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

LAGRANGE MULTIPLIERS

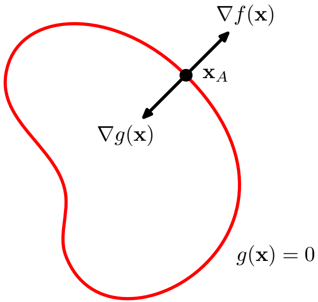
- Suppose we want to optimise $f(\mathbf{x})$ subject to the constraint $g(\mathbf{x}) = 0$.
- $g(\mathbf{x}) = 0$ defines a surface and $\nabla g(\mathbf{x})$ is always orthogonal to it.
- In a point of this surface in which $f(\mathbf{x})$ is optimal, it must hold that $\boxed{\nabla f(\mathbf{x}) = \lambda \nabla g(\mathbf{x})}$, i.e. the projection of $\nabla f(\mathbf{x})$ on the tangent space of the surface is zero.

$\lambda \in \mathbb{R}$

$\frac{\partial L}{\partial x} = \nabla f + \lambda \nabla g = 0$

$\frac{\partial L}{\partial \lambda} = g = 0$

$\nabla_{x\lambda} L = 0$



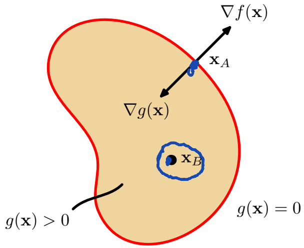
- We can then optimise the Lagrangian function

$$L(\mathbf{x}, \lambda) = f(\mathbf{x}) + \lambda g(\mathbf{x})$$

Deriving w.r.t $\mathbf{x}$ gives the condition on gradients, deriving w.r.t λ the constraint.

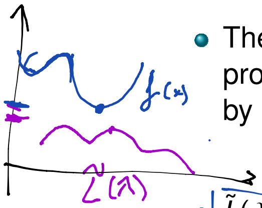
KARUSH-KUHN-TUCKER CONDITIONS

- Suppose we want to optimise $f(\mathbf{x})$ subject to the constraint $g(\mathbf{x}) \geq 0$.
- If an optimum $\mathbf{x}$ satisfies $\boxed{g(\mathbf{x}) > 0}$ (inactive constraint), then $\nabla f(\mathbf{x}) = 0$ and $\lambda = 0$, if instead $g(\mathbf{x}) = 0$ (active constraint), then $\boxed{\nabla f(\mathbf{x}) = -\lambda \nabla g(\mathbf{x}), \lambda > 0}$ because an increase of f cannot bring inside the feasible region.



- In any case $\boxed{\lambda g(\mathbf{x}) = 0}$ for an optimum point.
- We can then optimise the Lagrangian function $L(\mathbf{x}, \lambda) = f(\mathbf{x}) + \lambda g(\mathbf{x})$ subject to $\boxed{\lambda \geq 0, g(\mathbf{x}) \geq 0, \lambda g(\mathbf{x}) = 0,}$ known as the Karush-Kuhn-Tucker (KKT) conditions.
- To minimise $f(\mathbf{x})$, we minimise $L(\mathbf{x}, \lambda) = f(\mathbf{x}) - \lambda g(\mathbf{x})$

THE DUAL FORMULATION

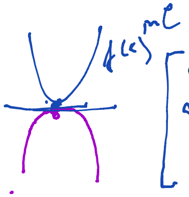


- The dual formulation of the constrained minimisation problem with Lagrangian $\boxed{L(\mathbf{x},\lambda) = f(\mathbf{x}) - \sum_j \lambda_j g_j(\mathbf{x})}$ is given by

$$\boxed{\tilde{L}(\lambda) = \inf_{\mathbf{x}\in\mathcal{D}} L(\mathbf{x},\lambda)}$$

≥ 0 on admiss. points.

- $\tilde{L}(\lambda)$ is a lower bound on $f(\mathbf{x})$. The dual optimisation problem is to maximise $\tilde{L}(\lambda)$ subject to KKT conditions.



