Machine Learning I Lecture VI: Linear models for Classification

Jakob H Macke
Max Planck Institute for Biological Cybernetics
Bernstein Center for Computational Neuroscience

XY.XY.2012

Plan for today

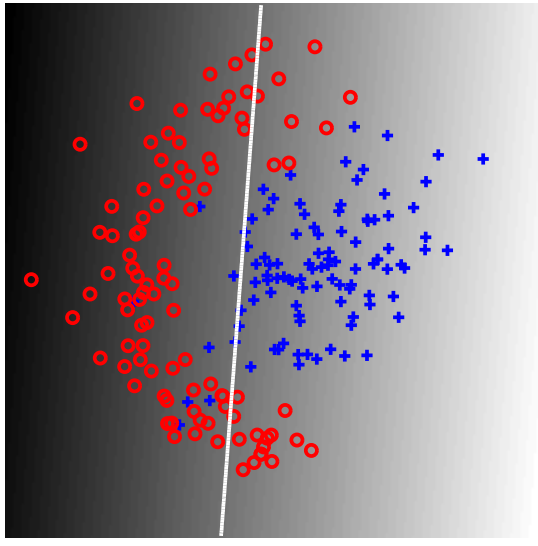
Binary classification

Least Square Classification

Fisher's linear discriminant

A generative model: Class-conditional Gaussians

Binary Classification: Assign each data point to one of two classes.

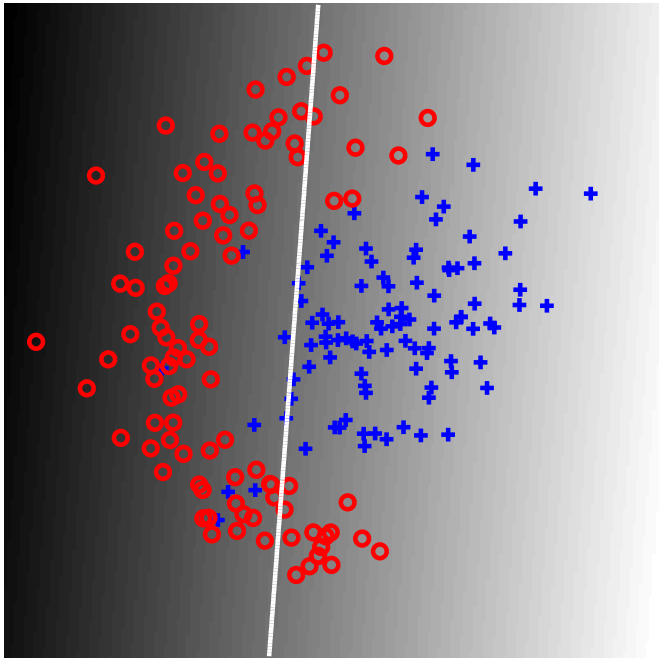


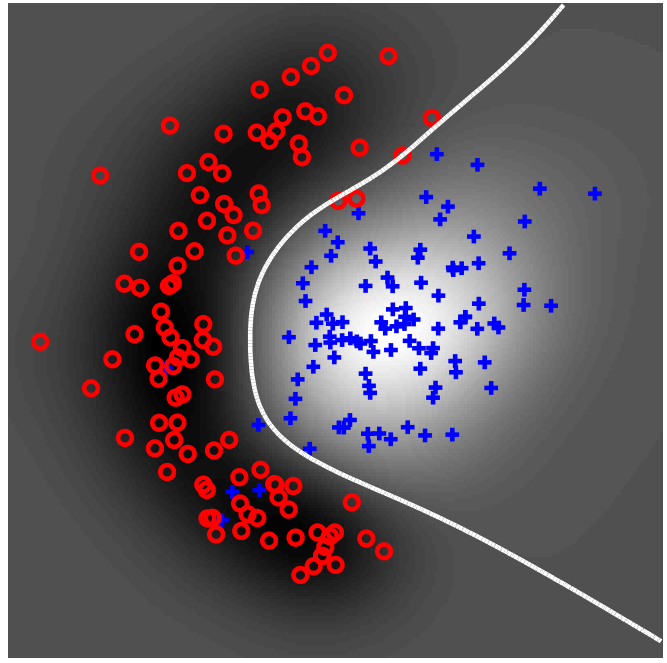
Examples:

- Is there a face in this image?
- Will this neuron spike in response to this stimulus?
- Based on this brain-scan, does this patient have a given disease or not?
- Will this customer buy this product or not?
- Is this person likely to be a democrat/republican?

