Learning From Data
Caltech - **edX** CS1156x

Fall 2017

Homework # 7

Due Monday, November 13, 2017, at 22:00 GMT/UTC

*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 other people's 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 up your sleeves, face uncertainties, and approach the problem from different angles.

- The problems range from easy to difficult, and from practical to theoretical. 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 instructions precisely in each problem. You are encouraged to explore the problem further by experimenting with variations on 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.

● **Validation**

In the following problems, use the data provided in the files `in.dta` and `out.dta` for Homework # 6. We are going to apply linear regression with a nonlinear transformation for classification (without regularization). The nonlinear transformation is given by ϕ_0 through ϕ_7 which transform (x_1, x_2) into

$$ 1 \quad x_1 \quad x_2 \quad x_1^2 \quad x_2^2 \quad x_1 x_2 \quad |x_1 - x_2| \quad |x_1 + x_2| $$

To illustrate how taking out points for validation affects the performance, we will consider the hypotheses trained on $\mathcal{D}_{\text{train}}$ (without restoring the full $\mathcal{D}$ for training after validation is done).

1. Split `in.dta` into training (first 25 examples) and validation (last 10 examples). Train on the 25 examples only, using the validation set of 10 examples to select between five models that apply linear regression to ϕ_0 through ϕ_k, with $k = 3, 4, 5, 6, 7$. For which model is the classification error on the validation set smallest?

    [a] $k = 3$
    [b] $k = 4$
    [c] $k = 5$
    [d] $k = 6$
    [e] $k = 7$

2. Evaluate the out-of-sample classification error using `out.dta` on the 5 models to see how well the validation set predicted the best of the 5 models. For which model is the out-of-sample classification error smallest?

    [a] $k = 3$
    [b] $k = 4$
    [c] $k = 5$
    [d] $k = 6$
    [e] $k = 7$

3. Reverse the role of training and validation sets; now training with the last 10 examples and validating with the first 25 examples. For which model is the classification error on the validation set smallest?

    [a] $k = 3$
    [b] $k = 4$

[c] $k = 5$

[d] $k = 6$

[e] $k = 7$

4. Once again, evaluate the out-of-sample classification error using `out.dta` on the 5 models to see how well the validation set predicted the best of the 5 models. For which model is the out-of-sample classification error smallest?

[a] $k = 3$

[b] $k = 4$

[c] $k = 5$

[d] $k = 6$

[e] $k = 7$

5. What values are closest in Euclidean distance to the out-of-sample classification error obtained for the model chosen in Problems 1 and 3, respectively?

[a] 0.0, 0.1

[b] 0.1, 0.2

[c] 0.1, 0.3

[d] 0.2, 0.2

[e] 0.2, 0.3

● **Validation Bias**

6. Let e_1 and e_2 be independent random variables, distributed uniformly over the interval $[0, 1]$. Let $e = \min(e_1, e_2)$. The expected values of e_1, e_2, e are closest to

[a] 0.5, 0.5, 0

[b] 0.5, 0.5, 0.1

[c] 0.5, 0.5, 0.25

[d] 0.5, 0.5, 0.4

[e] 0.5, 0.5, 0.5

● **Cross Validation**

7. You are given the data points (x, y): $(-1, 0), (\rho, 1), (1, 0)$, $\rho \geq 0$, and a choice between two models: constant $\{ h_0(x) = b \}$ and linear $\{ h_1(x) = ax + b \}$. For which value of ρ would the two models be tied using leave-one-out cross-validation with the squared error measure?

[a] $\sqrt{\sqrt{3}+4}$

[b] $\sqrt{\sqrt{3}-1}$

[c] $\sqrt{9+4\sqrt{6}}$

[d] $\sqrt{9-\sqrt{6}}$

[e] None of the above

● PLA vs. SVM

Notice: Quadratic Programming packages sometimes need tweaking and have numerical issues, and this is characteristic of packages you will use in practical ML situations. Your understanding of support vectors will help you get to the correct answers.

In the following problems, we compare PLA to SVM with hard margin[1] on linearly separable data sets. For each run, you will create your own target function f and data set $\mathcal{D}$. Take $d = 2$ and choose a random line in the plane as your target function f (do this by taking two random, uniformly distributed points on $[-1, 1] \times [-1, 1]$ and taking the line passing through them), where one side of the line maps to $+1$ and the other maps to -1. Choose the inputs $\mathbf{x}_n$ of the data set as random points in $\mathcal{X} = [-1, 1] \times [-1, 1]$, and evaluate the target function on each $\mathbf{x}_n$ to get the corresponding output y_n. If all data points are on one side of the line, discard the run and start a new run.

Start PLA with the all-zero vector and pick the misclassified point for each PLA iteration at random. Run PLA to find the final hypothesis g_{PLA} and measure the disagreement between f and g_{PLA} as $\mathbb{P}[f(\mathbf{x}) \neq g_{\text{PLA}}(\mathbf{x})]$ (you can either calculate this exactly, or approximate it by generating a sufficiently large, separate set of points to evaluate it). Now, run SVM on the same data to find the final hypothesis g_{SVM} by solving

$$\min_{\mathbf{w},b} \quad \frac{1}{2}\mathbf{w}^{\text{T}}\mathbf{w}$$
$$\text{s.t.} \quad y_n\left(\mathbf{w}^{\text{T}}\mathbf{x}_n + b\right) \geq 1$$

using quadratic programming on the primal[2] or the dual problem, or using an SVM package. Measure the disagreement between f and g_{SVM} as $\mathbb{P}[f(\mathbf{x}) \neq g_{\text{SVM}}(\mathbf{x})]$, and count the number of support vectors you get in each run.

8. For $N = 10$, repeat the above experiment for 1000 runs. How often is g_{SVM} better than g_{PLA} in approximating f? The percentage of time is closest to:

---

[1]For hard margin in SVM packages, set $C \to \infty$.

[2]Primal problem is the original formulation in slide 11 of Lecture 14.

[a] 20%

[b] 40%

[c] 60%

[d] 80%

[e] 100%

9. For $N = 100$, repeat the above experiment for 1000 runs. How often is g_{SVM} better than g_{PLA} in approximating f? The percentage of time is closest to:

[a] 5%

[b] 25%

[c] 45%

[d] 65%

[e] 85%

10. For the case $N = 100$, which of the following is the closest to the average number of support vectors of g_{SVM} (averaged over the 1000 runs)?

[a] 2

[b] 3

[c] 5

[d] 10

[e] 20