Computational Statistics
Linear Regression

Luca Bortolussi

Department of Mathematics and Geosciences
University of Trieste

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

Trieste, Winter Semester 2015/2016

MARGINAL LIKELIHOOD

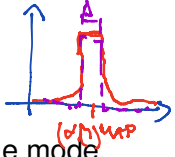
$$p(t^*|x^*, \mathbf{D}) = \int p(t^*|x^*, \mathbf{w}, \alpha, \beta) p(\mathbf{w}|\alpha, \beta, \mathbf{D}) \cdot p(\alpha, \beta|\mathbf{D}) \, d\mathbf{w} \, d\alpha \, d\beta$$

$\int p(\mathbf{D}|\mathbf{w}) p(\mathbf{w}) \, d\mathbf{w}$ → MARGINAL LIKELIHOOD

- The marginal likelihood $p(\mathbf{t}|\alpha, \beta)$, appearing at the denominator in Bayes theorem, can be used to identify good α and β, known as hyperparameters.
- Intuitively, we can place a prior distribution over α and β, compute their posterior, and use this in a fully Bayesian treatment of the regression:

$$p(\alpha, \beta|\mathbf{t}) \propto p(\mathbf{t}|\alpha, \beta) p(\alpha, \beta)$$

FLAT/UNIF.



- If we assume the posterior is peaked around the mode, then we can take the MAP as an approximation of the full posterior for α and β. If the flat is prior, this will boil down to the ML solution.

FLAT PRIOR ⇒ LOG POST = LOG LIKEL + COST

⇒ MAP = ML

MARGINAL LIKELIHOOD

(TYPE II LIKELIHOOD)

EVIDENCE OPTIMISATION

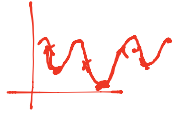
TYPE II MAXIMUM LIKELIHOOD

- Hence we need to optimise the marginal likelihood, which can be computed as:

$$\log p(\mathbf{t}|\alpha,\beta) = \frac{M}{2}\log\alpha + \frac{N}{2}\log\beta - E(\mathbf{m_N}) - \frac{1}{2}\log|\mathbf{S_N}^{-1}| - \frac{N}{2}\log 2\pi$$

with

$$E(\mathbf{m_N}) = \frac{\beta}{2}\|\mathbf{t} - \mathbf{\Phi}\mathbf{m_N}\|^2 + \frac{\alpha}{2}\mathbf{m_N}^T\mathbf{m_N}$$



- This optimisation problem can be solved with any optimisation routine, or with specialised methods, see Bishop.

BAYESIAN MODEL COMPARISON

- Consider $\mathcal{M}_1$ and $\mathcal{M}_2$ two different models, which one is the best to explain the data $\mathcal{D}$?
- In a Bayesian setting, we may place a prior $p(\mathcal{M}_j)$ on the models, and compute the posterior $p(\mathcal{M}_j|\mathcal{D}) = \frac{p(\mathcal{D}|\mathcal{M}_j)p(\mathcal{M}_j)}{\sum_j p(\mathcal{D}|\mathcal{M}_j)p(\mathcal{M}_j)}$.
- As we typically have additional parameters $\mathbf{w}$, the term $p(\mathcal{D}|\mathcal{M}_j)$ is the model evidence/ marginal likelihood.
- The ratio $p(\mathcal{D}|\mathcal{M}_1)/p(\mathcal{D}|\mathcal{M}_2)$ is known as Bayes Factor.

BAYESIAN MODEL COMPARISON

- In Bayesian model comparison, we can take two approaches.
- We can compute the predictive distribution for each model and average it by the posterior model probability

$$p(\mathbf{t}|\mathcal{D}) = \sum_j p(\mathbf{t}|\mathcal{M}_j, \mathcal{D}) p(\mathcal{M}_j|\mathcal{D})$$

- Alternatively, we can choose the model with larger Bayes Factor. This will pick the correct model on average. In fact, the average log Bayes factor (assuming $\mathcal{M}_1$ is the true model) is

$$\int p(\mathcal{D}|\mathcal{M}_1) \log \frac{p(\mathcal{D}|\mathcal{M}_1)}{p(\mathcal{D}|\mathcal{M}_2)} \geq 0$$

Outline

$$w_{ML} = \left(\Phi^T \Phi\right)^{-1} \Phi^T \vec{t}$$

$$w_{ML} \in \mathrm{SPAN}\left(\phi(x_1) \cdots \phi(x_N)\right)$$

$$\Phi = \begin{pmatrix} \phi_0(x_1) & \cdots & \phi_{M-1}(x_1) \\ \vdots & & \vdots \\ \phi_0(x_N) & \cdots & \phi_{M-1}(x_N) \end{pmatrix}$$

DUAL REPRESENTATION

- Consider a regression problem with data $(\mathbf{x_i}, y_i)$, and a linear model $\mathbf{w}^T\boldsymbol{\phi}(\mathbf{x})$.
- We can restrict the choice of $\mathbf{w}$ to the linear subspace spanned by $\boldsymbol{\phi}(\mathbf{x_1}), \ldots, \boldsymbol{\phi}(\mathbf{x_N})$, as any $\mathbf{w}_\perp$ othogonal to this subspace will give a contribution $\mathbf{w}_\perp{}^T\boldsymbol{\phi}(\mathbf{x_i}) = 0$ on input points:

$$\mathbf{w} = \sum_{j=0}^{N} a_j \boldsymbol{\phi}(\mathbf{x_j})$$

- $\mathbf{a}$ are known as the dual variables
- By defining the kernel $k(\mathbf{x_i}, \mathbf{x_j}) := \boldsymbol{\phi}(\mathbf{x_i})^T\boldsymbol{\phi}(\mathbf{x_j})$, we can write

$$\mathbf{w}^T\boldsymbol{\phi}(\mathbf{x_i}) = \mathbf{a}^T\mathbf{K^i}$$

Where $\mathbf{K^i}$ is the ith column of the Gram matrix $\mathbf{K}$, $K_{ij} = k(\mathbf{x_i}, \mathbf{x_j})$.