- If the original problem is convex (single global optimum), and under regularity conditions on the constraints (e.g. linear), then the solution of the dual gives exactly the minimum of the primal.
- For non-convex problems, there can be a $\boxed{\text{duality gap.}}$

- For quadratic objective functions and linear constraints, the dual objective can be computed easily, because $\partial L(\mathbf{x},\lambda)/\partial\mathbf{x}$ gives a linear system that can be solved to express $\mathbf{x}$ as a function of λ's ✓

Outline

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

KERNEL TRICK FOR CLASSIFICATION

$w = a^T \Phi$

$a^T \phi(x) = \sum_n a_n \underbrace{\phi(x_n)^T \phi(x)}_{K(x, x_n)}$

- The trick works similarly as for regression. Consider class conditionals $p(C_1|\mathbf{x}) = \sigma(\mathbf{w}^T\phi(\mathbf{x}))$.
- We can make the assumption that $\mathbf{w} = \sum_{n=1}^{N} a_n\phi(\mathbf{x_n})$ (this is consistent, as the ML solution will belong to the space spanned by $\phi(\mathbf{x_n})$), thus getting

$$\boxed{p(C_1|\mathbf{x}) = \sigma\left(\sum_{n=1}^{N} \alpha_n k(\mathbf{x}, \mathbf{x_n})\right)}$$

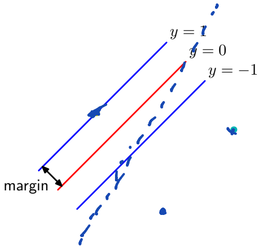
where we define the kernel function $\boxed{k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^T\phi(\mathbf{x}')}$

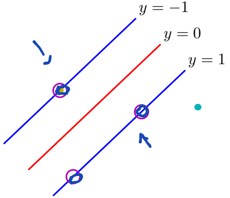
- We can write also $p(C_1|\mathbf{x}) = \sigma(\mathbf{a}^T\mathbf{k}(\mathbf{x}))$. The maximum likelihood solution can be found using the optimisation scheme introduced before.

$\left(K(x, x_1), \; \longrightarrow K(x, x_N)\right)$

Maximum margin classifiers

- We have 2-class data $\mathbf{x_n}, t_n$, with $t_n \in \{-1, 1\}$. We assume for the moment that the data is linearly separable in a feature space after applying the non-linear mapping $\phi(\mathbf{x})$.
- There may be many hyperplanes separating the data. An effective choice is to select the one maximising the margin, i.e. the smallest distance between the separating hyperplane and the data points.
- Only closest data points are needed to determine it.





MAXIMUM MARGIN CLASSIFIERS

- Write $\boxed{y(\mathbf{x}) = \mathbf{w}^T\phi(\mathbf{x}) + b}$.
- The distance between a point and the separating hyperplane $\mathbf{w}^T\phi + b$ is $\boxed{|y(\mathbf{x})|/||\mathbf{w}||}$.
- As we want to classify correctly all points, it will hold that $\boxed{t_n y(\mathbf{x_n}) \geq 0}$ by the choice of t_n encoding.
- Hence, to find the maximum margin, we need to find **w** and b such that:

$$\max_{\mathbf{w},b}\left[\frac{1}{||\mathbf{w}||}\min_n\{t_n\mathbf{w}^T\phi(\mathbf{x_n}) + b\}\right]$$

- The solution is defined up to an arbitrary rescaling of **w** and b, so we can set to 1 the margin, obtaining the constraint

$$\boxed{t_n\mathbf{w}^T\phi(\mathbf{x_n}) + b \geq 1,} \quad n = 1, \ldots, N$$

MAXIMUM MARGIN CLASSIFIERS

- The constraints $t_n \mathbf{w}^T \phi(\mathbf{x_n}) + b \geq 1$ known as the canonical representation. Points for which equality to 1 holds are called active, the others inactive.
- The maximisation above is equivalent to minimise $||\mathbf{w}||^2$:

$$\min_{\mathbf{w},b} \frac{1}{2}||\mathbf{w}||^2$$

subject to canonical constraints. b will be set via the constraints.

