

Feature Selection

Lecturer: Changshui Zhang zcs@mail.tsinghua.edu.cn

Student: XXX xxx@mails.tsinghua.edu.cn

Problem 1

Relation between Fisher Criterion and Least Squares

It's interesting to see that under some circumstances, the fisher criterion can be obtained as a special case of least squares. Consider the binary classification problem, let's unify the expression at the very beginning for convenience of the following steps. And you are required to obey the notations given below.

Suppose we have N_1 points of class \mathcal{C}_1 and N_2 of class \mathcal{C}_2 , then the mean vectors of the two classes are given by:

$$\mathbf{m}_1 = \frac{1}{N_1} \sum_{n \in \mathcal{C}_1} \mathbf{x}_n, \quad \mathbf{m}_2 = \frac{1}{N_2} \sum_{n \in \mathcal{C}_2} \mathbf{x}_n \quad (1)$$

In the lecture notes, we have defined *between-class* covariance matrix and *within-class* covariance matrix:

$$S_B = (\mathbf{m}_2 - \mathbf{m}_1)(\mathbf{m}_2 - \mathbf{m}_1)^T, \quad S_W = \sum_{n \in \mathcal{C}_1} (\mathbf{x}_n - \mathbf{m}_1)(\mathbf{x}_n - \mathbf{m}_1)^T + \sum_{n \in \mathcal{C}_2} (\mathbf{x}_n - \mathbf{m}_2)(\mathbf{x}_n - \mathbf{m}_2)^T \quad (2)$$

Now, let's turn to the least square problem. We take the targets for \mathcal{C}_1 to be N/N_1 and \mathcal{C}_2 to be $-N/N_2$ where $N = N_1 + N_2$ (This may be a little confusing, but you will see the reasons of doing so in a short time). Then the sum-of-square error function can be written as:

$$E = \frac{1}{2} \sum_{n=1}^N (\mathbf{w}^T \mathbf{x}_n + w_0 - t_n)^2 \quad (3)$$

Where, (\mathbf{x}_n, t_n) are the points we have. Target t_n equals to N/N_1 or $-N/N_2$ according to its class. Our goal is to estimate \mathbf{w} and w_0 .

1.1 Show that the optimal w_0 is: $w_0 = -\mathbf{w}^T \mathbf{m}$.

1.2 Derive the equation that the optimal \mathbf{w} should obey:

$$(S_W + \frac{N_1 N_2}{N} S_B) \mathbf{w} = N(\mathbf{m}_1 - \mathbf{m}_2) \quad (4)$$

1.3 Show me that \mathbf{w} satisfies: $\mathbf{w} \propto S_W^{-1}(\mathbf{m}_2 - \mathbf{m}_1)$ from equation (4), which means we've got the same form as that of fisher criterion.

Problem 2

Fisher's discriminant for multiple classes

Consider the generalization of the Fisher discriminant to $K > 2$ classes, and assume that the dimensionality of the input space is greater than the number K of classes. Next, we introduce $D' > 1$ linear 'features' $y_k = \mathbf{w}_k^T \mathbf{x}$. The weight vectors $\{\mathbf{w}_k\}'s$ can be considered to be the columns of a matrix \mathbf{W} , so that:

$$\mathbf{y} = \mathbf{W}^T \mathbf{x}$$

Where, $\mathbf{x} \in \mathcal{R}^D$ and $\mathbf{y} \in \mathcal{R}^{D'}$. By this means, we have projected the D -dimensional \mathbf{x} -space onto the D' -dimensional \mathbf{y} -space, in which we can better separate the data.

The generalization of the *within-class* covariance matrix to the case of K classes is:

$$S_W = \sum_{k=1}^K \sum_{n \in \mathcal{C}_k} (\mathbf{x}_n - \mathbf{m}_k)(\mathbf{x}_n - \mathbf{m}_k)^T \quad (5)$$

Where $\mathbf{m}_k = \frac{1}{N_k} \sum_{n \in \mathcal{C}_k} \mathbf{x}_n$.

The total covariance matrix is:

$$S_T = \sum_{n=1}^N (\mathbf{x}_n - \mathbf{m})(\mathbf{x}_n - \mathbf{m})^T \quad (6)$$

Where $\mathbf{m} = \frac{1}{N} \sum_{n=1}^N \mathbf{x}_n = \frac{1}{N} \sum_{k=1}^K N_k \mathbf{m}_k$.

2.1 Decompose the total covariance matrix S_T into *within-class* covariance matrix S_W and *between-class* covariance matrix S_B , and show that S_B has the form: $S_B = \sum_{k=1}^K N_k (\mathbf{m}_k - \mathbf{m})(\mathbf{m}_k - \mathbf{m})^T$

2.2 Write down the *within-class* covariance matrix s_W and *between-class* covariance matrix s_B of the projected D' -dimensional \mathbf{y} -space.

