Computational Statistics
Linear Classification

Luca Bortolussi

Department of Mathematics and Geosciences
University of Trieste

Office 238, third floor, H2bis
luca@dmi.units.it

Trieste, Winter Semester 2015/2016

Outline

1. Linear Classifiers
2. Logistic Regression
3. Laplace Approximation
4. Bayesian Logistic Regression
5. Constrained Optimisation
6. Support Vector Machines

LINEAR DISCRIMINANT CLASSIFIER

- $y(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b$, decode to class 1 iff $y(\mathbf{x}) > 0$, and to class 0 if $y(\mathbf{x}) < 0$.
- Typically here we use the encoding scheme $t_n \in \{0, 1\}$, but also $t_n \in \{-1, 1\}$ works (different solutions, though).
- Maximum likelihood training like in regression: minimise the sum-of-squares error function

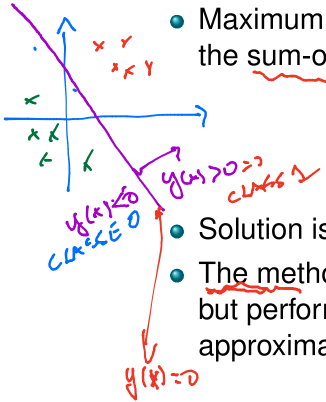
$$E_D(\mathbf{w}) = \frac{1}{2}\sum_i (\mathbf{w}^T\mathbf{x_i} \overset{+b}{} - t_i)^2.$$

$\phi_0 = 1$

$\phi_1 = x_1$

$\vdots$

$\phi_d = x_d$



- Solution is $\boxed{(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{t}}$.
- The method can be extended to k classes (see next slide), but performs poorly in general, because it tries to approximate a probability in $[0, 1]$ with a real number.

MULTI-CLASS STRATEGIES

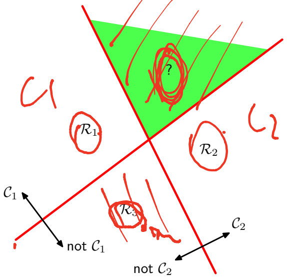
	C_1	C_2	C_3
N	$N/3$	$N/3$	$N/3$

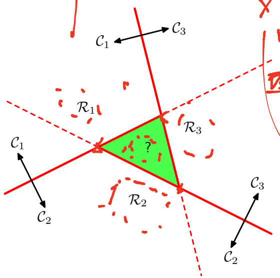
C_1	$C_2 \cup C_3$
$N/3$	$2/3\,N$

- Assume we have a binary classifier. We can train K classifiers, one-versus-the-rest strategy, class C_k versus all other points (unbalanced).
- Alternatively, there is the one-versus-one classifier, trains $K(K-1)/2$ for each pair of classes, decode by majority voting. Both are ambiguous.
- One can train K linear discriminants $y_k(\mathbf{x}) = \mathbf{w_k}^T\mathbf{x} + b_k$ and decode to j such that $y_j(\mathbf{x}) > y_i(\mathbf{x})$ for each $i \neq j$.

