Machine Learning
Exercise 4

Marc Toussaint
Machine Learning & Robotics lab, U Stuttgart
Universitätsstraße 38, 70569 Stuttgart, Germany

May 16, 2014

1 Labelling a time series

Assumes we are in a clinical setting. Several sensors measure different things of a patient, like heart beat rate, blood pressure, EEG signals. These measurements form a vector $x_t \in \mathbb{R}^n$ for every time step $t = 1, .., T$.

A doctor wants to annotate such time series with a binary label that indicates whether the patient was asleep. Your job is to develop a ML algorithm to automatically annotate such data. For this assume that the doctor has already annotated several time series of different pations: you are given the training data set

$$D = \{(x^i_{1:T}, y^i_{1:T})\}_{i=1}^n \;, \quad x^i_t \in \mathbb{R}^n,\ y^i_t \in \{0,1\}$$

where the superscript i denotes the data instance, and the subscript t the time step (e.g., each step could correspond to 10 seconds).

Develop a ML algorithms to do this job. First specify the model formally in all details (features, parameters, optimization objective). Then detail the algorithm to compute optimal parameters in pseudo code, being as precise as possible in all respects. Finally, don't forget to sketch an algorithm to actually do the annotation given an input $x^i_{1:T}$ and optimized parameters.

(No need to actually implement.)

2 CRFs and logistic regression

Slide 03-26 summarizes the core equations for CRFs.

a) Confirm the given equations for $\partial_\beta Z(x, \beta)$ and $\partial^2_\beta Z(x, \beta)$ (i.e., derive them from the definition of $Z(x, \beta)$).

b) Derive from these CRF equations the special case of logistic regression. That is, show that the gradient and Hessian given on slide 03-16 can be derived from the general expressions for $\partial_\beta Z(x, \beta)$ and $\partial^2_\beta Z(x, \beta)$. (The same is true for the multi-class case on slide 03-22.)

3 Kernel logistic regression

The "kernel trick" is generally applicable whenever the "solution" (which may be predictive function $f^{\text{rigde}}(x)$, or the discriminative function, or principal components...) can be written in a form that only uses the kernel function $k(x, x')$, but never features $\phi(x)$ or parameters β explicitly.

Derive a kernelization of Logistic Regression (slide 03-16). That is, think about how you could perform the Newton iterations based only on the kernel function $k(x, x')$.

Tips: Reformulate the Newton iterations

$$\beta \leftarrow \beta - (\boldsymbol{X}^\top W \boldsymbol{X} + 2\lambda I)^{\text{-1}} \,[\boldsymbol{X}^\top(\boldsymbol{p} - \boldsymbol{y}) + 2\lambda I \beta] \tag{1}$$

$$\tag{2}$$

using the two Woodbury identities

$$(X^\top W X + A)^{\text{-}1} X^\top W = A^{\text{-}1} X^\top (X A^{\text{-}1} X^\top + W^{\text{-}1})^{\text{-}1} \tag{3}$$

$$(X^\top W X + A)^{\text{-}1} = A^{\text{-}1} - A^{\text{-}1} X^\top (X A^{\text{-}1} X^\top + W^{\text{-}1})^{\text{-}1} X A^{\text{-}1} \tag{4}$$

Note that you'll need to handle the $\boldsymbol{X}^\top(\boldsymbol{p} - \boldsymbol{y})$ and $2\lambda I\beta$ differently.

Then think about what is actually been iterated in the kernalized case: surely we cannot iteratively update the optimal parameters, because we want to rewrite equations to never touch β or $\phi(x)$ explicitly.