Bayesian Machine Learning - Lecture 2

Guido Sanguinetti

Institute for Adaptive and Neural Computation
School of Informatics
University of Edinburgh
gsanguin@inf.ed.ac.uk

February 24, 2015

Today's lecture

1. Linear regression models
2. Bayesian linear regression
3. Generalised linear models
4. Convex duality and the kernel trick

Linear regression

- Suppose our input-output pairs are real numbers and we have observations $(x_i, y_i) \quad i = 1, \ldots, N$
- The simplest model (hypothesis) is of a linear relationship between input and output and of homogeneous, independent and identically distributed (i.i.d.) Gaussian noise
- We write this as

$$p(y|x) = \mathcal{N}\left(wx + b, \sigma^2\right)$$

or, equivalently

$$y = wx + b + \epsilon \quad \epsilon \sim \mathcal{N}\left(0, \sigma^2\right)$$

- This is called a *regression* model; the (vector) $\mathbf{w}$ are the *weights* and the intercept b the *bias*
- Let's work out the Maximum Likelihood estimate of the parameters $\mathbf{w}, b, \sigma^2$

More complicated regression models

- We may want to discard the hypothesis of a linear relationship, but assume that the input-output relationship can be expressed as a combination of fixed basis functions, e.g.

$$y = \sum_{n=1}^{M} w_n \sin(n\omega x) + \epsilon \quad \epsilon \sim \mathcal{N}\left(0, \sigma^2\right)$$

- The parameters can be found exactly as before by maximum likelihood.
- What happens when the number of basis functions becomes bigger than the number of data points?
- **Overfitting!**

More complicated regression models

- We may want to discard the hypothesis of a linear relationship, but assume that the input-output relationship can be expressed as a combination of fixed basis functions, e.g.

$$y = \sum_{n=1}^{M} w_n \sin(n\omega x) + \epsilon \quad \epsilon \sim \mathcal{N}\left(0, \sigma^2\right)$$

- The parameters can be found exactly as before by maximum likelihood.
- What happens when the number of basis functions becomes bigger than the number of data points?
- **Overfitting!**

Regularised regression models

- Technically, the objective function stops being (strongly) convex when overfitting
- This can be fixed by biasing the solution adding a (convex) penalty term
- Popular choices for penalties are the L^2 penalty $\lambda\|\mathbf{w}\|^2$ (ridge regression) or the L^1 penalty $\lambda \sum_i |w_i|$ (LASSO)
- These are called regularised regression models; as the penalty does not depend on the number of observations, the bias becomes vanishingly small for large data
- LASSO regression is particularly popular as it returns sparse solutions, i.e. it sets some w_i to 0

- Regularisation works by biasing
- One way to bias estimators is to have prior beliefs and being Bayesian
- Let's assume the regression weights have a Gaussian prior $\mathbf{w} \sim \mathcal{N}(0, I)$ and that the bias is zero
- Exercise I: work out the posterior over the weights
- Exercise II: work out the predictive distribution $p(y_{new}|\mathbf{x}_{new}, \mathcal{D})$ with $\mathcal{D}$ the previously observed data

Important observations

- Imposing a Gaussian prior over the weights is somewhat similar to L^2 regularisation (same objective function as the MAP): this correspondence priors- regularisation can be extended
- However, a Bayesian analysis is totally different from a regularised regression one!
- We were only able to compute posterior distributions due to the very special functional form of the distributions (both normal)
- Pairs of prior/ likelihood for which the posterior is analytically computable are called *conjugate*
- Examples: Beta/ Binomial, Dirichlet/ Multinomial, Normal/ InverseGamma (on the variance), Poisson/ Gamma

General responses

- Discriminative learning models directly the conditional probability of the response given the input, $p(y|\mathbf{x})$
- In the regression case, the conditional is often taken to be a Gaussian probability density function
- More general models, where the probability $p(y|\mathbf{x})$ is non-Gaussian, are possible
- When the model takes the form $p(y|\mathbf{x}) = f(\mathbf{w}^T\mathbf{x})$ with f a nonlinear function, these are called *generalised linear models*
- Generally intractable from the Bayesian point of view, estimation done by ML

Poisson regression

- When the output variable is integer (unbounded), we may want to model its dependence on the input via a Poisson distribution

$$p(y = k|\mathbf{x}) = \frac{\mu(\mathbf{x})^k}{k!} \exp(-\mu(\mathbf{x}))$$

  where $\mu(x)$ is the mean of the Poisson distribution and is assumed to be a function of the input variables (usually $exp(\mathbf{w}^T\mathbf{x})$ to ensure positivity)
- Useful e.g. in bioinformatics applications using Next Generation Sequencing

