

Homework 4

10-702/36-702 Statistical Machine Learning

Due: Friday Mar 18 3:00

Hand in to: Michelle Martin GHC 8001.

1 Minimax Theory

1. Recall that $\text{TV}(P, Q) = \sup_A |P(A) - Q(A)|$. Prove that

$$\text{TV}(P, Q) = \frac{1}{2} \int |p - q|.$$

2. Recall that $N(\epsilon)$ denotes the covering number and $P(\epsilon)$ denotes the packing number. Show that

$$P(2\epsilon) \leq N(\epsilon) \leq P(\epsilon).$$

3. Let \mathcal{F} denote all functions from $[0, 1]$ to $[0, 1]$ that are nondecreasing. Let P be a probability measure on $[0, 1]$. Recall that $N_{[]}(\epsilon, \mathcal{F}, L_2(P))$ is the smallest number of pairs $(\ell_1, u_1), \dots, (\ell_N, u_N)$ such that: (i) $\sqrt{\int (u_j - \ell_j)^2 dP} \leq \epsilon$ and (ii) for each f in \mathcal{F} there is a pair (ℓ_j, u_j) such that $\ell_j(x) \leq f(x) \leq u_j(x)$ for all x . Show that

$$N_{[]}(\epsilon, \mathcal{F}, L_2(P)) \leq \exp\left(\frac{C}{\epsilon}\right)$$

for some $C > 0$.

4. Let $X \in [0, 1]$ and $Y|X = x \sim \text{Bernoulli}(\eta(x))$ where $\eta(x) = P(Y = 1|X = x)$. Suppose the true distributions $P_{XY} \in \mathcal{P}$ is such that the Bayes optimal classifiers are indicators of the form $\mathbf{1}([t, 1])$. Show that

$$\inf_{\hat{f}_n} \sup_{P_{XY} \in \mathcal{P}} \mathbb{E}[R(\hat{f}_n) - R^*] \geq C \sqrt{\frac{1}{n}}$$

where the inf is over all possible classifiers based on n training examples, and $C > 0$ is a constant. This is the minimax rate of convergence for any parametric classification problem.

Hint 1: You only need to construct a set of two hypothesis $P^{(0)}$ and $P^{(1)}$ that belong to the class \mathcal{P} . A convenient choice is to consider two hypothesis that have the same marginal distribution $P_X^{(0)} = P_X^{(1)} \sim \text{uniform}([0, 1])$ and only differ in $P_{Y|X}^{(0)}, P_{Y|X}^{(1)}$.

2 Graphical Models

1. Consider a p -dimensional Gaussian graphical model $P_X \sim \mathcal{N}(0, \Sigma)$ defined on $X = (X_1, \dots, X_p)$. Let $\Omega = \Sigma^{-1}$ denote the precision matrix. In this problem, you will show that $\Omega_{ij} = 0$ iff X_i is conditionally independent of X_j given the remaining variables.
 - (a) Partition $X = (Y, Z)$ where Y is a subset of the p variables and Z denotes the remaining variables. What is the conditional distribution $P(Y|Z)$?
 - (b) Denoting the precision matrix in block form $\Omega = [\Omega_{YY} \ \Omega_{YZ}; \Omega_{ZY} \ \Omega_{ZZ}]$. Show that $\Omega_{YY} = \text{var}(Y|Z)$. (Hint: Use the form of inverse of a block matrix in terms of Schur complement)
 - (c) Using the above two results, argue that $\Omega_{ij} = 0$ iff X_i is conditionally independent of X_j given the remaining variables.

The above results motivate the Graphical Lasso (Glasso) algorithm. Suppose we have data (X^1, \dots, X^n) where each X^i is a 0-mean p -dimensional multivariate Gaussian, then we perform optimization:

$$\arg \min_{\Omega \in \mathcal{S}_p^+} \underbrace{-\log |\Omega| + \text{tr}(\Omega S_n)}_{\text{negative log-likelihood}} + \underbrace{\lambda \|\Omega\|_1}_{\text{regularization}}$$

where \mathcal{S}_p^+ is set of all $p \times p$ positive definite matrix, $S_n = \frac{1}{n} \sum_{i=1}^n X^i X^{iT}$ is the sample covariance, and $\|\Omega\|_1 = \sum_{i,j} |\Omega_{i,j}|$ is a ℓ_1 penalty on every element of Ω .

This optimization produce an inverse covariance matrix Ω with many zero-entries, which correspond to a sparse graph.

2. Meinshausen and Buhlmann in 2006 derived an alternative method of estimating a sparse Gaussian graphical model. Recall an important theorem from Stat 705:
 - Let X, Y be two random vectors and suppose we want to regress X onto Y . Then the expected-least-square regression function $m(X)$ is $\mathbb{E}[Y|X]$, i.e.:

$$\underset{m \text{ function } X \mapsto Y}{\text{argmin}} \quad \mathbb{E}[\|Y - m(X)\|_2^2] = \mathbb{E}[Y|X]$$

We use the following set-up for the problem: Let $X = (X_1, \dots, X_p)$ be a p -dimensional random Gaussian vector.

- (a) Suppose we regress all X_j for $j \neq i$ onto the variable X_i , prove that the expected-least-square regression function $m(\{X_j\}_{j \neq i})$ is $m(\{X_j\}_{j \neq i}) = \sum_{j \neq i} \beta^i(j) X_j$ where $\beta^i(j) = -\frac{\Omega_{ij}}{\Omega_{ii}}$ (Hint: Use Schur Complement again)

Let X be a $n \times p$ data matrix where n is the number of samples. Let X_i denote the i -th column of X . The Meinshausen-Buhlmann multiple Lasso procedure is the following:

For $i = 1, \dots, p$, solve

$$\arg \min_{\beta^i \in \mathbb{R}^{p-1}} \frac{1}{2n} \|X_i - \sum_{j \neq i} \beta^i(j) X_j\|_2^2 + \lambda \|\beta^i\|_1$$

Put an edge between X_i and X_j if either $\beta^i(j) \neq 0$ OR $\beta^j(i) \neq 0$

3. The file “Xs.txt” contains 400 samples of a 100 dimensional multivariate Gaussian whose true inverse-covariance matrix is specified by “Omega.txt”. For the following exercises, we highly recommend that you use the R programming language since it already has packages that implemented the glasso and lasso algorithms (refer to the `glasso` and `lars` packages)
 - (a) Using either your own implementation or publicly available package, run the Glasso algorithm on the data. Try various values for the λ tuning parameter until you get a graph of about 400 to 600 edges. Plot the precision (percentage of true edges among the total edges recovered) vs. the recall (percentage of edges recovered among the true edges) of glasso that you obtain by varying λ . Remember to standardize your data so that each covariate has variance 1 (so each column should have norm \sqrt{n}).
 - (b) Using the same data, perform the same empirical study with the Meinshausen and Buhlmann multiple Lasso algorithm. Again, standardize your data and report the precision vs. recall by varying λ .
 - (c) Include a printout of your code in your homework submission.

Note: R is a powerful and easy to learn language. There are various tutorials online. For your convenience, you may also refer to “example.txt” for an example of a simple R script. “example.txt” generates a random sparse inverse covariance matrix and samples from the resulting distribution.

If you choose to not use R, you can find the implementation details of glasso in the following paper:

- <http://jmlr.csail.mit.edu/papers/volume9/banerjee08a/banerjee08a.pdf>