Homework #2

RELEASE DATE: 04/18/2017

DUE DATE: 05/09/2017, BEFORE 14:00

QUESTIONS ABOUT HOMEWORK MATERIALS ARE WELCOMED ON THE FACEBOOK FORUM.

Unless granted by the instructor in advance, you must turn in a printed/written copy of your solutions (without the source code) for all problems.

For problems marked with (*), please follow the guidelines on the course website and upload your source code to designated places. You are encouraged to (but not required to) include a README to help the TAs check your source code. Any programming language/platform is allowed.

Any form of cheating, lying, or plagiarism will not be tolerated. Students can get zero scores and/or fail the class and/or be kicked out of school and/or receive other punishments for those kinds of misconducts.

Discussions on course materials and homework solutions are encouraged. But you should write the final solutions alone and understand them fully. Books, notes, and Internet resources can be consulted, but not copied from.

Since everyone needs to write the final solutions alone, there is absolutely no need to lend your homework solutions and/or source codes to your classmates at any time. In order to maximize the level of fairness in this class, lending and borrowing homework solutions are both regarded as dishonest behaviors and will be punished according to the honesty policy.

You should write your solutions in English or Chinese with the common math notations introduced in class or in the problems. We do not accept solutions written in any other languages.

This homework set comes with 160 points and 40 bonus points. In general, every homework set would come with a full credit of 160 points, with some possible bonus points.

Descent Methods for Probabilistic SVM

Recall that the probabilistic SVM is based on solving the following optimization problem:

$$\min_{A,B} \quad F(A,B) = \frac{1}{N}\sum_{n=1}^{N} \ln\left(1+\exp\left(-y_n\left(A\cdot\left(\mathbf{w}_{\text{SVM}}^T\boldsymbol{\phi}(\mathbf{x}_n)+b_{\text{SVM}}\right)+B\right)\right)\right).$$

1. When using the gradient descent for minimizing $F(A,B)$, we need to compute the gradient first. Let $z_n = \mathbf{w}_{\text{SVM}}^T\boldsymbol{\phi}(\mathbf{x}_n)+b_{\text{SVM}}$, and $p_n = \theta(-y_n(Az_n+B))$, where $\theta(s) = \frac{\exp(s)}{1+\exp(s)}$ is the usual logistic function. What is the gradient $\nabla F(A,B)$ in terms of only y_n, p_n, z_n and N? Prove your answer.

2. When using the Newton method for minimizing $F(A,B)$ (see Homework 3 of Machine Learning Foundations), we need to compute $-(H(F))^{-1}\nabla F$ in each iteration, where $H(F)$ is the Hessian matrix of F at (A,B). Following the notations of Question 1, what is $H(F)$ in terms of only y_n, p_n, z_n and N? Prove your answer.

Kernel Ridge Regression

3. Assume that all $\mathbf{x}_n$ are different. When using the Gaussian kernel with $\gamma \to \infty$, what does the kernel matrix K used in kernel ridge regression look like? What is the optimal $\boldsymbol{\beta}$? Prove your answer.

4. When using the Gaussian kernel with $\gamma \to 0$, what does the kernel matrix K used in kernel ridge regression look like? What is the optimal $\boldsymbol{\beta}$? Prove your answer.

Support Vector Regression

The usual support vector regression model solves the following optimization problem.

$$(P_1) \min_{b,\mathbf{w},\boldsymbol{\xi}^\vee,\boldsymbol{\xi}^\wedge} \quad \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{n=1}^{N}(\xi_n^\vee + \xi_n^\wedge)$$
$$\text{s.t.} \quad -\epsilon - \xi_n^\vee \le y_n - \mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_n) - b \le \epsilon + \xi_n^\wedge$$
$$\xi_n^\vee \ge 0, \xi_n^\wedge \ge 0.$$

Usual support vector regression penalizes the violations $\xi_n^\vee$ and $\xi_n^\wedge$ linearly. Another popular formulation, called ℓ_2 loss support vector regression in (P_2), penalizes the violations quadratically, just like the ℓ_2 loss SVM introduced in Homework 1 of Machine Learning Techniques.

$$(P_2) \min_{b,\mathbf{w},\boldsymbol{\xi}^\vee,\boldsymbol{\xi}^\wedge} \quad \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{n=1}^{N}\left((\xi_n^\vee)^2 + (\xi_n^\wedge)^2\right)$$
$$\text{s.t.} \quad -\epsilon - \xi_n^\vee \le y_n - \mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_n) - b \le \epsilon + \xi_n^\wedge.$$

5. Write down an equivalent 'unconstrained' form of (P_2) that is similar to page 10 of the "Support Vector Regression" lecture and prove the equivalence.