DUAL REGRESSION PROBLEM

- In the dual variables, we have to optimise the following regression equation

$$E_d(\mathbf{a}) + \lambda E_W(\mathbf{a}) = \sum_{i=1}^{N} (t_i - \mathbf{a}^T \mathbf{k}_i)^2 + \lambda \mathbf{a}^T \mathbf{K} \mathbf{a}$$

- By deriving w.r.t $\mathbf{a}$ and setting the gradient to zero, we obtain the solution

$$\hat{\mathbf{a}} = (\mathbf{K} + \lambda \mathbf{I})^{-1} \mathbf{t}$$

- At a new input $\mathbf{x}^*$, the prediction will then be

$$y(\mathbf{x}^*) = \mathbf{k}_*^T (\mathbf{K} + \lambda \mathbf{I})^{-1} \mathbf{t}$$

with $\mathbf{k}_*^T = (k(\mathbf{x}^*, \mathbf{x}_1), \ldots, k(\mathbf{x}^*, \mathbf{x}_N))$

THE KERNEL TRICK

- The dual objective function depends only on the scalar product of input vectors
- We can replace the Euclidean scalar product with *any* (non-linear) scalar product
- This is usually obtained by giving directly a non-linear *kernel* function $k(\mathbf{x}_i, \mathbf{x}_j)$ (*kernel trick*)
- This enables us to work with more general set of basis functions, even countable. See Gaussian processes.
- The same dual procedure applies to other algorithms, notably linear classification and SVMs

THE KERNEL TRICK

- The dual objective function depends only on the scalar product of input vectors
- We can replace the Euclidean scalar product with *any* (non-linear) scalar product
- This is usually obtained by giving directly a non-linear *kernel* function $k(\mathbf{x}_i, \mathbf{x}_j)$ (*kernel trick*)
- This enables us to work with more general set of basis functions, even countable. See Gaussian processes.
- The same dual procedure applies to other algorithms, notably linear classification and SVMs
- The computational cost to solve the primal problem is $O(M^3)$, while the dual costs $O(N^3)$. They can be both prohibitive is N and M are large. In this case, one can optimise the log likelihood directly, using gradient based methods.

$\phi_0 (= 1) \ldots \phi_{M-1} : \mathbb{R}^K \to \mathbb{R}$ $\quad M > K$

$y(x) = w^T \phi(x) \Rightarrow t \sim \mathcal{N}(y(x), \beta^{-1})$

ML $\Rightarrow$ quadratico unconstrained

Bayesiano $p(w)$ calcolando $p(w|D)$

Formulazione duale con: kernel $k(x, x')$

Computational Statistics
Linear Classification

Luca Bortolussi

Department of Mathematics and Geosciences
University of Trieste

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

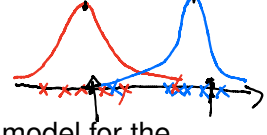
Trieste, Winter Semester 2015/2016

Outline

INTRODUCTION

- Data: $\mathbf{x_i}, t_i$. Output are discrete, either binary or multiclass (K classes), and are also denoted by y_i. Classes are denoted by $C_1, \ldots, C_K$.
- Discriminant function: we construct a function $f(\mathbf{x}) \in \{1, \ldots, K\}$ associating with each input a class.
- Generative approach: We consider a prior over classes, $p(C_k)$, and the class-conditional densities $p(\mathbf{x}|C_k)$, from a parametric family. We learn class-conditional densities from data, and then compute the class posterior.

$$p(C_k|\mathbf{x}) = \frac{p(\mathbf{x}|C_k)p(C_k)}{p(\mathbf{x})}$$



- Discriminative approach: we learn directly a model for the class posteriori $p(C_k|\mathbf{x})$, typically as $p(C_k|\mathbf{x}) = f(\mathbf{w}\boldsymbol{\phi}(\mathbf{x}))$. f is called an activation function (and f^{-1} a link function).

ENCODING OF THE OUTPUT

- For a binary classification problem, usually we choose $t_n \in \{0, 1\}$. The interpretation is that of a "probability" to belong to class C_1.
- In some circumstances (perceptron, SVM), we will prefer the encoding $t_n \in \{-1, 1\}$.
- For a multiclass problem, we usually stick to a boolean encoding: $\mathbf{t_n} = (t_{n,1}, \ldots, t_{n,K})$, with $t_{n,j} \in \{0, 1\}$, and t_n is in class k if and only if $t_{n,k} = 1$ and $t_{n,j} = 0$, for $j \neq k$.

$(1,0,0) \sim C_1$
$(0,1,0) \sim C_2$
$(0,0,1) \sim C_3$