Machine Learning	Dr. Mario Fritz and Dr. Björn Andres
Exercise Sheet 5 - 01.12.2014	due: 08.12.2014

Exercise 14 - Dual Coordinate Ascent for Linear SVM

In this exercise we implement coordinate ascent method for solving the dual of the soft-margin SVM problem. For simplicity, we restrict to the case where the offset b is fixed at zero. The dual problem in this case is given by

$$\max_{\alpha \in \mathbb{R}^n} \ \Psi(\alpha) \tag{1}$$
$$\text{subject to: } 0 \leq \alpha_i \leq \frac{C}{n}, \quad \forall i = 1, \ldots, n$$

where $\Psi(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1,j=1}^{n} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle$ and $x_i \in \mathbb{R}^d$ is the i^{th} training example and $y_i \in \{-1, +1\}$ is its label.

a. (**Bonus 2 Points**) Derive the dual formulation given above for the soft-margin SVM problem when the offset $b = 0$.

b. Coordinate descent/ascent is an optimization method which can be applied when the objective is differentiable and the optimization variables are not coupled by the constraints. This method solves a sequence of smaller subproblems where the objective is optimized over a single coordinate while fixing the values for the other coordinates. The coordinates for the subproblems are typically chosen in a cyclic order.

   The subproblem for the dual (1) over the coordinate r is given by

$$\alpha_r^{\text{new}} = \arg\max_{\alpha_r \in \mathbb{R}} \quad \Psi\left(\alpha_1^{\text{old}}, \ldots, \alpha_{r-1}^{\text{old}}, \alpha_r, \alpha_{r+1}^{\text{old}} \ldots, \alpha_n^{\text{old}}\right)$$
$$\text{subject to: } 0 \leq \alpha_r \leq \frac{C}{n}$$

   where all the variables except for α_r are fixed at previous values. Note that this is a quadratic problem in one variable α_r with an interval constraint: $\alpha_r \in [0, \frac{C}{n}]$.

   1. (**3 Points**) First derive the unconstrained solution $\bar{\alpha}_r$ of the above subproblem (i.e., ignoring the interval constraint). Next show that the solution in the presence of constraints is simply the projection of $\bar{\alpha}_r$ onto the interval $[0, \frac{C}{n}]$. That is

$$\alpha_r^{\text{new}} = \max\left\{0, \min\left\{\bar{\alpha}_r, \frac{C}{n}\right\}\right\}$$

   2. (**1 Point**) Show that the KKT conditions given in the lecture can be rewritten as the following:

$$\forall i = 1, \ldots, n :$$
$$\alpha_i = 0 \Rightarrow y_i \langle w, x_i \rangle \geq 1, \quad 0 < \alpha_i < \frac{C}{n} \Rightarrow y_i \langle w, x_i \rangle = 1, \quad \alpha_i = \frac{C}{n} \Rightarrow y_i \langle w, x_i \rangle \leq 1$$

c. (**2 Points**) Complete the given Matlab function `CoordinateDescentSVM.m` which takes as arguments the training data `Xtrain` $\in \mathbb{R}^{n \times d}$, the class labels `ytrain` $\in \{-1, +1\}^n$, the error parameter C as well as the test data `Xtest`, `ytest` and returns the dual solution α, the primal solution w and training and test errors `TrainErrs`, `TestErrs` computed at each step of the coordinate descent method. Note that test data is only used to compute the test errors at intermediate steps! You have to fill-in the following blocks in the code labelled as `Fill-in`:

1. Compute the unconstrained solution $\bar{\alpha}_r$ and the constrained solution α_r^{new} of the sub-problems
2. Compute the dual objective, training and test errors in each iteration from the current iterate $\alpha^k \in \mathbb{R}^n$
3. Implement stopping criteria: Check if the KKT conditions given above are satisfied by the current iterate α^k upto the given tolerance `EPS`.
4. Compute the primal solution w from the dual solution α

d. (**2 Points**) Train the SVM classifiers for different choices of the error parameter $C = \{10, 100, 200, 500\}$ on the given USPS digit dataset `DIGITS01`. This is a dataset of handwritten digits containing the digits 0 and 1. Load the file `DIGITS01`. The variables `Xtrain` and `Xtest` contain respectively the training and test digit data (each digit is an image of 16×16, so we have 256 gray values) and the variables `ytrain` and `ytest` contains the class labels (-1 for the digit 0 and 1 for the digit 1) for the training and test examples. Plot the training and test errors that you obtained in each iteration of the coordinate descent method and save them as `USPS01TrainErrs_C.png` and `USPS01TestErrs_C.png` for each choice of C.

Hints:

a. Avoid `for` loops in your implementation. All the computations that you need to fill-in can be expressed as matrix-vector multiplication. The operator `.*` (see `MATLAB` help for `times`) that computes element-by-element multiplication of vectors might be helpful here.

b. Note that the dual variable α and the primal variable w are related by

$$w = \sum_{i=1}^{n} y_i \alpha_i x_i,$$

where x_i is the i^{th} row of X. Recall that the prediction at a point x is given by $\text{sign}(\langle w, x \rangle)$ (since the offset $b = 0$).

Submission:

- Create **one** zip/rar/tar.gz/tgz-file containing the m-file (`CoordinateDescentSVM.m`), your plots as .png files and the matlab data file `Solution.mat` containing `TrainErrs` and the `TestError` and send the file to your tutor. The filename has to follow the following convention:
  `[group:A,B,C]_[matrikel numbers separated by underscore]_ex[nr].[extension]`
  e.g. if you are in group B and your team members have matrikelnumbers 3503239, 3028258 and the current exercise number is 14 then the filename reads: `B_3503239_3028258_ex14.zip`.

Exercise 15 - LDA and VC Dimension

a. (**2 Points**) In statistical learning theory, the **VC dimension** of a classifier is the largest cardinality n of a set of points such that the classifier can achieve any of the 2^n possible binary labellings. Prove that the VC dimension of any linear classifier in d dimensions is $d + 1$.

b. (**2 Points**) Consider a binary classification problem in $\mathbb{R}^{50}$ where each class has a normal distribution with zero mean and unit covariance. Suppose you train an LDA classifier on a training sample containing 25 points from each class. Visualize the two-dimensional embedding obtained by LDA. What would be the training error of the LDA classifier? Would the test error also be as small as the training error?

Hints: In part a), it is actually easier to show that there exists $x_i \in \mathbb{R}^d$, $i = 1, \ldots, d+1$ such that for any possible label vector $y \in \{-1, 1\}^{d+1}$ there exist $w \in R^d$ and $b \in R$ such that

$$\langle w, x_i \rangle + b = y_i, \quad i = 1, \ldots, d+1,$$

and then in the next step to argue that there can be no larger set of cardinality $d + 2$.