Machine Learning
Exercise 4

Marc Toussaint
TAs: Janik Hager, Philipp Kratzer
Machine Learning & Robotics lab, U Stuttgart
Universitätsstraße 38, 70569 Stuttgart, Germany

May 7, 2019

At the bottom several 'extra' exercises are given. These are not part of the tutorial and only for your interest.

(BSc Data Science students may skip preparing exercise 2.)

(There were questions about an API documentation of the C++ code. See `https://github.com/MarcToussaint/rai-maintenance/blob/master/help/doxygen.md`.)

1 Logistic Regression (6 Points)

On the course webpage there is a data set `data2Class.txt` for a binary classification problem. Each line contains a data entry (x, y) with $x \in \mathbb{R}^2$ and $y \in \{0, 1\}$.
a) Compute the optimal parameters β and the mean neg-log-likelihood $-\frac{1}{n} \log L(\beta)$ for logistic regression using linear features. Plot the probability $P(y = 1 \,|\, x)$ over a 2D grid of test points. (4 P)

- Recall the objective function, and its gradient and Hessian that we derived in the last exercise:

$$L(\beta) = -\sum_{i=1}^n \log P(y_i \,|\, x_i) + \lambda \|\beta\|^2 \tag{1}$$

$$= -\sum_{i=1}^n \Big[y_i \log p_i + (1 - y_i) \log[1 - p_i] \Big] + \lambda \|\beta\|^2 \tag{2}$$

$$\nabla L(\beta) = \frac{\partial L(\beta)}{\partial \beta}^\top = \sum_{i=1}^n (p_i - y_i)\ \phi(x_i) + 2\lambda I \beta = X^\top (p - y) + 2\lambda I \beta \tag{3}$$

$$\nabla^2 L(\beta) = \frac{\partial^2 L(\beta)}{\partial \beta^2} = \sum_{i=1}^n p_i (1 - p_i)\ \phi(x_i)\ \phi(x_i)^\top + 2\lambda I = X^\top W X + 2\lambda I \tag{4}$$

$$\text{where} \quad p(x) := P(y{=}1 \,|\, x) = \sigma(\phi(x)^\top \beta)\ , \quad p_i := p(x_i)\ , \quad W := \text{diag}(p \circ (1 - p)) \tag{5}$$

- Setting the gradient equal to zero can't be done analytically. Instead, optimal parameters can quickly be found by iterating Newton steps: For this, initialize $\beta = 0$ and iterate

  iterate as long as the simultaneous Beta's become very small

$$\beta \leftarrow \beta - (\nabla^2 L(\beta))^{-1}\ \nabla L(\beta)\ . \tag{6}$$

  You usually need to iterate only a few times ($\sim$10) til convergence.

- As you did for regression, plot the discriminative function $f(x) = \phi(x)^\top \beta$ or the class probability function $p(x) = \sigma(f(x))$ over a grid.

Useful gnuplot commands:

```
splot [-2:3][-2:3][-3:3.5] 'model' matrix \
   us ($1/20-2):($2/20-2):3 with lines notitle
plot [-2:3][-2:3] 'data2Class.txt' \
   us 1:2:3 with points pt 2 lc variable title 'train'
```

b) Compute and plot the same for quadratic features. (2 P)

2 Structured Output (4 Points)

(Warning: This is one of these exercises that do not have "one correct solution".)

Consider data of tuples (x, y_1, y_2) where

- $x \in \mathbb{R}^d$ is the age and some other features of a spotify user
- $y_1 \in \mathbb{R}$ quantifies how much the user likes HipHop
- $y_2 \in \mathbb{R}$ quantifies how much the user likes Classic

Naively one could train separate regressions $x \mapsto y_1$ and $x \mapsto y_2$. However, it seems reasonable that somebody that likes HipHop might like Classic a bit less than average (anti-correlated).

a) Properly define a *representation* and *objective* for modelling the prediction $x \mapsto (y_1, y_2)$. (2 P)

Tip: Discriminative functions $f(y, x)$ can not only be used to define a class prediction $F(x) = \text{argmax}_y f(y, x)$, but equally also continuous predictions where $y \in \mathbb{R}$ or $y \in \mathbb{R}^O$. How can you setup and parameterize a discriminative function for the mapping $x \mapsto (y_1, y_2)$?

b) Assume you would not only like to make a deterministic prediction $x \mapsto (y_1, y_2)$, but map x to a probability distribution $p(y_1, y_2|x)$, where presumably y_1 and y_2 would be anti-correlated. How can this be done? (2 P)

c) Extra: Assume that the data set would contain a third output variable $y_3 \in \{0, 1\}$, e.g. y_3 may indicate whether the user pays for music. How could you setup learning a model $x \mapsto (y_1, y_2, y_3)$?

d) Extra: General discussion: Based on this, how are regression, classification, and conditional random fields related?

3 Extra: Discriminative Function in Logistic Regression

Logistic Regression (slide 03:19) defines class probabilities as proportional to the exponential of a discriminative function:

$$P(y|x) = \frac{\exp f(x, y)}{\sum_{y'} \exp f(x, y')}$$

Prove that, in the binary classification case, you can assume $f(x, 0) = 0$ without loss of generality.

This results in

$$P(y = 1|x) = \frac{\exp f(x, 1)}{1 + \exp f(x, 1)} = \sigma(f(x, 1)).$$

(Hint: first assume $f(x, y) = \phi(x, y)^\top \beta$, and then define a new discriminative function f' as a function of the old one, such that $f'(x, 0) = 0$ and for which $P(y|x)$ maintains the same expressibility.)

4 Extra: Special cases of CRFs

Slide 03:31 summarizes the core equations for CRFs.

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

b) Binary logistic regression is clearly a special case of CRFs. Sanity check that the gradient and Hessian given on slide 03:20 can alternatively be derived from the general expressions for $\nabla_\beta L(\beta)$ and $\nabla_\beta^2 L(\beta)$ on slide 03:31. (The same is true for the multi-class case .)

c) Proof that ordinary ridge regression is a special case of CRFs if you choose the discriminative function $f(x, y) = -y^2 + 2y\phi(x)^\top \beta$.