Notation: we have data $D = \{(x_1, t_1), \ldots, (x_N, t_N)\}$, with $t_n = 1$ if x_n belongs to class 1 and $t_n = -1$ if x_n belongs to class -1.

We focus on linear decision rules, also known as 'linear discriminant functions'.





Of course, linear algorithms can be used together with nonlinear feature spaces or nonlinear basis functions in order to solve nonlinear classification problems!

Linear discriminants separate the space by a hyperplane, and the parameters define its normal vector.

- Decision function: $y(\mathbf{x}) = \omega^\top \mathbf{x} + \omega_o$
- Classification:

$$\text{if } y(\mathbf{x}) > 0 \text{ say } \mathbf{x} \text{ belongs to class 1} \quad (1)$$
$$\text{if } y(\mathbf{x}) < 0 \text{ say } \mathbf{x} \text{ belongs to class -1} \quad (2)$$
$$(3)$$

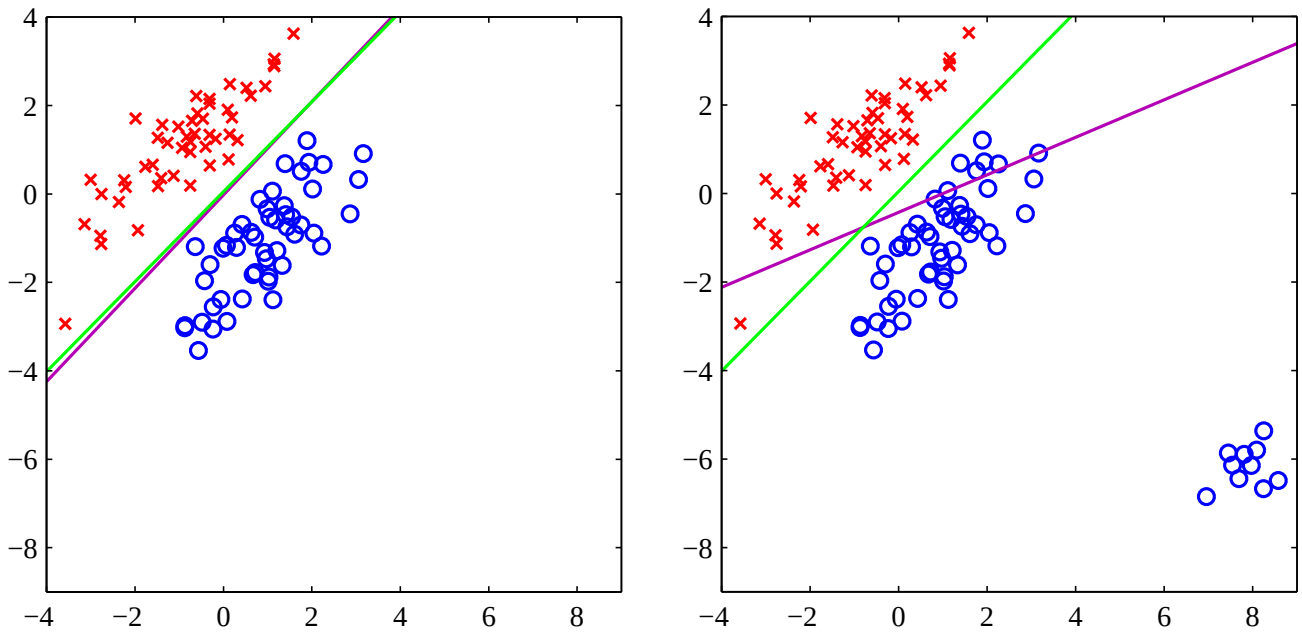
- The decision-surface has equation $y(\mathbf{x}) = 0$, and is a hyperplane of dimensionality $D - 1$.
- ω is the normal vector to the plane, and points into the positive class.
- ω_o determines the location of the decision-surface
- $|y(\mathbf{x})|$ is proproptional to the perpendicular distance to the decision-surface (with factor 1 if $||\omega|| = 1$).

