EC503 Introduction to Learning from Data
Francesco Orabona

Boston University
Spring 2021

Homework 4
Due: Tuesday, March 9, 3:30pm

Important: Submit the code on Blackboard and anything else on Gradescope.

Support Vector Machines

Problem 4.1 In this problem, you will have to implement the soft SVM classification problem in the primal and dual form using the Matlab Quadratic Programming (QP) solver ('quadprog').

QPs are optimization problems in which the objective function is a convex quadratic function and the constraints are linear.

(a) The primal SVM formulation is

$$\begin{aligned}\min_{\mathbf{w},b,\xi_i}\ & \lambda\|\mathbf{w}\|^2+\frac{1}{m}\sum_{i=1}^m \xi_i\\ \text{s.t. } & y_i(\langle \mathbf{w},\mathbf{x}_i\rangle+b)\geq 1-\xi_i, \qquad i=1,\dots,m,\\ & \xi_i\geq 0, \qquad i=1,\dots,m.\end{aligned} \tag{1}$$

Cast this problem as a QP. Lookup the `quadprog` function of Matlab, and write down what `H, f, A, b, Aeq, beq, lb, ub` are. *Hints:* you have to stack all the variable of the optimization in a single vector. So, construct a vector $\mathbf{z}=[\mathbf{w},b,\boldsymbol{\xi}_1,\dots\boldsymbol{\xi}_m]\in\mathbb{R}^{d+1+m}$. Now, you have to express the above optimization problem as a quadratic programming as we have seen in class. Consider one term at the time. First, $\lambda\|\mathbf{w}\|^2$ can be written as $\mathbf{z}^\top\begin{bmatrix}\lambda I_d & Z_1\\ Z_2 & Z_3\end{bmatrix}\mathbf{z}$ where I_d is the identity matrix $d\times d$ and $Z_1\in\mathbb{R}^{d\times 1+m}, Z_2\in\mathbb{R}^{1+m\times d}, Z_3\in\mathbb{R}^{m+1\times m+1}$ are all matrices filled with zeros. In the same way, $\frac{1}{m}\sum_{i=1}^m\xi_i=\mathbf{z}^\top[\underbrace{0,\dots,0,0}_{d+1 \text{ terms}},\underbrace{1/m,\dots,1/m}_{m \text{ terms}}]$.

And so on.

(b) Using the above answer, implement a function

```
[w,b] = train_svm_primal(X, y, lambda)
```

that solves the primal problem for SVMs, where $X\in\mathbb{R}^{m\times d}$ is the matrix of m input vectors in d dimension, i.e., each input $\mathbf{x}_i$ is a row of X, and $\boldsymbol{y}$ is a column vector containing the labels associated with the training samples.

(c) The dual of the SVM problem is:

$$\max_{\boldsymbol{\alpha}}\ \sum_{i=1}^m\alpha_i-\frac{1}{4\lambda}\sum_{i,j=1}^m\alpha_i\alpha_j y_i y_j\langle\mathbf{x}_i,\mathbf{x}_j\rangle$$

$$s.t. \sum_{i=1}^{m} \alpha_i y_i = 0, \tag{2}$$

$$0 \le \alpha_i \le \frac{1}{m}.$$

Write the SVM dual objective as a quadratic program. Use again the `quadprog` function of Matlab, and write down what `H, f, A, b, Aeq, beq, lb, ub` are. *Hint:* don't overthink: this one is simpler than the one above because you only have one vector $\boldsymbol{\alpha}$. The only difficulty here is to find a matrix Q such that $\sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle = \boldsymbol{\alpha}^\top Q \boldsymbol{\alpha}$.

(d) Implement a Matlab function that solves the dual problem and, given the dual solution $\boldsymbol{\alpha}$, computes the primal solution $(\mathbf{w}, b)$

```
[w,b] = train_svm_dual(X, y, lambda)
```

To compute b, use the following code inside of `train_svm_dual`

```
idx=find(alpha>0.00001 & alpha<1/m-0.00001);
b=mean(y(idx)-X(idx,:)*w);
```

We will explain its meaning in class.

(e) Verify that the two (primal and dual) Matlab functions produce the same solution on the Adult dataset from assignment 3, using the script `test_svm.m`

Problem 4.2 In this problem, you will use several SVM variants in a real-world classification problem. For solving the various SVM problems you can use (and modify as necessary) the Matlab functions you implemented above, or use Matlab SVM implementations.

Consider a dataset containing information for 452 patients who may, or may not have, irregular heart beat — so called, arrhythmia. For each patient, there are 279 features, including age, height, weight, and features extracted from an ECG. Details about the features included are at `https://archive.ics.uci.edu/ml/datasets/arrhythmia`. I have pre-processed the data and made them available in the file `arrhythmia.mat`. Loading this file Matlab loads a matrix with the features per patient $\mathbf{X} \in \mathbb{R}^{452 \times 279}$ and a vector $\mathbf{y} \in \mathbb{R}^{452}$ with the label for each patient. If $y_i = 0$, then patient i has arrhythmia, whereas $y_i = 1$ indicates no arrhythmia. (Depending on your SVM implementation, you may need to switch the labels to ±1 instead of 0/1.)

(a) The matrix $\mathbf{X}$ has missing data, denoted as NaN. Impute the missing entries, replacing them with the median of the corresponding feature (column in $\mathbf{X}$). Randomly split the data in 80% for training and 20% for testing. Save and submit your training/test completed data.

(b) Train the following SVM classifiers: linear SVM ($K(\mathbf{x}, \mathbf{x}') = \mathbf{x}^\top \mathbf{x}'$), SVM with the Gaussian kernel ($K(\mathbf{x}, \mathbf{x}') = \exp(-\gamma \|\mathbf{x} - \mathbf{x}'\|^2)$), and SVM with a polynomial kernel ($K(\mathbf{x}, \mathbf{x}') = (1 + \mathbf{x}^\top \mathbf{x}')^k$). Do 3-fold cross-validation to select optimal hyperparameters: λ for the trade-off between errors and margin, γ for the Gaussian kernel, and degree for the polynomial kernel k (try 1, 2, and 3). For each classifier (linear, Gaussian, or polynomial), trained using the optimal parameter set, evaluate its performance on the test set and report the test accuracies. Also, submit the Matlab code you used. *Practical hint:* remember to use a logarithmic scale for γ and λ, for example use powers of 2.

Code-submission via Blackboard: Create three dot-m files, `train_svm_primal.m`, `train_svm_dual.m`, `problem_4_2_b.m`. Place them in a **single** directory which should be zipped and uploaded into Blackboard. Your directory must be named as follows: `<yourBUemailID>_hwX` where `X` is the homework number. For example, if your BU email address is `charles500@bu.edu` then for homework number 4 you would submit a single directory named: `charles500_hw4.zip` which contains all the MATLAB code (and only the code). Reach-out to the TAs via Piazza and their office/discussion hours for questions related to coding.