6. By a slight modification of the representer theorem presented in the class, the optimal $\mathbf{w}_*$ for (P_2) must satisfy $\mathbf{w}_* = \sum_{n=1}^{N}\beta_n\mathbf{z}_n$. We can substitute the form of the optimal $\mathbf{w}_*$ into the answer in Question 4 to derive an optimization problem that contains $\boldsymbol{\beta}$ (and b) only, which would look like

$$\min_{b,\boldsymbol{\beta}} F(b,\boldsymbol{\beta}) = \frac{1}{2}\sum_{m=1}^{N}\sum_{n=1}^{N}\beta_n\beta_m K(\mathbf{x}_n,\mathbf{x}_m) + \text{ something},$$

where $K(\mathbf{x}_n,\mathbf{x}_m) = (\boldsymbol{\phi}(\mathbf{x}_n))^T(\boldsymbol{\phi}(\mathbf{x}_m))$ is the kernel function. One thing that you should see is that $F(b,\boldsymbol{\beta})$ is differentiable to β_n (and b) and hence you can use gradient descent to solve for the optimal $\boldsymbol{\beta}$. For any $\boldsymbol{\beta}$, let $s_n = \sum_{m=1}^{N}\beta_m K(\mathbf{x}_n,\mathbf{x}_m) + b$. What is $\frac{\partial F(b,\boldsymbol{\beta})}{\partial \beta_m}$? Prove your answer.

Blending

7. Consider the case where the target function $f : [0,1] \to \mathbb{R}$ is given by $f(x) = x^2$ and the input probability distribution is uniform on $[0,1]$. Assume that the training set has only two examples generated independently from the input probability distribution and noiselessly by f, and the learning model is usual linear regression that minimizes the mean squared error within all hypotheses of the form $h(x) = w_1x + w_0$. What is $\bar{g}(x)$, the expected value of the hypothesis, that the learning algorithm produces (see Page 10 of Lecture 207)? Prove your answer.

Test Set Linear Regression.

The root-mean-square-error (RMSE) of a hypothesis h on a test set $\{(\tilde{\mathbf{x}}_n,\tilde{y}_n)\}_{n=1}^{\tilde{N}}$ ($\tilde{\mathbf{x}} \in R^d$, $\tilde{y} \in R$) is defined as

$$\text{RMSE}(h) = \sqrt{\frac{1}{\tilde{N}}\sum_{n=1}^{\tilde{N}}(\tilde{y}_n - h(\tilde{\mathbf{x}}_n))^2}$$

In the next questions, please consider a case of knowing all the $\tilde{\mathbf{x}}_n$, none of the $\tilde{y}_n$, but allowed to query RMSE(h) for T (different) h.

8. How many queries is needed for constructing some hypothesis g with RMSE(g) = 0? In other words, what is the minimum number of queries needed for "cheating" from the RMSEs to obtain $g(\tilde{\mathbf{x}}_n) = \tilde{y}_n$ for every $n = 1, 2, \cdots, \tilde{N}$? Please illustrate your answer.

9. The algorithm above is slow (and thus "impractical") if $\tilde{N}$ is too large. Let us start designing a smarter way of "cheating." For any given hypothesis g, let

$$\begin{aligned} \mathbf{g} &= (g(\tilde{\mathbf{x}}_1), g(\tilde{\mathbf{x}}_2), \cdots, g(\tilde{\mathbf{x}}_{\tilde{N}})) \\ \tilde{\mathbf{y}} &= (\tilde{y}_1, \tilde{y}_2, \cdots, \tilde{y}_{\tilde{N}}). \end{aligned}$$

Note that you can compute $\mathbf{g}$ but you do not know $\tilde{\mathbf{y}}$. What is the minimum number of queries needed for computing $\mathbf{g}^T\tilde{\mathbf{y}}$? Please illustrate your answer.

10. For any given set of hypotheses $\{g_1, g_2, \cdots, g_K\}$, use the result in the previous question to design an algorithm to solve

$$\min_{\alpha_1,\alpha_2,\cdots,\alpha_K} \text{RMSE}\left(\sum_{k=1}^{K} \alpha_k g_k\right),$$

and obtain the optimal weights $\alpha_1, \cdots, \alpha_K$. What is the minimum number of queries needed? Please illustrate your answer.