Multiple algorithms and methods exist for finding a good ω.

- Mis-classification rate $C(\omega) = \frac{1}{N} \sum_n \delta \left[y(\mathbf{x}_n) = t_n \right]$ (i.e. average number of errors) difficult to optimize over ω, and might have multiple solutions.
- Many algorithms can be derived by replacing C by another cost-function which can be optimized.
- Linear classification algorithms include Least-square classification, Fisher's linear Discriminant, Logistic regression, Support Vector Machines and Rosenblatts' perceptron.

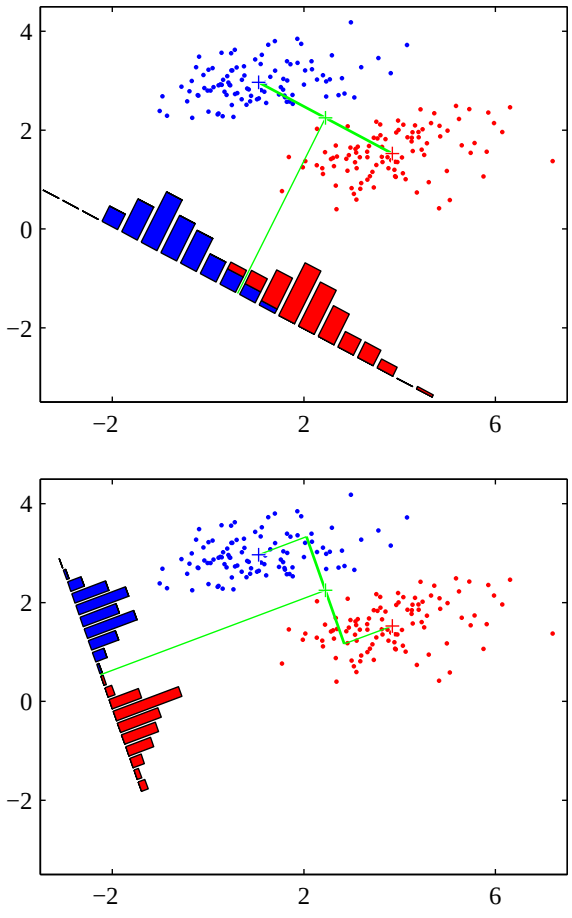
You already know one algorithm for linear classification: least square classification.

- We have to fit the function $y(\mathbf{x}) = \omega^\top \mathbf{x} + \omega_o$ to data.
- Simply do a linear regression from $\mathbf{x}$ to t by minimizing the sum-of-squared errors $\sum_n (y(\mathbf{x}_n) - t_n)^2$.
- $\omega_{reg} = \left(\sum_n x_n x_n^\top\right)^{-1} \sum_n x_n t_n$
- Q: In what situations might this be a bad idea?