C_1 $(0..0\ 1\ 0..0)$ K

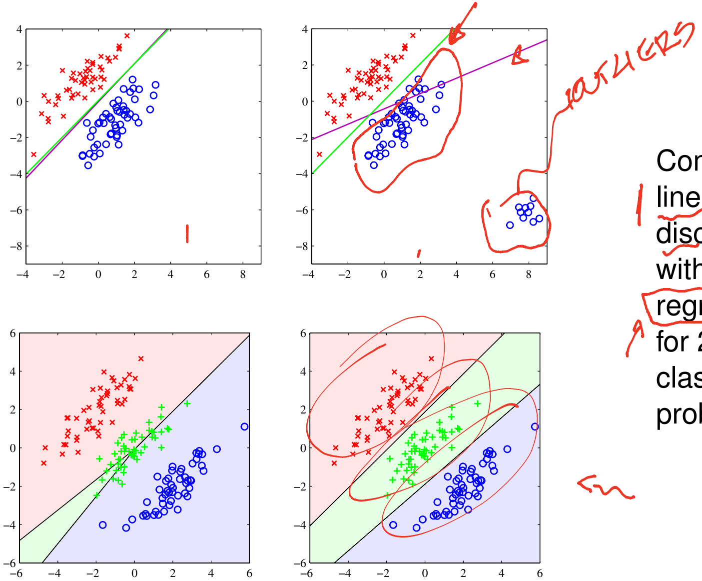
C_1 1
C_2 0
C_K





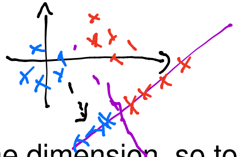
$y_1(x)$
~~$y_2(x)$~~
$y_K(x)$

LINEAR DISCRIMINANT - EXAMPLE

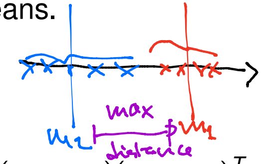


Comparing linear discriminant with logistic regression, for 2 and 3 classes problems.

FISHER'S DISCRIMINANT



- Idea: project data linearly in one dimension, so to separate as much as possible the two classes. The projection is $\boxed{y(\mathbf{x}) = \mathbf{w}^T\mathbf{x}.}$
- Choose the projection that (a) maximises the separation between the two classes, either by maximising the projected class means distance, or by maximising the ratio between between-class and within-class variances.
- $\mathbf{m_i} = 1/N_i \sum_{j\in C_i} \mathbf{x_i}$, $m_i = \mathbf{w}^T\mathbf{m_i}$, class means.
- Between-class variance $\boxed{\mathbf{w}^T\mathbf{S_B}\mathbf{w},}$ $\mathbf{S_B} = (\mathbf{m_2} - \mathbf{m_1})(\mathbf{m_2} - \mathbf{m_1})^T$



- Within-class variance $\mathbf{w}^T\mathbf{S_W}\mathbf{w}$, $\mathbf{S_W} = \sum_{j\in C_1}(\mathbf{x_j} - \mathbf{m_1})(\mathbf{x_j} - \mathbf{m_1})^T + \sum_{j\in C_2}(\mathbf{x_j} - \mathbf{m_2})(\mathbf{x_j} - \mathbf{m_2})^T.$

N_1 N_2

FISHER'S DISCRIMINANT

- Maximise the ratio

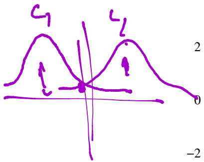
$$J(\mathbf{w}) = \frac{\mathbf{w}^T \mathbf{S_B} \mathbf{w}}{\mathbf{w}^T \mathbf{S_W} \mathbf{w}}$$

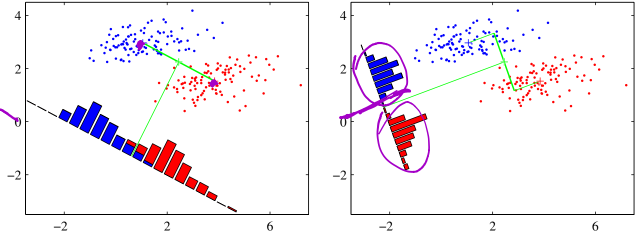
between class variance
within class

- Deriving and setting the derivative to zero, we get $\mathbf{w} \propto \mathbf{S_W}^{-1}(\mathbf{m_2} - \mathbf{m_1})$. cm
- Choose the best y_0 that separates the projected data. Classify to C_1 if $y(\mathbf{x}) \geq y_0$. Idea: approximate the projected class distributions $\boxed{p(y|C_k)}$ as Gaussians and then find y_0 such that $p(y_0|C_1)p(C_1) = p(y_0|C_2)p(C_2)$.