- To solve this quadratic program, we introduce a Langrange multiplier a_n for each constraint, resulting in the following Lagrangian

$$L(\mathbf{w}, b, \mathbf{a}) = \frac{1}{2}||\mathbf{w}||^2 - \sum_{n=1}^{N} a_n[t_n \mathbf{w}^T \phi(\mathbf{x_n}) + b - 1]$$

which has to be minimised w.r.t **w** and b, and maximised w.r.t **a**.

The dual formulation of the maximum margin problem

- Starting from the Lagrangian $L(\mathbf{w}, b, \mathbf{a})$ we compute derivatives w.r.t. $\mathbf{w}$ and b and set them to zero, obtaining constraints

$$\mathbf{w} = \sum_n a_n t_n \boxed{\phi(\mathbf{x_n})} \qquad \boxed{0 = \sum_n a_n t_n}$$

- By substituting them in the Lagrangian, we obtain the dual representation

$$\boxed{\tilde{L}(\mathbf{a}) = \sum_{n=1}^{N} a_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} a_n a_m t_n t_m \boxed{k(\mathbf{x_n}, \mathbf{x_m})}}$$

subject to the constraints

$$\boxed{a_n \geq 0,}\; n = 1, \ldots, N; \qquad \boxed{\sum_n a_n t_n = 0}$$

- $k(\mathbf{x_n}, \mathbf{x_m}) = \phi(\mathbf{x_n})^T \phi(\mathbf{x_m})$ is the kernel function.

THE DUAL FORMULATION OF THE MAXIMUM MARGIN PROBLEM

- This optimisation problem can be solved in $O(N^3)$ time. Its main advantage is that it depends on the kernel, not on basis functions, hence it can be applied for more general kernels.
- The prediction for a new point $\mathbf{x}$ is obtained by using the dual formulation of $\mathbf{w}$, giving

$$y(\mathbf{x}) = \sum_n a_n t_n k(\mathbf{x}, \mathbf{x_n}) + b$$

ul small margin

SPARSITY OF THE SOLUTION

- The optimisation problem satisfies the KKT conditions:

$$\boxed{a_n \geq 0;} \quad \boxed{t_n y(\mathbf{x_n}) - 1 \geq 0} \quad \boxed{a_n[t_n y(\mathbf{x_n}) - 1] = 0}$$

- This implies that either $\boxed{t_n y(\mathbf{x_n}) = 1}$ (the vector $\mathbf{x_n}$ is at minimum distance from the margin) or $\underbrace{a_n = 0}$ (it does not contribute to the predictions).
- Let us indicate with $\mathcal{S}$ the set of support vectors.

DETERMINING b

- From any $\mathbf{x_n} \in \mathcal{S}$, by using $\boxed{t_n y(\overset{t}{\mathbf{x_n}}) = 1,}$ we can determine b by solving

$$t_n \sum_{m \in \mathcal{S}} a_m t_m k(\mathbf{x_n}, \mathbf{x_m}) + t_n b = 1$$

- To have a more stable solution, one multiplies by $\tilde{t}_n$, uses $t_n^2 = 1$, and averages for the different support vectors:

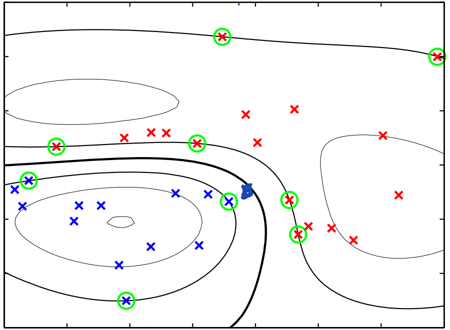
$$b = \boxed{\frac{1}{N_{\mathcal{S}}} \sum_{n \in \mathcal{S}} \left(t_n - \sum_{m \in \mathcal{S}} a_m t_m k(\mathbf{x_n}, \mathbf{x_m}) \right)}$$

EXAMPLE OF SVM

$K(x,y) = \exp\left[-\|x-y\|^2/\lambda^2\right]$

- Example of data linearly separable in the space defined by the Gaussian kernel function.
- Sparsity: only support vectors define the maximum margin hyperplane: moving the other is irrelevant, as far as they remain on the same side.

