Data Mining - Assignment 2

3130000696 黄昭阳

A Walk Through Linear Models

In this problem, you will implement a whole bunch of linear classifiers and compare theirperformance and properties.

We assume our target function is f : [−1, 1] × [−1, 1] → {−1, +1}. In each run, choosea random line in the plane as your target function f, where one side of the line maps to +1and the other maps to -1.

Skeleton code and MATLAB functions including mkdata and plotdata are given for yourconvenience, see the comments in the code for more details. What you need to do is imple-menting each algorithm and write scripts to play with them. Your algorithm implementationsshould be able to handle data in arbitrary dimension. For most of the questions below, youshould repeat the experiments for 1000 times and take the average. See run.m for a scriptexample.

Hint: Do not forget to add the bias constant term!

Perceptron

Implement Perceptron learning algorithm (in perceptron.m), then answer the followingquestions.

(i)

Question:

What is the training error rate and expected testing error rate (we can generatea large set of testing points to estimate test error), when the size of training setis 10 and 100 respectly?

Answer:

Loss function:

$$E(a) = -\sum_{iI_M} a^T x_i y_i$$

Use gradient descent to get a solution:

$$a(k+1) = a(k) + \lambda(k) \sum_{y} y$$

learning rate $\lambda = 1$

```
Size of Training Set = 10:
E_train is 0.000000, E_test is 0.104440.

Size of Training Set = 100;
E_train is 0.000000, E_test is 0.013713.
```

Apparently larger training set will result in less testing error.

(ii)

Question:

What is the average number of iterations before your algorithm converges whenthe size of training set is 10 and 100 respectively ?

Answer:

```
Size of Training Set = 10:
Average number of iterations is 6.020000e+00.

Size of Training Set = 100:
Average number of iterations is 4.339900e+01.
```

(iii)

Question:

What will happen if the training data is not linearly separable (Use mkdata(N,'noisy') to generate non-linearly separable data) ?

Answer:

In fact, It will trap into infinite loop if I've not set limitation as the training error can not be reduced to zero, so I set a upbound to restrict the times of iteration.

```
E_train is 0.100000, E_test is 0.034000.
Average number of iterations is 1000.
```

Through the process of training, I've recorded the best W which makes the training error minimum, and make it as output. Surprisely, it works much better than I expect. You can see the testing error is just about 3.4%。

Linear Regression

Implement Linear Regression (in linear regression.m), then answer the following questions.

Note that we use Linear Regression here to classify data points. This technique is called Linear Regression on indicator response matrix.

(i)

Question:

What is the training error rate and expected testing error rate if the size of training set is 100 ?

Answer:

We can get the optimize equation(Rigde Regression)

$$\min_w \sum_{i=1}^{N}(y_i - x_i^T w) + \lambda \sum_{i=1}^{P} w_i^2$$
$$\downarrow$$
$$\min_w (y - x^T w)^T (y - x^T w) + \lambda w^T w$$

But we should implement linear regression here, the regularization form should be omitted. The formula is transformed as below:

$$\min_w (y - x^T w)^T (y - x^T w)$$

Set gradient to zero, we can get the solution.

$$2x(y - x^T w) = 0$$
$$\downarrow$$
$$w = (xx^T)^{-1} xy$$

To use this method, we should ensure the matrix xx^T is not singular, or we can get its invert matrix.

```
E_train is 0.037850, E_test is 0.047788.
```

(ii)

Question:

What is the training error rate and expected testing error rate if the training datais noisy and not linearly separable (nTrain = 100) ?

Answer:

```
E_train is 0.136720, E_test is 0.061100.
```

(iii)

Question:

Run Linear Regression on dataset poly train.mat. What is the training error rate? What is the testing error rate on poly test.mat ?

Answer:

```
E_train is 0.490000, E_test is 0.549600.
```

(iv)

Question:

Do the following data transformation

(1, x1, x2) → (1, x1, x2, x1x2, x21, x2).

on dataset poly train.mat. After this transformation, what is the training error rate? What is the testing error rate on poly test.mat ?