2.3 Many possible choices of criterion can be implemented to decide the weight matrix \mathbf{W} , more than 5 examples are shown in Professor Zhang's ppt. Here, we are using another criterion:

$$J(W) = \frac{\Pi_{diag} S_B}{\Pi_{diag} S_W} \quad (7)$$

Where, $\Pi_{diag} A$ means multiplication of the diagonal elements of matrix A .

Represent $J(\mathbf{W})$ explicitly with \mathbf{W} , S_W and S_B .

2.4 As is stated above, we now want to project the original data space onto a space with D' dimensions, while at the same time trying to maximize $J(\mathbf{W})$ represented by equation (7). Write down the equations that columns of weight matrix \mathbf{W} should obey (which means the selected projection directions).

2.5 As is stated in the problem, we have K classes in all and we are trying to find D' linear 'features' (or projection directions) by maximizing $J(\mathbf{W})$. How many such 'features' at most are we able to find? Give me your reason.

Problem 3: Programming

Feature selection with L1-norm regularization

This programming task seems to be heavy, but don't be afraid. This is actually an easy work as long as you follow the steps and hints provided.

In this problem, let's consider a regularized approach to feature selection in a simple regression context. Unlike fisher criterion, we are not going to reduce dimensions, but just select features from the existed ones. Suppose we have training inputs $X = (\mathbf{x}_1, \dots, \mathbf{x}_n)^T$ and corresponding outputs $\mathbf{y} = (y_1, \dots, y_n)^T$, where $\mathbf{x}_i \in \mathcal{R}^D$ and $y_i \in \mathcal{R}$. We want to train a linear predictor $\hat{y}(\mathbf{x}; \mathbf{w}) = \mathbf{w}^T \phi(\mathbf{x})$ with $\phi(\mathbf{x})$ indicating the M features $\phi(\mathbf{x}) = (\phi_1(\mathbf{x}), \dots, \phi_M(\mathbf{x}))^T$. The objective is regularized least-squares as following:

$$J(\mathbf{w}; \lambda) = \frac{1}{n} \sum_{i=1}^n \frac{1}{2} (y_i - \mathbf{w}^T \phi(\mathbf{x}_i))^2 + \lambda \|\mathbf{w}\|_1 \quad (8)$$

Where $\|\mathbf{w}\|_1$ is the L1-norm:

$$\|\mathbf{w}\|_1 = \sum_{k=1}^M |w_k| \quad (9)$$

Thus, our task is to minimize the regularized objective in order to seek for the optimal parameters $\hat{\mathbf{w}} = \hat{\mathbf{w}}(\lambda)$:

$$\hat{\mathbf{w}} = \operatorname{argmin}_{\mathbf{w} \in \mathcal{R}^M} J(\mathbf{w}; \lambda) \quad (10)$$

Implementation of the L1-norm will lead to a sparse \mathbf{w} . What's more, with λ increasing, more elements of weight vector \mathbf{w} are forced to zero, which means the corresponding features are ignored in the predictor.

Coordinate descent will be used to solve this L1-regularized least-squares problem. In this approach, we adjust one parameter at a time so as to minimize the objective while keeping the remaining parameters fixed:

$$\hat{w}_k = \operatorname{argmin}_{w_k} J(\mathbf{w}; \lambda) \quad (11)$$

With k iterating over all indices $\{1, \dots, M\}$, and repeating the process for many times, parameter \mathbf{w} asymptotically converges to the global minimum of the convex objective (Not familiar with definition of convex? wiki or google).

Since the L1 regularization is not smooth, we can't take derivatives as the differentiable functions to minimize the objective. Instead, we make use of the *subdifferential of a convex function*, which is defined as:

$$\partial f(w) = \{s | f(w + \Delta) \geq f(w) + s\Delta, \forall \Delta \in \mathcal{R}\} \quad (12)$$

This is a set-valued generalization of the normal derivative and reduces to the normal derivative $\partial f(w) = \{\frac{\partial f(w)}{\partial w}\}$ whenever f is differentiable. Taking the absolute value function $f(w) = |w|$ as an example:

$$\partial f(w) = \begin{cases} \{-1\}, & w < 0 \\ [-1, +1], & w = 0 \\ \{+1\}, & w > 0 \end{cases}$$

We will use the following result from non-smooth analysis:

Optimality Condition: \hat{w} is a global minimizer of a convex function $f(w)$ if and only if $0 \in \partial f(\hat{w})$.