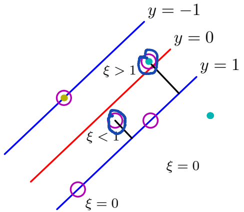
Example of synthetic data from two classes in two dimensions showing contours of constant $y(\mathbf{x})$ obtained from a support vector machine having a Gaussian kernel function. Also shown are the decision boundary, the margin boundaries, and the support vectors.



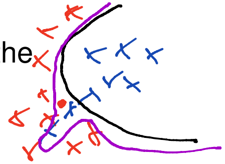
SOFT MARGIN SVM

- If class conditionals overlap, then an exact (non-linear) separation of training data may result in poor generalisation. It is better to allow some training points to be misclassified, by relaxing the constraint $\boxed{t_n y(\mathbf{x_n}) \geq 1}$
- We will do this by introducing N new slack variables $\xi_n \geq 0$, rewriting constraint as $\boxed{t_n y(\mathbf{x_n}) \geq 1 - \xi_n.}$

- For points correctly classified and inside the margin, we have $\xi_n = 0$, while for other points we have $\boxed{\xi_n = |t_n - y(\mathbf{x_n})|}$. It follows that misclassified points will have $\boxed{\xi_n > 1}$, while $\xi_n = 1$ only if a point lies in the separating hyperplane.
- $\boxed{\sum_n \xi_n}$ is an upper bound on misclassified training points.



SOFT MARGIN SVM



- The primal objective function is modified to penalise the number of misclassified points:

$$C\sum_{n=1}^{N}\xi_n + \frac{1}{2}\|\mathbf{w}\|^2$$

- C is a regularisation term: it controls the trade-off between correct classification of training points and model complexity. For $C \to \infty$, we recover the previous SVM.
- The Lagrangian $L(\mathbf{w}, b, \mathbf{a}, \mu)$ is now given by

$$C\sum_{n-1}^{N}\xi_n + \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{n=1}^{N} a_n[t_n\mathbf{w}^T\phi(\mathbf{x_n}) + b - 1 + \xi_n] - \sum_{n=1}^{N}\mu_n\boxed{\xi_n}$$

with $\overline{a_n, \mu_n}$ Lagrange multipliers. We omit the KKT conditions.

SOFT MARGIN SVM: DUAL FORMULATION

- By taking partial derivatives w.r.t $\mathbf{w}$, b, and ξ_n, we obtain the dual formulation:

$$\tilde{L}(\mathbf{a}) = \sum_{n=1}^{N} a_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} a_n a_m t_n t_m k(\mathbf{x_n}, \mathbf{x_m})$$

  which has to satisfy the following box constraints

$$\boxed{0 \leq a_n \leq C,} \; n = 1, \ldots, N; \; \sum_n a_n t_n = 0$$

- In the solution, we can have $a_n = 0$ (points inside the margin , for which $\xi_n = 0$), $0 < a_n < C$ (points on the margin, for which $\xi_n = 0$), or $a_n = C$ (points on the wrong side of the margin, $\boxed{\xi_n > 0}$).
- b can be determined as for the hard margin case, by restricting to support vectors on the margin.

SVM: COMMENTS

- The quadratic problem is convex, hence has a unique minimum, but a classic optimisation can be challenging for large problems (N large). Specialised methods have been developed, that try to decompose the problem into simpler pieces. E.g. Sequential minimal optimisation works by optimising two a_n's at time.
- SVM are hard to generalise to multi-class problems (one-versus-the-rest approach being the typical approach)
- SVM do not have a probabilistic interpretation, and some ad-hoc processing is required.
- SVM can be quite sensitive to outliers (misclassified points deeply inside the other's class region).