582744 Advanced Course in Machine Learning

Exercise 1 — Due Mar 21, 23:55 AM

Rules:

1. Return your solutions in Moodle by the deadline.
2. The submission consists of two parts: (a) A single PDF file containing your answers to all questions. (b) A single file containing your code (either a single plain text source code or a compressed file). Do not include datasets, plots etc in this file, only the code.
3. Please typeset your work using appropriate software such as LaTeX. However, there is no need to typeset the pen and paper answers – you can also include a scanned hand-written version.
4. Pay attention to how you present the results. Be concise.
5. Spotted a mistake? Something is unclear? Ask for clarifications in Moodle.

This set of exercises is due on Tuesday Mar 21, before 23:55 AM.

1 Your background (1 pt)

Explain briefly your background and level of knowledge for the prerequisities. Note that your subjective evaluations here have no impact on the course grade, and will only be needed for planning the course and for statistical purposes.

(a) Which other courses on machine learning and statistical modeling you have taken?

(b) How familiar you are with statistics? Estimate your knowledge on a scale from 0-5

(c) How strong you are in linear algebra? Use the same scale.

(d) Why did you take this course?

(e) What would you like to learn during the course? What would you want us to focus on?

2 Expectations (3 pts)

(a) Show that the variance of a sum of two random variables is given by $\text{var}(X+Y) = \text{var}(X) + \text{var}(Y) + 2\text{cov}(X,Y)$. (Exercise 2.3 in MLaPP)

(b) Compute the mean (expected value) of the gamma distribution with probability density function

$$p_X(x|\alpha,\beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x},$$

where $\Gamma(\alpha)$ is the gamma function. Perform the necessary calculations with pen and paper.

Hints: Direct integration might be tricky, but remember that integral of any density function must be one. The following result for gamma functions might be useful: $\Gamma(\alpha+1) = \alpha\Gamma(\alpha)$

(c) Assume a D-dimensional random variable X follows a probability density

$$p_X(\mathbf{x}) = \frac{1}{(2\pi)^{D/2}} \exp\left(-\frac{1}{2}\mathbf{x}^T\mathbf{x}\right).$$

Now assume that $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{b}$, where $\mathbf{A}$ is a square matrix and $\mathbf{b}$ is a D-dimensional vector. What is the mean of Y? How about the covariance of Y? Write down the full probability density of the resulting distribution.

3 Eigen-value decomposition (programming exercise) (2 pts)

This exercise is primarily intended for verying that you are comfortable operating with numerical matrices in your favorite programming environment, and that you can create simple plots for visualizing the results. You are free to choose the environment within reason, but you can use this exercise to guide the selection: You should pick a language so that this kind of problems can be solved with only a couple of lines of code. One good choice is Python, for which you can consult http://cs231n.github.io/python-numpy-tutorial/ for a good tutorial on numeric computation. You can also check http://hyperpolyglot.org/numerical-analysis for side-by-side reference for numerical computation in various languages.

Principal component analysis (PCA) is a linear method used for dimensionality reduction. Given a set of N data vectors collected in a matrix $\mathbf{X} \in \mathbb{R}^{D\times N}$ the goal is to learn a set of L orthonormal basis vectors $\mathbf{W} \in \mathbb{R}^{D\times L}$ such that we get the best possible L-dimensional representation for our data vectors by projecting them onto that basis with $\mathbf{z}_n = \mathbf{W}^T\mathbf{x}_n$.

The optimality criterion is here the reconstruction error. The reconstruction of a sample is obtained by projecting the low-dimensional vector $\mathbf{z}_n$ back to the observation space with $\widehat{\mathbf{x}}_n = \mathbf{W}\mathbf{z}_n$, and we use squared error

$$||\mathbf{X} - \widehat{\mathbf{X}}||_F^2 = ||\mathbf{X} - \mathbf{W}\mathbf{Z}^T||_F^2 = \sum_{i=1}^{N}\sum_{j=1}^{D}(\mathbf{x}_{ij} - \widehat{\mathbf{x}}_{ij})^2$$

as the loss function. The solution $\mathbf{W}$ is found by computing the first L eigenvectors of the covariance matrix of $\mathbf{X}$ (see section 12.2.1). Perform the following steps to experiment with PCA.

(a) Read in a small data set $\mathbf{X}$ provided in Moodle. The data is given in a CSV-file where each row corresponds to a single data vector (that is, one column of $\mathbf{X}$). What are N and D here?

(b) Compute the covariance matrix of $\mathbf{X}$, and then compute its eigenvalues and eigenvectors. Write down the eigenvalues in descending order.

(c) Plot the data projected onto the first two eigenvectors. "First" refers here to the ones corresponding to the largest eigenvalues.

(d) Plot the reconstruction error as a function of the reduced dimensionality L, starting from $L = 2$ and going to $L = D$. How does this relate to the eigenvalues computed in step (c)?

What to report: Provide a clean report that includes the plots for (c) and (d) with understandable axis labels and titles, complemented with written answers for the questions.

4 Derivatives, gradients and all that (2pts)

Logistic regression is a model for binary classification. It states that the probability of the positive class label is given by $s(\boldsymbol{\theta}^T\mathbf{x})$, where $s(z)$ is the *logistic function* (or sigmoid function):

$$s(z) = \frac{1}{1 + e^{-z}}.$$

The logarithmic loss for a single data point (input $\mathbf{x} \in \mathbb{R}^D$ and output $y \in [0, 1]$) is

$$L((\mathbf{x}, y), \boldsymbol{\theta}) = -y \log s(\boldsymbol{\theta}^T\mathbf{x}) - (1 - y)\log(1 - s(\boldsymbol{\theta}^T\mathbf{x})).$$

Where does this come from?

Given the loss function above, compute

(a) The gradient $\nabla L((x, y), \boldsymbol{\theta}) = [\frac{\partial L}{\partial \theta_1}, \ldots, \frac{\partial L}{\partial \theta_d}]^T$.

(b) The Hessian matrix containing all second derivatives

$$H(L((\mathbf{x}, y), \boldsymbol{\theta})) = \begin{bmatrix} \frac{\partial^2 L}{\partial\theta_1\partial\theta_1} & \frac{\partial^2 L}{\partial\theta_1\partial\theta_2} & \cdots & \frac{\partial^2 L}{\partial\theta_1\partial\theta_D} \\ \frac{\partial^2 L}{\partial\theta_2\partial\theta_1} & \frac{\partial^2 L}{\partial\theta_2\partial\theta_2} & \cdots & \frac{\partial^2 L}{\partial\theta_2\partial\theta_D} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 L}{\partial\theta_D\partial\theta_1} & \frac{\partial^2 L}{\partial\theta_D\partial\theta_2} & \cdots & \frac{\partial^2 L}{\partial\theta_D\partial\theta_D} \end{bmatrix}.$$

(c) What are the numerical values for the gradient and Hessian if $\boldsymbol{\theta} = [1, 1]$, $\mathbf{x} = [-1, 2]$ and $y = 1$?

Hints: Remember the chain rule of derivation. It probably helps if you first compute the derivative $\frac{ds(z)}{dz}$ of the sigmoid function for an arbitrary input and simplify the result as much as possible.