Bishop PRML Figure 4.4

'Fisher's linear discriminant' is a classical and simple algorithm for linear classification



Bishop PRML Figure 4.6

- $\mathbf{m}_+ = \frac{1}{N_+} \sum_{n \in C_+} x_n$
  $\mathbf{m}_- = \frac{1}{N_-} \sum_{n \in C_-} x_n$
- Maximize projection-distance of class means [projected mean/variance: on board]
  $\omega_{simple} \propto \mathbf{m}_+ - \mathbf{m}_-$
- Maximizing distance between means ignores that the projected variances might also be big.
- Fix: Maximize the ratio of between-class variance to within-class variance ('signal to noise'). Fisher criterion

$$J_\omega = 2 \frac{(m_+ - m_-)^2}{s_+^2 + s_-^2} \tag{4}$$

[Details and solution: on board]
$\omega_{lda} = \Sigma_w^{-1}(\mathbf{m}_+ - \mathbf{m}_-)$

Aside: The multivariate Gaussian

- Probability density function of D dimensional Gaussian with mean μ an covariance Σ:

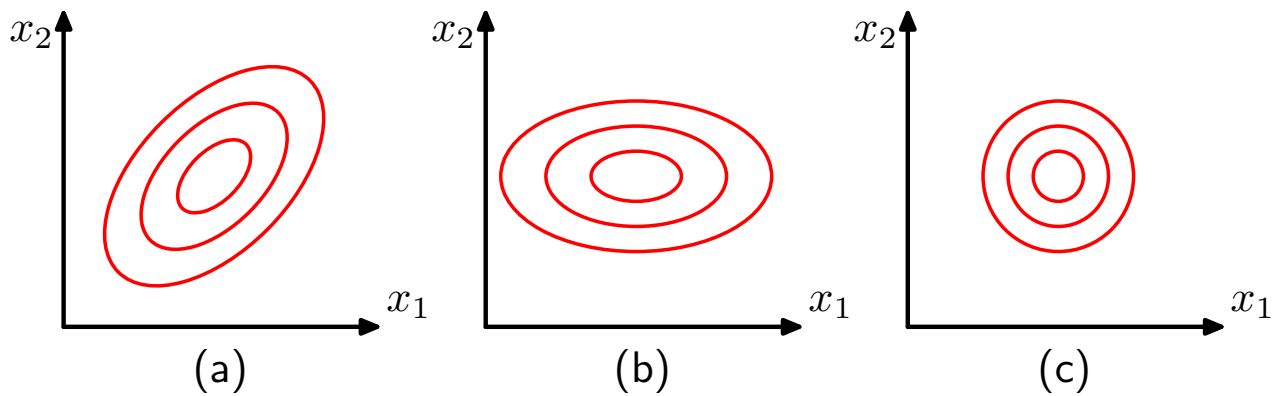
$$p(x|\mu,\Sigma) = (2\pi)^{-D/2}|\Sigma|^{-1/2}\exp\left(-\frac{1}{2}(x-\mu)^\top\Sigma^{-1}(x-\mu)\right) \quad (5)$$

$$(6)$$

- Maximum likelihood estimation of parameters:

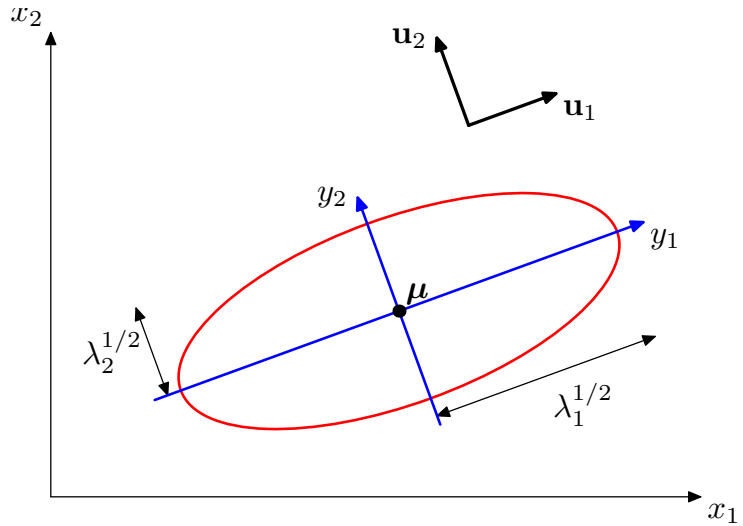
$$\hat{\mu} = \frac{1}{N}\sum_n x_n \quad \text{(empirical mean)} \quad (7)$$

$$\hat{\Sigma} = \frac{1}{N}\sum_n x_n x_n^\top - \hat{\mu}\hat{\mu}^\top \quad \text{(empirical covariance)} \quad (8)$$