Answer:

```
E_train is 0.050000, E_test is 0.066000.
```

Logistic Regression

Implement Logistic Regression (in logistic.m), then answer the following questions.Remember the form of Logistic Regression

$$p(y|x) = \frac{1}{1 + e^{-\theta^T x}}$$

Let us assume

$$P(y=1|x;\theta)=h_\theta(x), P(y=0|x;\theta)=1-h_\theta(x),$$

then

$$P(y|x;\theta)=(h_\theta(x))^y(1-h_\theta(x))^{1-y},$$

Now you can learn your logistic model using MLE. Not like Linear Regression, we haveno closed-form solution to solve the MLE problem. So you should use gradient descentto maximize the log-likelihood function iteratively.

Derive the gradient descent update rule, choose a proper learning rate, and repeat the experiment for 100 times to take average.

(i)

Question:

What is the training error rate and expected testing error rate if the size of trainingset is 100 ?

Answer:

Use Newton method to finish gradient descent.

$$\theta^{t+1}=\theta^t-\lambda(\frac{\partial^2 l(\theta)}{\partial\theta\partial\theta^T})^{-1}\frac{\partial l(\theta)}{\partial\theta}$$

According to the given assumption:

$$P(y|x;\theta)=(h_\theta)^y(1-h_\theta)^{1-y}$$
$$p(y|x)=\frac{1}{1+e^{-\theta^T x}}$$
$$l(\theta)=-ln(\prod_{i=1}^{N}P(y_i|x_i;\theta))$$
$$\hat{X}=(1;X)$$

We can induce the corrsponding formula

$$l(\theta) = \sum_{i=1}^{N} -y_i\theta^T\hat{x_i} + ln(1 + e^{\theta^T\hat{x_i}})$$

$$\frac{\partial l(\theta)}{\partial \theta} = -\sum_{i=1}^{N}\hat{x_i}(y_i - h_\theta(\hat{x_i}))$$

$$\frac{\partial^2 l(\theta)}{\partial \theta \partial \theta^T} = \sum_{i=1}^{N}\hat{x_i}\hat{x_i}^T h_\theta(\hat{x_i})(1 - h_\theta(\hat{x_i}))$$

```
E_train is 0.000000, E_test is 0.010700.
```

Large learning rate can reach the best point fast but may result in thresh around the point, in the other hand small learning rate lead to a slow speed to approach the best poitn, so we should find a proper learning rate.

After many hyperparameter tests, I find learning rate $\lambda = 0.5$ and threshold of gradient $||grad||^2 \leq 0.0000001$ will lead to fast convergence with good performance. I've considered the problem that we should break out when the gradient satisfing the threshold or just run our of the iterating limitation. The gradient threshold will not be equivalent in strictly speaking when the distribution of the dataset changed, but if we abandon the threshold the hessian matrix approaches singular.

According to the several faults of the direct gradient descent and Newton method, I finally choose DFP to complete the taks:

$$direction: d_k = -D_k grad_k$$

$$s_k = \lambda_k d_k$$

$$w_{k+1} = w_k + s_k$$

$$grad_{k+1} = -\sum_{i=1}^{N}\hat{x_i}(y_i - h_\theta(\hat{x_i}))$$

$$y_k = grad_{k+1} - grad_k$$

$$D_{k+1} = D_k + \frac{s_k sk^T}{s_k^T y_k} - \frac{D_k y_k y_k^T D_k}{y_k^T D_k y_k}$$

Initialization:$D_0 = I$

(ii)

What is the training error rate and expected testing error rate if the training datais noisy and not linearly separable (nTrain = 100) ?

```
E_train is 0.125905, E_test is 0.050531.
```

Support Vector Machine

Implement Support Vector Machine without introducing slack variables (in svm.m), then answer the following questions.

Hint: Using MATALB built-in function quadprog, you should be able to implementSVM in less than 10 lines of code.

(i)

Question:

What is the training error rate and expected testing error rate if the size of trainingset is 30 ? (Use plotdata to plot your learnt model, hope the graph will help youunderstand SVM as a maximum margin classifier.)