For example, the optimality condition $0 \in \partial f(w)$ for the absolute value function $f(w) = |w|$ holds if and only if $w = 0$. Hence, $w = 0$ is the unique global minimizer of $|w|$.

Below, we will guide you through the analysis to solve the L1-norm regularized least-squares problem. 3.1 Show that the sub-differential of $J(\mathbf{w}; \lambda)$ with respect to parameter w_k is:

$$\partial_{w_k} J(\mathbf{w}; \lambda) = (a_k w_k - c_k) + \lambda \partial_{w_k} |w_k| = \begin{cases} \{a_k w_k - (c_k + \lambda)\}, & w_k < 0 \\ [-c_k - \lambda, -c_k + \lambda], & w_k = 0 \\ \{a_k w_k - (c_k - \lambda)\}, & w_k > 0 \end{cases}$$

With:

$$a_k = \frac{1}{n} \sum_{i=1}^n \phi_k^2(\mathbf{x}_i) \quad (13)$$

$$c_k = \frac{1}{n} \sum_{i=1}^n \phi_k(\mathbf{x}_i)(y_i - \mathbf{w}_{-k}^T \phi_{-k}(\mathbf{x}_i)) \quad (14)$$

Where \mathbf{w}_{-k} (respectively ϕ_{-k}) denote the vector of all parameters (features) except for parameter w_k (feature ϕ_k). Observe equations (15) and (16), you can find that a_k 's are non-negative and constant parameters while each c_k depends on all parameters except for the parameter w_k that we are to update. Try to interpret the coefficient c_k .

3.2 Solve the non-smooth optimality condition for \hat{w}_k s.t. $0 \in \partial_{w_k} J(\hat{w}_k)$. It's helpful to consider each of the following cases separately:

- (a) $c_k < -\lambda$
- (b) $c_k \in [-\lambda, +\lambda]$
- (c) $c_k > +\lambda$

In each case, provide a plot $\partial_{w_k} J(w_k)$ versus w_k and label the zero-crossing \hat{w}_k . Then, with the help of those plots, you can express \hat{w}_k as a function of a_k , c_k and λ . Finally, provide a plot of \hat{w}_k versus c_k . What role does the regularization parameter λ play in this context relative to the coefficient c_k ?

Hint: There is no need for you to spend too much time generating those plots required above. Since those plots are just small tools helping you understand the final equations. You can just provide me with your hand-drawn plots, either inserted in your report, or included in the codes' folder.

3.3 We have provided you with training and testing data in the file **least_sq.mat**. Load this file into MATLAB workspace and you will find 4 structures **test**, **train_large**, **train_mid**, **train_small** each containing an $n \times M$ matrix X and a $n \times 1$ vector \mathbf{y} .

For simplicity, we only consider the linear regression, say $\phi(\mathbf{x}) = \mathbf{x}$. Write a MATLAB subroutine based on the skeleton **least_sq_L1.m** to solve for the optimal parameter $\hat{\mathbf{w}}$ given the training data (X, \mathbf{y}) .

Hint: The skeleton code is just a suggestion, you are not strictly required to follow it as long as you can realize the algorithm.

What's more, before writing the code, review the equations derived in 4.2, you may find that the expression for \hat{w}_k considered under different cases can be included into just one simple equation, which can accelerate your code by eliminating "if-else" snippets.

3.4 Complete the skeleton **mainFunc.m** to run your algorithm for a sequence of λ values $[0.01: 0.01: 2.0]$. And we use $\mathbf{w}_0 = (X^T X)^{-1} X^T \mathbf{y}$, the usual least-squares parameter estimation as the initial state for \mathbf{w} . Also, a function **least_sq_multi.m** is provided, you should read the code carefully.

Plot each of the following versus λ :

- (a) the training error $J(\hat{\mathbf{w}}(\lambda); 0)$
- (b) the regularization penalty $\|\hat{\mathbf{w}}(\lambda)\|_1$
- (c) the minimized objective $J(\hat{\mathbf{w}}(\lambda); \lambda)$
- (d) the number of non-zero parameters $\|\hat{\mathbf{w}}(\lambda)\|_0$ (the L0-norm)
- (e) the test error

Run this experiment for each of the three training set **train_large**, **train_mid** and **train_small**, then test on set **test**. Comment on the behavior of each of these quantities as a function of λ . For each training set, what value of λ minimizes the test error? How does this vary with the size of the training set? How could we estimate from the training data the appropriate value of λ to use so as to approximately minimize the test error?

Hint: Above all, we've guided you through the basic idea of optimizing the $L1$ -norm problem. $L1$ -regularization is greatly related to the field of compress sensing or sparse representation which was a hot topic for a few years, to learn more, wiki or google.