Bishop PRML Figure 2.8

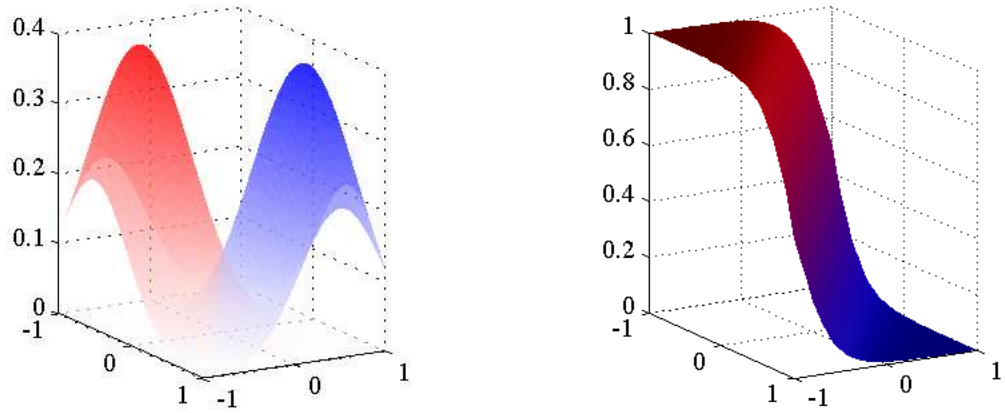
A (super brief) primer on covariance matrices. (more details/intuition in second half of course?)



Bishop PRML Figure 2.7

- Covariance matrices are symmetric.
- Diagonal entries: variances along coordinate-axes
- Eigenvectors: principal axes of ellipsoid
- Eigenvalues: variances along eigen-vectors
- Eigenvector with maximal/minimal eigen-value: Direction of maximal/minimal variance
- Covariance matrices are 'positive definite', i.e. all their eigenvalues are non-negative.
- Most of this can be derived from $a^\top \mathrm{Cov}(X)a = \mathrm{Var}(a^\top X)$

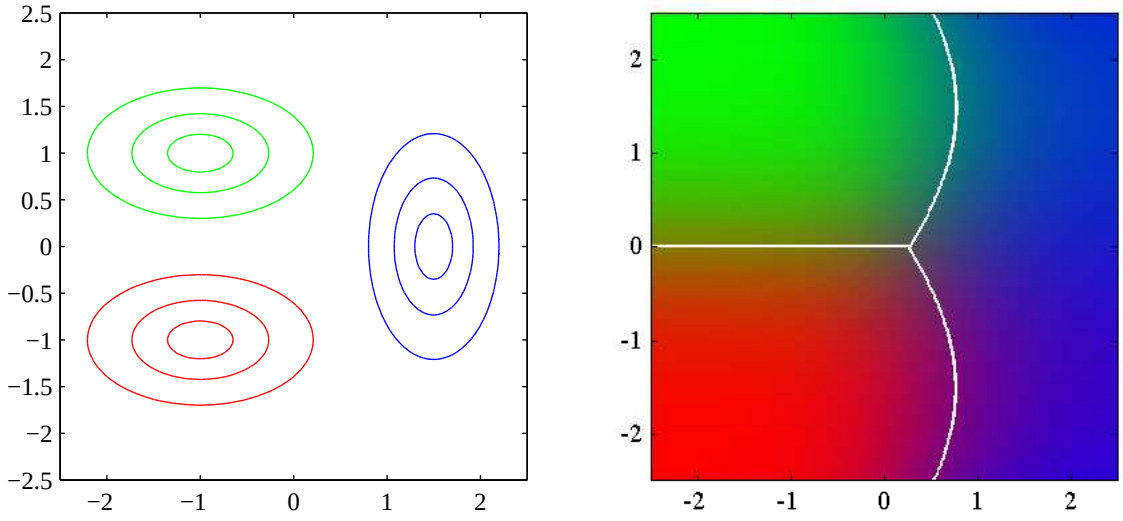
A tale of two Gaussian: We can use a probablistic model of the data for classification