Answer:

According to the document of MATLAB, the function quadprog $x = quadprog(H, f, A, b)$ can solve this optimization problem:

$$\begin{gathered}\min_x \frac{1}{2} x^T H x + f^T x \\ s.t. \\ A \cdot x \leq b \\ Aeq \cdot x = beq \\ lb \leq x \leq ub\end{gathered}$$

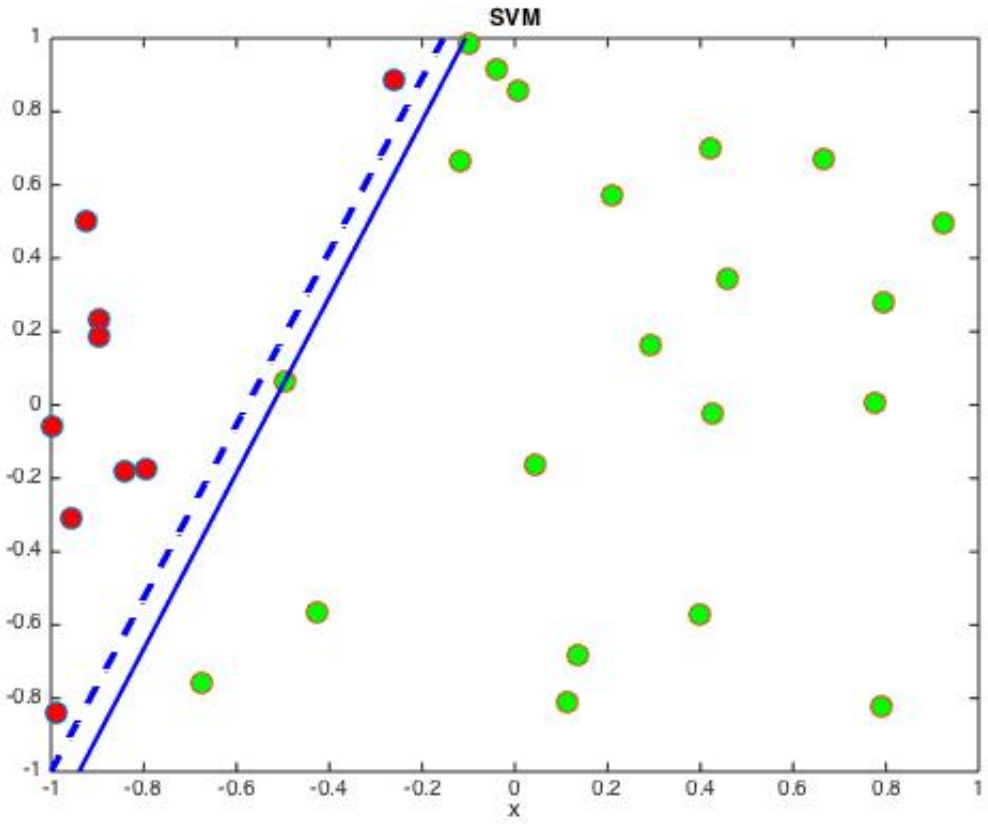
The Support Vector Machine should solve the optimization problem below:

$$\begin{gathered}\min_w \frac{1}{2} ||w||^2 \\ s.t. \\ y_i(w^T \cdot x_i + b) \geq 1\end{gathered}$$

So we transform $y_i(w^T \cdot x_i + b) \geq 1$ to $(y. * X)^T \cdot x \geq 1$(A matrix operation in MATLAB). Then assign value to the parameter of quadfun function:

$$
\begin{aligned}
x &= w \\
H &= I \\
f &= 0 \\
A &= -(y. * X)^T \\
b &= -1
\end{aligned}
$$

Set the size of training dataset be 30, the diagram, training error, test error and support vector number is showed below.




```
E_train is 0.000000, E_test is 0.037141.
```

According to the diagram showed above, we can clearly know SVM will find a plane to seperate two class that maximize the margin ultimately.

(ii)

Question:

What is the training error rate and expected testing error rate if the size of trainingset is 100 ?

Answer:

```
E_train is 0.000000, E_test is 0.011141.
```

(iii)

Question:

For the case nTrain = 100, what is average number of support vectors in yourtrained SVM models?

Answer:

Support vector must satisfy the equation:$|w' \cdot x_i| = 1$, but it's therical and we should take the calculation error into consideration to get experimental data. Set the tolerance be 0.000001, so the samples satisfing $|1 - |w' \cdot x_I|| \leq 0.000001$ is support vectors.

```
E_num is 3.384000.
```

Regularization and Cross-Validation

We now learnt the undesirable consequence of overfitting, just as this quotation says:

If you torture the data for long enough, it will confess to anything.

