\mathrm{w},b,M}}_{(1)} \left[\underbrace{\frac{1}{M}}_{(2)} + \frac{1}{N} \sum_{k=1}^{N} \underbrace{\max(0, M - (\mathbf{w}^\top \mathbf{x}_k + b) t_k)}_{(3)} \right] \quad \text{s.t.} \quad \underbrace{\|\mathbf{w}\| = 1}_{(4)} \qquad (4)$$

- (1) actively optimize the margin
- (2) apply a penalty if the margin is too large
- (3) apply a penalty if some points violate the margin and
- (4) constrain the projection so that M is interpretable as a margin.

Large Margin Classifiers



- An equivalent (and more standard) formulation of the optimization problem is given by:

$$\boxed{\arg\min_{\mathrm{w},b} \frac{1}{N} \sum_{k=1}^{N} \max(0, 1 - (\mathbf{w}^\top \mathbf{x}_k + b)t_k) + \|\mathbf{w}\|^2} \tag{5}$$

(proof in next slide).

- Large Margin Classifiers are typically solved using (stochastic) gradient descent or quadratic programming.
- More will be said about these model during the lecture on support vector machines.

Derivation of Eq. (5) from Eq. (4)

$$\boxed{\arg\min_{\mathrm{w},b,M} \frac{1}{N}\sum_{k=1}^{N} \max(0, M - (\mathrm{w}^\top \mathrm{x}_k + b)t_k) + \frac{1}{M} \quad \text{s.t.} \quad \|\mathrm{w}\| = 1}$$

The same objective can be rewritten as:

$$= \arg\min_{\mathrm{w},b,M} \frac{1}{N}\sum_{k=1}^{N} \max(0, M - (\mathrm{w}^\top \mathrm{x}_k + b)t_k) + \frac{\|\mathrm{w}\|^2}{M} \quad \text{s.t.} \quad \|\mathrm{w}\| = 1$$

Dividing by M does not change the argmin nor the constraint

$$= \arg\min_{\mathrm{w},b,M} \frac{1}{N}\sum_{k=1}^{N} \max(0, 1 - (\frac{\mathrm{w}^\top}{M} \mathrm{x}_k + \frac{b}{M})t_k) + \frac{\|\mathrm{w}\|^2}{M^2} \quad \text{s.t.} \quad \frac{\|\mathrm{w}\|}{M} = \frac{1}{M}$$

and redefining $\mathrm{w} \leftarrow \mathrm{w}/M$ and $b \leftarrow b/M$, we get:

$$= \arg\min_{\mathrm{w},b,M} \frac{1}{N}\sum_{k=1}^{N} \max(0, 1 - (\mathrm{w}^\top \mathrm{x}_k + b)t_k) + \|\mathrm{w}\|^2 \quad \text{s.t.} \quad \|\mathrm{w}\| = \frac{1}{M}$$

Because optimizing w.r.t. M is now trivial, we can further simplify to

$$= \boxed{\arg\min_{\mathrm{w},b} \frac{1}{N}\sum_{k=1}^{N} \max(0, 1 - (\mathrm{w}^\top \mathrm{x}_k + b)t_k) + \|\mathrm{w}\|^2}$$

Advanced Classification Topics

- **Nonlinear Classification** (Kernel SVM, Artificial Neural Networks, Decision Trees, Random Forests, Boosting). (Covered in ML1)
- Incorporating **prior knowledge** such as invariances into a classifier. Example: the convolutional neural network. (Covered in ML2)
- Classification in **high dimensions**: Unintuitively, methods that actually promote within-class variance instead of reducing it tend to perform better in this setting.

Summary

- In practice, it is preferable to **train a classifier directly** rather than learning the class distributions in the first place.
- The **maximum mean separation** and **Fisher discriminant** are two such instances, where one only needs to estimate the mean and the covariance of the data, without having to estimate full distributions.
- The **perceptron** (and its **large-margin** extension) more specifically focus on the decision boundary, which typically leads to higher classification accuracy on general tasks.