Homework #4

TA in charge: Chao-Kai Chiang

RELEASE DATE: 10/25/2010

DUE DATE: 11/08/2010, 4:00 pm IN CLASS

TA SESSION: 11/04/2010, 6:00 pm IN R110

Unless granted by the instructor in advance, you must turn in a hard 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.

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 with the common math notations introduced in class or in the problems. We do not accept solutions written in any other languages.

4.1 More on Growth Function and VC Dimension

(1) (5%) Let $\mathcal{H} = \{h_1, h_2, \ldots, h_M\}$ with some finite M. Prove that $d_{\mathsf{VC}}(\mathcal{H}) \leq \log_2 M$.

(2) (5%) For hypothesis sets $\mathcal{H}_1$, $\mathcal{H}_2$, $\cdots$, $\mathcal{H}_K$ with finite VC-dimensions $d_{\mathsf{VC}}(\mathcal{H}_k)$, derive and prove the tightest lower bound that you can get on $d_{\mathsf{VC}}\left(\cap_{k=1}^{K}\mathcal{H}_k\right)$.

(3) (5%) For hypothesis sets $\mathcal{H}_1$, $\mathcal{H}_2$, $\cdots$, $\mathcal{H}_K$ with finite VC-dimensions $d_{\mathsf{VC}}(\mathcal{H}_k)$, derive and prove the tightest upper bound that you can get on $d_{\mathsf{VC}}\left(\cap_{k=1}^{K}\mathcal{H}_k\right)$.

(4) (5%) For hypothesis sets $\mathcal{H}_1$, $\mathcal{H}_2$, $\cdots$, $\mathcal{H}_K$ with finite VC-dimensions $d_{\mathsf{VC}}(\mathcal{H}_k)$, derive and prove the tightest lower bound that you can get on $d_{\mathsf{VC}}\left(\cup_{k=1}^{K}\mathcal{H}_k\right)$.

(5) (5%) For hypothesis sets $\mathcal{H}_1$, $\mathcal{H}_2$, $\cdots$, $\mathcal{H}_K$ with finite VC-dimensions $d_{\mathsf{VC}}(\mathcal{H}_k)$, derive and prove the tightest upper bound that you can get on $d_{\mathsf{VC}}\left(\cup_{k=1}^{K}\mathcal{H}_k\right)$.

4.2 The Hat of Linear Regression

(1) (3%) Do Exercise 3.3-1 of LFD.

(2) (3%) Do Exercise 3.3-2 of LFD.

(3) (3%) Do Exercise 3.3-3 of LFD.

(4) (3%) Do Exercise 3.3-4 of LFD.

(5) (3%) Do Exercise 3.3-5 of LFD.

4.3 The Feature Transforms

(1) (4%) Do Exercise 3.6 of LFD.

(2) (6%) Do Exercise 3.7 of LFD.

(3) (5%) Do Exercise 3.11 of LFD.

4.4 Gradient and Newton Directions

Consider a function

$$E(u, v) = e^u + e^{2v} + e^{uv} + u^2 - 3uv + 4v^2 - 3u - 5v,$$

(1) (3%) Approximate $E(u + \Delta u, v + \Delta v)$ by $\hat{E}_1(\Delta u, \Delta v)$, where $\hat{E}_1$ is the first-order Taylor's expansion of E around $(u, v) = (0, 0)$. Suppose $\hat{E}_1(\Delta u, \Delta v) = a_u \Delta u + a_v \Delta v + a$. What are the values of a_u, a_v, and a?

(2) (3%) Minimize $\hat{E}_1$ over all possible $(\Delta u, \Delta v)$ such that $\|(\Delta u, \Delta v)\| = 0.5$. In class, we proved that the optimal column vector $\begin{bmatrix}\Delta u \\ \Delta v\end{bmatrix}$ is parallel to the column vector $-\nabla E(u, v)$, which is called the *negative gradient direction*. Compute the optimal $(\Delta u, \Delta v)$ and the resulting $E(u+\Delta u, v+\Delta v)$.

(3) (3%) Approximate $E(u + \Delta u, v + \Delta v)$ by $\hat{E}_2(\Delta u, \Delta v)$, where $\hat{E}_2$ is the second-order Taylor's expansion of E around $(u, v) = (0, 0)$. Suppose

$$\hat{E}_2(\Delta u, \Delta v) = b_{uu}(\Delta u)^2 + b_{vv}(\Delta v)^2 + b_{uv}(\Delta u)(\Delta v) + b_u \Delta u + b_v \Delta v + b.$$

What are the values of b_{uu}, b_{vv}, b_{uv}, b_u, b_v, and b?

(4) (3%) Minimize $\hat{E}_2$ over all possible $(\Delta u, \Delta v)$ (regardless of length). Use the fact that $\nabla^2 E(u, v)$ (the Hessian matrix) is positive definite to prove that the optimal column vector

$$\begin{bmatrix}\Delta u \\ \Delta v\end{bmatrix} = -\left(\nabla^2 E(u, v)\right)^{-1} \nabla E(u, v),$$

which is called the *Newton direction*.

(5) (3%) Numerically compute the following values:

(a) the vector $(\Delta u, \Delta v)$ of length 0.5 along the Newton direction, and the resulting $E(u + \Delta u, v + \Delta v)$.

(b) the vector $(\Delta u, \Delta v)$ of length 0.5 that minimizes $E(u + \Delta u, v + \Delta v)$, and the resulting $E(u + \Delta u, v + \Delta v)$. (*Hint: let* $\Delta u = 0.5 \sin\theta$.)