Regularization and Cross-Validation2 are the two most common used methods to con-trol overfitting. In this problem, we will use these two techniques together to relieve the overfitting problem.

Cross-Validation is a standard technique for adjusting regularization parameters of learn-ing models. We first split the dataset into training set and validation set. Then we computethe validation error on validation set using different values for regularization parameters.Finally, we pick the parameter with the lowest validation error and use it for training ourlearning model. To reduce variability, we can run validation multiple rounds using differentdataset partitions, and the validation results are averaged over all the rounds.

In the following questions, we will use leave-one-out cross-validation to choose regular-ization parameters. As the name suggests, leave-one-out cross-validation (LOOCV) involvesusing a single observation from the original sample as the validation data, and the remainingobservations as the training data. This is repeated such that each observation in the sampleis used once as the validation data.

You may use validation.m as a skeleton.

Ridge Regression and LOOCV

(i)

Question:

What is the λ chosen by LOOCV ?

Answer:

In fact, it's an image vector, so I don't think implementing regularization as $x = \frac{x-\bar{x}}{\sigma}$ for each feature to make them has zero mean and unit variance is reasonable. Instead, implementing regularization as $x_i = \frac{x_i}{||x_i||} \rightarrow x = x - \bar{x}$ is more reliable. But according to the requirement of the assignment, I choose the first method to complete the task .

There is still a problem involving the regularization that some of its variance is zero. The features that have zero variance make no contribution to classifing, so we just don't process them.

Ridge regression optimal equation showed below:

$$\min_w \sum_{i=1}^{N}(y_i - x_i^T w) + \lambda \sum_{i=1}^{P} w_i^2$$
$$\downarrow$$
$$\min_w (y - x^T w)^T (y - x^T w) + \lambda w^T$$

Set the gradient to zero, we can get the solution:

$$-2x(y - x^T w) + 2\lambda w = 0$$
$$\downarrow$$
$$w = (xx^T + \lambda I)^{-1} xy$$

There is a significent problem here that lack of training sample when $\lambda = 0$ will result in the matrix xx^T is singular, so we should use SVD to get the Moore-Penrose pseudo inverse of the matrix, which provided by MATLAB as function pinv(). However pinv() is much slower than left divide, so we use pinv() when $\lambda = 0$, and use left divide otherwise, although it may disturb the criteria.

```
lambda = 100
```

(ii)

Question:

What is $\sum_{i=1}^{m} w_i^2$ with and without regularization (let λ = 0) ?

Answer:

```
lambda = 0
w=1.020477e+00
lambda = 100
w=1.328952e-01
```

Apparently regularization term will restrict the length of vector w to maintain lower variance of the learning model.

(iii)

Question:

What's training/testing error rate with and without regularization?

Answer:

```
lambda = 0
w=1.010186e+00
E_test = 1.260673e-01
lambda = 100
w=3.645480e-01
E_test = 5.976896e-02
```

Logistic Regression

Question:

Implement Logistic Regression with regularization (in logistic r.m), and use LOOCV to tune the regularization parameter λ.

Train your algorithm on digit train.mat and test on digit test.mat. Then report thetraining/testing error rate with and without regularization (let λ = 0). You should also report the λ chosen by LOOCV.

Answer:

Similar to non-regularization logistic regression, I've tried to use direct gradient descent here as the calculation of hessian matrix and its inverse cost much time, but I find that when $\lambda >= 100$ the gradient grows crazily and never converge. Then I find a trick that the product of learning step μ and regularization coefficient λ should be less than 1($\mu * \lambda \leq 1$), otherwise the gradient never converge. After that, I choose Newton method(It cost much time but works properly) to complete the gradient descent task finally. And specailly, When $\lambda = 0$ the hessian matrix is singular according to poor samples, but the pseudoinverse calculation of hessian matrix is so so so so slow and this case should run 199 times, so I choose direct gradient descent.

$$\theta^{t+1} = \theta^t - \lambda(\frac{\partial^2 l(\theta)}{\partial\theta\partial\theta^T})^{-1}\frac{\partial l(\theta)}{\partial\theta}$$