- Suppose that each of the two classes is modelled by a Gaussian: $x|x \in C_+ \sim \mathcal{N}(\mu_+, \Sigma_+)$, $x|x \in C_- \sim \mathcal{N}(\mu_-, \Sigma_-)$,
- [On board] Calculation of posterior class probabilities and decision criterion
- If we assume $\Sigma_+ = \Sigma_-$, we get $\omega_{gauss} \propto \Sigma_+^{-1}(\mathbf{m}_+ - \mathbf{m}_-)$
- Note: We take the t_n as given and built a model of $x_n|t_n$, contrast with linear regression, where we took x_n as given and modelled $t_n|x_n$.

Bishop PRML Figure 4.10

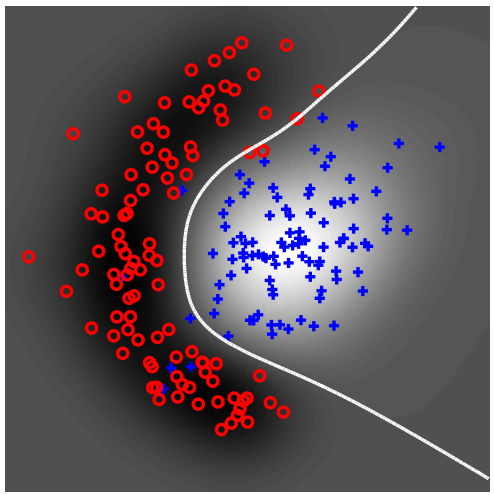
This approach directly generalizes to classification with unequal covariances and multi-class classification.



- Quadratic discriminant analysis: $\Sigma_+ \neq \Sigma_o$, decisison boundary is of form $y(\mathbf{x}) = \mathbf{x}^\top A\mathbf{x} + \omega^\top \mathbf{x} + \omega_o$
- Multi-class: Assign each data-point to class with highest posterior probability (or calculate best assignment from cost-function).

Bishop PRML Figure 4.10

A simple nonlinear classifier can be constructed from kernel density estimates of the probability densities.



- Idea: Once we have an estimate of the class-conditional densities $P(x|t = \pm 1)$, we can construct a rule from $d(x) = P(x|t = +1) - P(x|t = -1)$.
- Use kernel density estimation to estimate $P(x|t = \pm 1)$, i.e. place a 'Gaussian bump' on each data-point:

$$P(x|t = 1) = \frac{1}{Z} \sum_{n_+} \exp\left(\frac{x - x_n}{\sigma}\right)^2 \quad (9)$$

This leads to a classifier of the form

$$d(x) = \sum_n \alpha t_n \exp\left(\frac{x - x_n}{\sigma}\right)^2 \quad (10)$$

Support vector machine with radial basis functions has decision rule

$$d(x) = \sum_n \alpha_n \exp\left(\frac{x - x_n}{\sigma}\right)^2 \quad (11)$$

Summary: One for the price of three.

- Today, you learned about three different algorithms for binary classification with linear decision rules.
- One was based on a hack, the second one on a plausible (but ad-hoc) criterion, and the third one an a probabilistic model of the data.
- All three algorithms are equivalent.
- We showed that the Fisher discriminant and the probabilistic model based on two Gaussians have the same decision criterion. In fact, it can be shown that linear regression has the same weights (Bishop 4.1.5)
- The third motivation had immediate extensions to nonlinear algorithms and multi-class classification, and posterior probabilities.
- Next week, we will learn an algorithm which actually is different, and usually better than the ones discussed today.