Compare the values of $E(u + \Delta u, v + \Delta v)$ in (2), (5a), and (5b). Briefly state your findings.

The negative gradient direction and the Newton direction are quite fundamental for designing optimization algorithms. It is important to understand these directions and put them in your toolbox for designing ML algorithms.

4.5 Least-squares Linear Regression (*)

(1) (8%) Implement the least-squares linear regression algorithm taught in class to compute the optimal $(d + 1)$-dimensional $\mathbf{w}$ that solves

$$\min_{\mathbf{w}} \sum_{n=1}^{N} \Big(y_n - \big(\mathbf{w} \bullet \mathbf{x}_n\big)\Big)^2.$$

Run the algorithm on the following set for training (each row represents a pair of $(\mathbf{x}_n, y_n)$, where $\mathbf{x}_n$ is the "thin" version. The first column is $\mathbf{x}_n[1]$, the second one is $\mathbf{x}_n[2]$, and the third one is y_n):

`http://www.csie.ntu.edu.tw/~htlin/course/ml10fall/data/hw4_train.dat`

and the following set for testing:

`http://www.csie.ntu.edu.tw/~htlin/course/ml10fall/data/hw4_test.dat`

Report the $\mathbf{w}$ you find. Let $g(x) = \text{sign}\big(\mathbf{w} \bullet \mathbf{x}\big)$. What is $E_{\text{in}}(g)$ in terms of the 0/1 loss (classification)? How about $E_{\text{out}}(g)$?

Please check the course policy carefully and do not use sophisticated packages in your solution. You ***can*** *use standard matrix multiplication and inversion routines.*

4.6 Gradient Descent for Logistic Regression (*)

Consider the formulation (so-called *logistic regression*)

$$\min_{\mathbf{w}} \quad E(\mathbf{w}), \tag{A1}$$

$$\text{where} \quad E(\mathbf{w}) = \frac{1}{N}\sum_{n=1}^{N} E^{(n)}(\mathbf{w}), \text{ and } E^{(n)}(\mathbf{w}) = \ln\left(1 + \exp\Big(-y_n\big(\mathbf{w} \bullet \mathbf{x}_n\big)\Big)\right).$$

(1) (3%) Prove that $\frac{1}{\ln 2}E^{(n)}(\mathbf{w})$ is an upper bound of $[\![\text{sign}(\mathbf{w} \bullet \mathbf{x}_n) \neq y_n]\!]$ for any $\mathbf{w}$.

(2) (3%) For a given $(\mathbf{x}_n, y_n)$, derive its gradient $\nabla E^{(n)}(\mathbf{w})$.

(3) (8%) Implement the (fixed-step) stochastic gradient descent algorithm below for (A1).

(a) initialize a $(d+1)$-dimensional vector $\mathbf{w}^{(0)}$, say, $\mathbf{w}^{(0)} \longleftarrow (0, 0, \ldots, 0)$.

(b) for $t = 1, 2, \ldots, T$

- randomly pick one n from $\{1, 2, \ldots, N\}$.
- update

$$\mathbf{w}^{(t)} \longleftarrow \mathbf{w}^{(t-1)} - \eta \cdot \nabla E^{(n)}(\mathbf{w}^{(t-1)}).$$

Assume that

$$g_1^{(t)}(\mathbf{x}) = \text{sign}\left(\mathbf{w}^{(t)} \bullet \mathbf{x}\right),$$

where $\mathbf{w}^{(t)}$ are generated from stochastic gradient descent algorithm above. Run the algorithm with $\eta = 0.001$ and $T = 2000$ on the following set for training:

`http://www.csie.ntu.edu.tw/~htlin/course/ml10fall/data/hw4_train.dat`

and the following set for testing:

`http://www.csie.ntu.edu.tw/~htlin/course/ml10fall/data/hw4_test.dat`

Plot $E_{\text{in}}\left(g_1^{(t)}\right)$ and $E_{\text{out}}\left(g_1^{(t)}\right)$ as a function of t and briefly state your findings.

(4) (8%) Implement the (fixed-step) gradient descent algorithm below for (A1).

(a) initialize a $(d+1)$-dimensional vector $\mathbf{w}^{(0)}$, say, $\mathbf{w}^{(0)} \longleftarrow (0, 0, \ldots, 0)$.

(b) for $t = 1, 2, \ldots, T$

- update

$$\mathbf{w}^{(t)} \longleftarrow \mathbf{w}^{(t-1)} - \eta \cdot \nabla E(\mathbf{w}^{(t-1)}) .$$

Assume that

$$g_2^{(t)}(\mathbf{x}) = \text{sign}\left(\mathbf{w}^{(t)} \bullet \mathbf{x}\right),$$

where $\mathbf{w}^{(t)}$ are generated from gradient descent algorithm above. Run the algorithm with $\eta = 0.001$ and $T = 2000$ on the following set for training:

`http://www.csie.ntu.edu.tw/~htlin/course/ml10fall/data/hw4_train.dat`

and the following set for testing:

`http://www.csie.ntu.edu.tw/~htlin/course/ml10fall/data/hw4_test.dat`

Plot $E_{\text{in}}\left(g_2^{(t)}\right)$ and $E_{\text{out}}\left(g_2^{(t)}\right)$ as a function of t, compare it to your plot for $g_1^{(t)}$, and briefly state your findings.