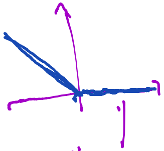
$P(C_i) = N_i/N$

ML estimate to Binomial





THE PERCEPTRON ALGORITHM



$w^T \phi_i t_i$

- For binary classes, proposed by Rosenblatt in 62. Typically one maps the input data in a higher dimensional space $\phi(\mathbf{x_i})$, chooses the coding $t_i \in \{-1, 1\}$, and decodes to C_1 if $y(\mathbf{x}) = f(\mathbf{w}^T \phi(\mathbf{x_i})) \geq 0$, where the activation function is the step function $f(a) = 1$, if $a \geq 0$ and $f(a) = -1$ if $a < 0$.
- A correctly classified pattern satisfies $\mathbf{w}^T \phi(\mathbf{x_i}) t_i \geq 0$. A misclassified pattern instead $\boxed{\mathbf{w}^T \phi(\mathbf{x_i}) t_i} < 0$.
- We pick as error function $E_P(\mathbf{w}) = -\sum_{i \in \mathcal{M}} \mathbf{w}^T \phi(\mathbf{x_i}) t_i$, which generalises the idea of minimising the number of misclassified patterns $\mathcal{M}$.
- Optimise it by stochastic gradient ~~ascend~~ descend:

$w^0, w^1, \ldots, w^n$

w^0 fixed

$n = 0, \ldots, N-1$

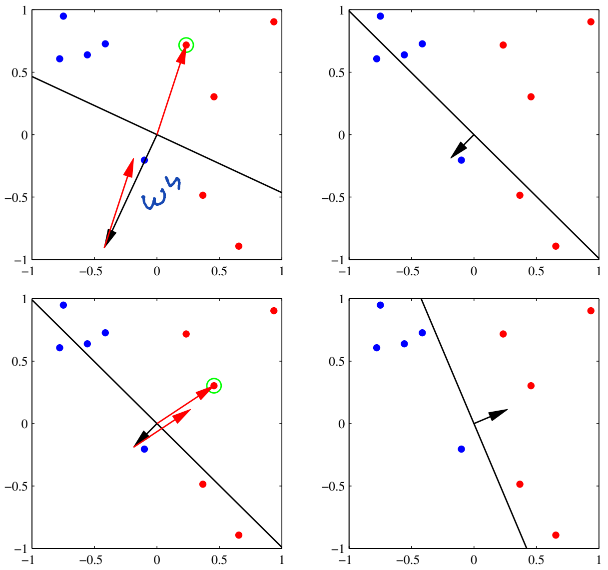
$$\mathbf{w}^{n+1} = \mathbf{w}^n + \eta \boxed{\phi(\mathbf{x_n}) t_n} \mathbb{I}(\mathbf{w}^n \phi(\mathbf{x_n}) t_n < 0)$$

(w^t = ith point in the sequence)

(typically, $\eta = 1$)

- If the data is linearly separable (in the feature space ϕ), then the algorithm converges. (to an hyperplane dep. on w^0)

THE PERCEPTRON ALGORITHM

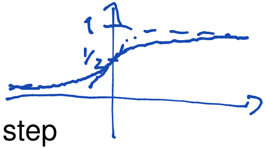


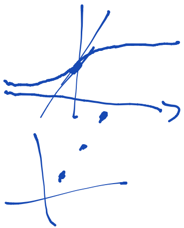
Outline

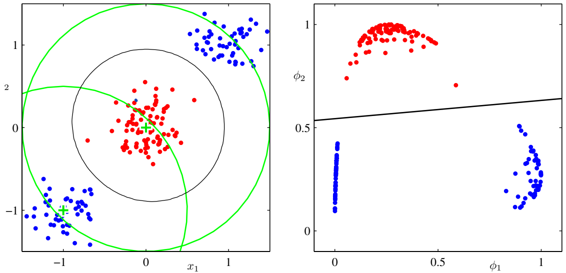
1. Linear Classifiers
2. Logistic Regression
3. Laplace Approximation
4. Bayesian Logistic Regression
5. Constrained Optimisation
6. Support Vector Machines

LOGIT AND PROBIT REGRESSION (BINARY CASE)