Logistic regression

- Simplest binary classifier:

$$p(y = 1|\mathbf{x}) = \frac{1}{1 + \exp(-\mathbf{w}^T\mathbf{x})}$$

- Can be written compactly as

$$p(y|\mathbf{x}) = \sigma(y\mathbf{w}^T\mathbf{x}) = \frac{1}{1 + \exp(-y\mathbf{w}^T\mathbf{x})}$$

- Often interested in a regularised version, with log likelihood given by

$$\mathcal{L} = -\sum_i \log\left(1 + \exp(-y_i\mathbf{w}^T\mathbf{x}_i)\right) - \lambda\mathbf{w}^T\mathbf{w}$$

Maximum likelihood estimation

- The regularised likelihood can be optimised using gradient based methods
- The gradient of the log likelihood is given by

$$\nabla_{\mathbf{w}}\mathcal{L} = \sum_i \sigma(y_i \mathbf{w}^T \mathbf{x}_i) y_i \mathbf{x}_i - 2\lambda \mathbf{w}$$

- The *Hessian* of the log-likelihood is negative definite$\longrightarrow$ convex problem
- Various efficient tools available such as (iterative) *Newton's method* (complexity $O(d^3)$ where d is the dimensionality of the input)

Aside: Newton's method

- Suppose we have a parabola (or in more dim a paraboloid)
- This has three parameters which can be determined by knowing the value at one point, its first and second derivative
- Hence we get an explicit relation between the minimum and the derivatives
- If the function is not a parabola, this won't yield the minimum, but will decrease the function$\longrightarrow$ iterate to convergence

Dual optimisation

- General principle in optimisation
- Given a constrained optimisation problem

$$\text{maximize} \quad f(\mathbf{w}) \quad s.t. \quad f_0(\mathbf{w}) = 0$$

  there exists a function $F(\mathbf{w}, \alpha), \quad \alpha \in \mathbb{R}$ that upper bounds f for every value of α
- If the original problem has a single maximum, then the bound is *tight*, so $\exists$ a unique α_0 such that $F(\mathbf{w}, \alpha_0) = f(\mathbf{w})$ (of course this is the minimum of F for fixed $\mathbf{w}$)
- Dual optimisation consists of working directly with the upper bound: sometimes easier and the core of the kernel trick

Dual problem for logistic regression

- Consider again the logistic regression objective

$$\mathcal{L}(\mathbf{w}) = -\sum_i \log\left(1 + \exp(y_i \mathbf{w}^T \mathbf{x}_i)\right) - \lambda \mathbf{w}^T \mathbf{w}$$

- It can be shown that $\forall \alpha \in [0, 1]$

$$-\log\left(1 + \exp(\xi)\right) \leq \alpha\xi + \alpha \log \alpha + (1 - \alpha) \log(1 - \alpha)$$

- Hence, $\forall \mathbf{w}, \alpha$

$$\mathcal{L}(\mathbf{w}) \leq l(\mathbf{w}, \alpha) = \sum_i \alpha_i y_i \mathbf{w}^T \mathbf{x}_i + \alpha \log \alpha + (1-\alpha) \log(1-\alpha) - \lambda \mathbf{w}^T \mathbf{w} \tag{1}$$

Dual problem for logistic regression

- Equation (1) gives a function with a very simple dependence on $\mathbf{w}$ (quadratic)
- By duality, we can maximise (1) for fixed α to get

$$\mathbf{w}(\boldsymbol{\alpha}) = \sum_i \lambda^{-1} \alpha_i y_i \mathbf{w}^T \mathbf{x}_i$$

giving a (new) objective function

$$J(\boldsymbol{\alpha}) = \lambda^{-1} \sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_j^T \mathbf{x}_i - \sum_i [\alpha_i \log \alpha_i + (1 - \alpha_i) \log(1 - \alpha_i)] \tag{2}$$

The kernel trick

- The objective function in (2) 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 defined through a non-linear *kernel* function $k(\mathbf{x}_i, \mathbf{x}_j)$ (*kernel trick*)
- This enables us to extend the algorithm to objects which are not numeric but have a concept of distance (e.g. sequences)
- The same dual procedure applies to other algorithms, notably SVMs