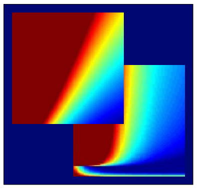
Learning From Data
CaltechX - CS1156x
https://courses.edx.org/courses/CaltechX/CS1156x/Fall2013/info
Fall 2013



Homework # 8

Due Monday, November 25, 2013, at 21:00 GMT

*All questions have multiple-choice answers (***[a]***,* **[b]***,* **[c]***,* **...***). You can collaborate with others, but do not discuss the selected or excluded choices in the answers. You can consult books and notes, but not homework solutions. Your solutions should be based on your own work. Definitions and notation follow the lectures.*

Note about the homework

- The goal of the homework is to facilitate a deeper understanding of the course material. The questions are not designed to be puzzles with catchy answers. They are meant to make you roll your sleeves, face uncertainties, and approach the problem from different angles.

- The problems range from easy to hard, and from theoretical to practical. Some problems require running a full experiment to arrive at the answer.

- The answer may not be obvious or numerically close to one of the choices, but one (and only one) choice will be correct if you follow the exact instructions in each problem. You are encouraged to explore the problem further by experimenting with variations of these instructions, for the learning benefit.

- You are also encouraged to take part in the discussion forum. Please make sure you don't discuss specific answers, or specific excluded answers, before the homework is due.

Primal versus Dual Problem

1. Recall that N is the size of the data set and d is the dimensionality of the input space. The original formulation of the hard-margin SVM problem (minimize $\frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}$ subject to the inequality constraints), without going through the Lagrangian dual problem, is

~~**[a]**~~ a quadratic programming problem with N variables all N α_n ?

~~**[b]**~~ a quadratic programming problem with $N+1$ variables $\|w\| + b$

[c] a quadratic programming problem with d variables

[✓ **[d]**] a quadratic programming problem with $d+1$ variables

[e] not a quadratic programming problem

Notice: The following problems deal with a real-life data set. In addition, the computational packages you use may employ different heuristics and require different tweaks. This is a typical situation that a Machine Learning practitioner faces. There are uncertainties, and the answers may or may not match our expectations. Although this situation is not as 'sanitized' as other homework problems, it is important to go through it as part of the learning experience.

Support Vector Machines With Soft Margins

In this homework set, we are going to experiment with a real-world dataset. Download the processed US Postal Service Zip Code dataset with extracted features of symmetry and intensity for training and testing:

http://www.amlbook.com/data/zip/features.train

http://www.amlbook.com/data/zip/features.test

(the format of each row is: **digit symmetry intensity**). We will train two types of binary classifiers; one-versus-one (one digit is class $+1$ and another digit is class -1, with the rest of the digits disregarded), and one-versus-all (one digit is class $+1$ and the rest of the digits are class -1).

The data set has thousands of points, and some quadratic programming packages cannot handle this size. We recommend that you use the packages in libsvm:

http://www.csie.ntu.edu.tw/~cjlin/libsvm/

Implement SVM with soft margin on the above zip-code data set by solving

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N}\alpha_n\alpha_m y_n y_m K(\mathbf{x}_n,\mathbf{x}_m) - \sum_{n=1}^{N}\alpha_n \\ \text{s.t.} \quad & \sum_{n=1}^{N} y_n\alpha_n = 0 \\ & 0 \le \alpha_n \le C \quad n = 1,\cdots,N \end{aligned}$$

When evaluating E_{in} and E_{out} of the resulting classifier, use binary classification error.

Practical remarks:

(i) For the purpose of this homework, do not scale the data when you use libsvm or other packages, lest you should change the effective kernel and get different results.

(ii) In some packages, you need to specify double precision.

(iii) In 10-fold cross validation, if the data size is not a multiple of 10, the sizes of the 10 subsets may be off by 1 data point.

(iv) Some packages have software parameters whose values affect the outcome. ML practitioners have to deal with this kind of added uncertainty.

Polynomial Kernels

Consider the polynomial kernel $K(\mathbf{x}_n,\mathbf{x}_m) = (1 + \mathbf{x}_n^{\mathrm{T}}\mathbf{x}_m)^Q$, where Q is the degree of the polynomial.