Experiment with Kernel Ridge Regression. Write a program to implement the kernel ridge regression algorithm from Lecture 206, and use it for classification (i.e. implement LSSVM). Consider the following data set

`hw2_lssvm_all.dat`

Use the first 400 examples for training and the remaining for testing. Calculate E_{in} and E_{out} with the 0/1 error. Consider the Gaussian-RBF kernel $\exp\left(-\gamma\|\mathbf{x}-\mathbf{x}'\|^2\right)$. Try all combinations of parameters $\gamma \in \{32, 2, 0.125\}$ and $\lambda \in \{0.001, 1, 1000\}$.

11. (*) Among all parameter combinations, which combination results in the minimum $E_{\text{in}}(g)$? What is the corresponding $E_{\text{in}}(g)$?

12. (*) Among all parameter combinations, which combination results in the minimum $E_{\text{out}}(g)$? What is the corresponding $E_{\text{out}}(g)$?

.

Experiment with Support Vector Regression.

Write a program to implement the nonlinear SVR from Lecture 205, and use the SVR for classification. Consider the following data set

`hw2_lssvm_all.dat`

Use the first 400 examples for training and the remaining for testing. Calculate E_{in} and E_{out} with the 0/1 error. Consider the Gaussian-RBF kernel $\exp\left(-\gamma\|\mathbf{x}-\mathbf{x}'\|^2\right)$. With a fixed $\epsilon = 0.5$, try all combinations of parameters $\gamma \in \{32, 2, 0.125\}$ and $C \in \{0.001, 1, 1000\}$ (The original problem uses λ instead of C. It is okay if you have taken $C = N/\lambda$ to solve the problem, but please mark so carefully to facilitate the TAs in grading.) . (*Note: For this problem, you CAN use any package you want. A recommended choice is LIBSVM developed by Prof. Chih-Jen Lin in our department*)

13. (*) Among all parameter combinations, which combination results in the minimum $E_{\text{in}}(g)$? What is the corresponding $E_{\text{in}}(g)$?

14. (*) Among all parameter combinations, which combination results in the minimum $E_{\text{out}}(g)$? What is the corresponding $E_{\text{out}}(g)$?

.

Experiment with Bagging Ridge Regression.

Write a program to implement bagging on top of linear LSSVM (i.e. kernel ridge regression for classification). You can reuse the code in the previous problem. Again consider the following data set

`hw2_lssvm_all.dat`

Use the first 400 examples for training and the remaining for testing. Calculate E_{in} and E_{out} with the 0/1 error. Note that each linear LSSVM should take the sign operation before uniform aggregation (with voting). Consider $\lambda \in \{0.01, 0.1, 1, 10, 100\}$. Take at least 200 iterations for bagging. It is suggested to add $x_0 = 1$ to your data, but not a must. You can just illustrate what you have done clearly.

15. (*) Among all parameter combinations, which combination results in the minimum $E_{\text{in}}(g)$? What is the corresponding $E_{\text{in}}(g)$?

16. (*) Among all parameter combinations, which combination results in the minimum $E_{\text{out}}(g)$? What is the corresponding $E_{\text{out}}(g)$?

Bonus: Linear Blending with SVM Solver

Consider blending T hypothesis g_1, g_2, …, g_T linearly with coefficients α_1, α_2, …, α_T using the hinge error function. That is, we want to solve

$$\min_{\alpha_t \geq 0} \frac{1}{N} \sum_{n=1}^{N} \max\left(1 - y_n \sum_{t=1}^{T} \alpha_t g_t(\mathbf{x}_n), 0\right)$$

For your information, one of the solvers in the LIBLINEAR package from Prof. Chih-Jen Lin's group solves the following SVM problem, which is slightly different from the problem introduced in class. In particular, there is no variable b in the formulation.

$$\begin{aligned} \min_{\mathbf{w}} \quad & \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{n=1}^{N} \xi_n \\ \text{subject to} \quad & y_n(\mathbf{w}^T\mathbf{x}_n) \geq 1 - \xi_n \text{ and } \xi_n \geq 0 \text{ for } n = 1, 2, \ldots N. \end{aligned}$$

17. (Bonus, 20 points) Describe a procedure to solve the linear blending problem above **without the** $\alpha_t \geq 0$ **constraints** using only the LIBLINEAR solver above.

18. (Bonus, 20 points) The constraints $\alpha_t \geq 0$ denote the non-negative vote of each g_t. If you are so confident of your g_t such that you think every hypothesis should deserve at least one vote, you can make the constraints even more strict with $\alpha_t \geq 1$. Describe a procedure to solve the linear blending problem above **with the** $\alpha_t \geq 1$ **constraints** using only the LIBLINEAR solver above. (*Hint: Can we use* $\tilde{N}$ *instead of* N *examples?*)

Note: In the two questions above, you can reasonably consider setting the parameter C properly to approximate the problem that you want to solve.