- We model directly the conditional class probabilities $p(C_1|\mathbf{x}) = f(\mathbf{w}^T\phi(\mathbf{x}))$, after a (nonlinear) mapping of the features $\phi(\mathbf{x}) = \phi_1(\mathbf{x}), \ldots, \phi_m(\mathbf{x})$. $f: \mathbb{R} \to [0,1]$
- Common choices for f are the logistic or logit function $\sigma(a) = \frac{1}{1+e^{-a}}$ and the probit function $\psi(a) = \int_{-\infty}^{a} \mathcal{N}(\theta|0,1)d\theta$. ~ erf
- We will focus on logistic regression.
- The non-linear embedding is an important step







ESERCIZIO:
SE N punti x_n in $\mathbb{R}^d$, PER QUALE d I PUNTI SONO LIN. SEPARABILI? SOTTO QUALE IPOTESI?

LOGISTIC REGRESSION

y(ϕ) dipende da w (in modo NON-LINEARE)

- We assume $\boxed{p(C_1|\phi)} = \boxed{y(\phi)} = \sigma(\mathbf{w}^T\phi)$ where $\phi = \phi(\mathbf{x})$ and $\phi_i = \phi(\mathbf{x_i})$.
- As $y = y(\phi(\mathbf{x})) \in [0, 1]$ we interpret is as the probability of assigning input $\mathbf{x}$ to class 1, so that the likelihood is

$t_i \in \{0,1\}$

$$\boxed{p(\mathbf{t}|\mathbf{w}) = \prod_{i=1}^{N} y_i^{t_i}(1-y_i)^{1-t_i}}$$

where $y_i = \sigma(\mathbf{w}^T\phi_i)$.

- We need to minimise minus the log-likelihood, i.e.

$$\boxed{E(\mathbf{w})} = -\log p(\mathbf{t}|\mathbf{w}) = -\sum_{i=1}^{N} t_i \log y_i + (1-t_i)\log(1-y_i)$$

NUMERICAL OPTIMISATION

$$\frac{d}{dx}\sigma(x) = \sigma(x)(1-\sigma(x))$$

- The gradient of $E(\mathbf{w})$ is $\nabla E(\mathbf{w}) = \sum_{i=1}^{N}(y_i - t_i)\phi_i$. The equation $\boxed{\nabla E(\mathbf{w}) = 0}$ has no closed form solution, so we need to solve it numerically.
- One possibility is gradient descend. We initialise $\boxed{\mathbf{w}^0}$ to any value and then update it by

$$\boxed{\mathbf{w}^{n+1} = \mathbf{w}^n + \eta\nabla E(\mathbf{w}^n)}$$

where the method converges for η small.
- We can also use stochastic gradient descent for online training, using the update rule for $\mathbf{w}$:

$$\mathbf{w}^{n+1} = \mathbf{w}^n + \eta\nabla_{n+1}E(\mathbf{w}^n),$$

with $\nabla_n E(\mathbf{w}) = (y_n - t_n)\phi_n$

LOGISTIC REGRESSION: OVERFITTING

$w_0 + w_1 x_1 + w_2 x_2 + w_3 x_1 x_2 + w_4 x_1^2 + w_5 x_2^2 = 0$ $\quad 1, x_1, x_2, x_1 x_2, x_1^2, x_2^2$

- If we allocate each point $\mathbf{x}$ to the class with highest probability, i.e. maximising $\sigma(\mathbf{w}^T\phi(\mathbf{x}))$, then the separating surface is an hyperplane in the feature space and is given by the equation $\boxed{\mathbf{w}^T\phi(\mathbf{x}) = 0}$.
- If the data is linearly separable in the feature space, then any separable hyperplane is a solution, and the magnitude of $\mathbf{w}$ tends to go to infinity during optimisation. In this case, the logistic function converges to the Heaviside function.
- To avoid this issue, we can add a regularisation term to $E(\mathbf{w})$, thus minimising $E(\mathbf{w}) + \boxed{\alpha\mathbf{w}^T\mathbf{w}}$.

$w = K \cdot \bar{w}_0$

$\|\bar{w}\| = 1$

$\sigma(K \boxed{w^T x})$