2. With $C = 0.01$ and $Q = 2$, which of the following classifiers has the **highest** E_{in}?

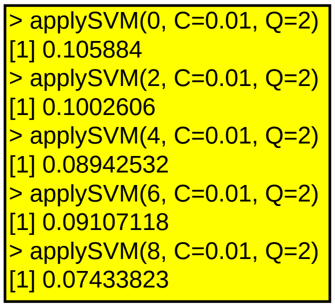
[✓ [a]] 0 versus all

[b] 2 versus all

[c] 4 versus all

[d] 6 versus all

[e] 8 versus all



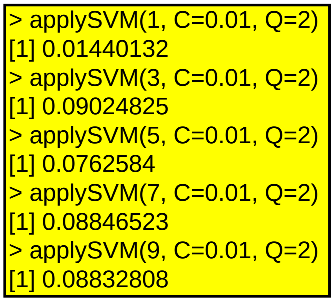
```
> applySVM(0, C=0.01, Q=2)
[1] 0.105884
> applySVM(2, C=0.01, Q=2)
[1] 0.1002606
> applySVM(4, C=0.01, Q=2)
[1] 0.08942532
> applySVM(6, C=0.01, Q=2)
[1] 0.09107118
> applySVM(8, C=0.01, Q=2)
[1] 0.07433823
```

3. With $C = 0.01$ and $Q = 2$, which of the following classifiers has the **lowest** E_{in}?

[✓ [a]] 1 versus all

[b] 3 versus all

[c] 5 versus all



```
> applySVM(1, C=0.01, Q=2)
[1] 0.01440132
> applySVM(3, C=0.01, Q=2)
[1] 0.09024825
> applySVM(5, C=0.01, Q=2)
[1] 0.0762584
> applySVM(7, C=0.01, Q=2)
[1] 0.08846523
> applySVM(9, C=0.01, Q=2)
[1] 0.08832808
```

[d] 7 versus all

[e] 9 versus all

4. Comparing the two classifiers from Problems 2 and 3, which of the following values is the closest to the difference between the number of support vectors of these two classifiers?

[a] 600

[b] 1200

[✓ [c]] 1800

[d] 2400

[e] 3000

1793

```
> m1 <- applySVM(1,5, C=0.001, Q=2)
[1] 0.004484305
> m2 <- applySVM(1,5, C=0.01, Q=2)
[1] 0.004484305
> m3 <- applySVM(1,5, C=0.1, Q=2)
[1] 0.004484305
> m4 <- applySVM(1,5, C=1, Q=2)
[1] 0.003203075
> sapply(list(m1,m2,m3,m4), function(x) length(x$index))
[1] 76 34 24 24
> sapply(list(m1,m2,m3,m4), function(x) mean(test2$digit != predict(x, test2[,2:3])))
[1] 0.01650943 0.01886792 0.01886792 0.01886792
```

5. Consider the 1 versus 5 classifier with $Q = 2$ and $C \in \{0.001, 0.01, 0.1, 1\}$. Which of the following statements is correct?

[a] The number of support vectors goes down when C goes up

[b] The number of support vectors goes up when C goes up

[c] E_{out} goes down when C goes up

[✓ [d]] Maximum C achieves the lowest E_{in}

[e] None of the above

6. In the 1 versus 5 classifier, comparing $Q = 2$ with $Q = 5$, which of the following statements is correct?

[a] When $C = 0.0001$, E_{in} is higher at $Q = 5$

[✓ [b]] When $C = 0.001$, the number of support vectors is lower at $Q = 5$

[c] When $C = 0.01$, E_{in} is higher at $Q = 5$

[d] When $C = 1$, E_{out} is lower at $Q = 5$

[e] None of the above

Cross Validation

In the next two problems, we will experiment with 10-fold cross validation for the polynomial kernel. Because E_{cv} is a random variable that depends on the random partition of the data, we will try 100 runs with different partitions, and base our answer on the number of runs that lead to a particular choice.

7. Consider the 1 versus 5 classifier with $Q = 2$. We use E_{cv} to select $C \in \{0.0001, 0.001, 0.01, 0.1, 1\}$. If there is a tie in E_{cv}, select the smaller C. Within the 100 random runs, which of the following statements is correct?

[a] $C = 0.0001$ is selected most often

[✓ **[b]** $C = 0.001$ is selected most often]

[c] $C = 0.01$ is selected most often

[d] $C = 0.1$ is selected most often

[e] $C = 1$ is selected most often

8. Again, consider the 1 versus 5 classifier with $Q = 2$. For the winning selection in the previous problem, the average value of E_{cv} over the 100 runs is closest to

(a) 0.001

(b) 0.003

[✓ (c) 0.005]

(d) 0.007

(e) 0.009

RBF Kernel

Consider the radial basis function (RBF) kernel $K(\mathbf{x}_n, \mathbf{x}_m) = \exp\left(-||\mathbf{x}_n - \mathbf{x}_m||^2\right)$. Focus on the 1 versus 5 classifier.

9. Which of the following values of C results in the lowest E_{in}?

[a] $C = 0.01$

[b] $C = 1$

[c] $C = 100$

[d] $C = 10^4$

[✓ **[e]** $C = 10^6$]

10. Which of the following values of C results in the lowest E_{out}?

[a] $C = 0.01$

[b] $C = 1$

[✓ **[c]** $C = 100$]

[d] $C = 10^4$

[e] $C = 10^6$