Regularization term will slightly modify the hessian matrix and gradient as below:

$$l(\theta) = \sum_{i=1}^{N} -y_i\theta^T\hat{x_i} + ln(1+e^{\theta^T\hat{x_i}}) + \lambda\sum_{j=1}^{p}|a_j^2|$$

$$\frac{\partial l(\theta)}{\partial\theta} = -\sum_{i=1}^{N}\hat{x_i}(y_i - h_\theta(\hat{x_i})) + \lambda a$$

$$\frac{\partial^2 l(\theta)}{\partial\theta\partial\theta^T} = \sum_{i=1}^{N}\hat{x_i}\hat{x_i}^T h_\theta(\hat{x_i})(1-h_\theta(\hat{x_i})) + \lambda I$$

```
lambda =

   0.100000000000000

E_test = 5.123054e-02
E_train = 0
lambda =

     0

E_test = 7.132094e-02
E_train = 0
```

Similarly, we also introduce DFP to complete the task.

$$direction: d_k = -D_k grad_k$$
$$s_k = rate_k d_k$$
$$w_{k+1} = w_k + s_k$$
$$grad_{k+1} = -\sum_{i=1}^{N} \hat{x_i}(y_i - h_\theta(\hat{x_i})) + \lambda * w_k$$
$$y_k = grad_{k+1} - grad_k$$
$$D_{k+1} = D_k + \frac{s_k sk^T}{s_k^T y_k} - \frac{D_k y_k y_k^T D_k}{y_k^T D_k y_k}$$

Compared to the algorithm above, it will converge when $\lambda \geq 100$, robust to singular hessian matrix and much faster.

SVM with slack variable

Similar to the last question about SVM, we have the formula below.

$$\min_w \frac{1}{2}||w||^2 + C\sum_{i=1}^{n}\xi_i$$
$$s.t.$$
$$y_i(w^T \cdot x_i + b) \geq 1 - \xi_i$$
$$\xi_i \geq 0$$

The quadprog provide the optimization result below.

$$\min_x \frac{1}{2}x^T Hx + f^T x$$
$$s.t.$$
$$A \cdot x \leq b$$
$$Aeq \cdot x = beq$$
$$lb \leq x \leq ub$$

So we can get the corresponding terms.

$$x = \{w; \xi\}$$
$$H = \begin{Bmatrix} I_{P+1*P+1} & 0_{P+1*P} \\ 0_{P*P+1} & 0_{P*P} \end{Bmatrix}$$
$$f = \{0_{P+1*1}; C * 1_{P*1}\}$$
$$A = \begin{Bmatrix} -(y. * X)^T & -I \\ 0 & -I \end{Bmatrix}$$
$$b = \{-1_{P+1*1}; 0_{P*1}\}$$

Results:

```
E_test =

   0.055750878955299
E_train =

   0
num_sc =

    62
```

Bias Variance Trade-off

Let us review the bias-variance decomposition first.

The intuition behind it is straight-forward: if the model is too simple, the learnt functionis biased and does not fit the data. If the model is too complex then it is very sensitive tosmall changes in the data.

If we were able to sample a dataset D infinite many times, we will learn different g(x)for each time, and get an expected hypothesis $\bar{g}(x)$. So bias means the difference betweenthe truth and what you expect to learn. It measures how well our model can approximatethe truth at best.

However, it is impossible to sample the training dataset multiple time, so variance meansthe difference between what you learn from a particular dataset and what you expect tolearn.

Now please answer the following questions:

(a) True of False:

(i) If a learning algorithm is suffering from high bias, adding more training examples will improve the test error significantly.

False.

(ii) We always prefer models with high variance (over those with high bias) as they will able to better fit the training set.

True.

We can do nothing to the models with high bias, but high variance can be reduced through a series of processing to increase generalization of a model.

(iii) A model with more parameters is more prone to overfitting and typically has higher variance.

False.

The factor that determines the complexity of a model, which determines variance of a model, is "Effective number".

eg.

KNN(k=1) has only 1 parameter.

2-features logistic regression has 3 paramters.

According to the observation on the diagram, we can realize that KNN is much complex that logistic regression.

(iv) Introducing regularization to the model always results in equal or better perfor-mance on the training set.

False.

(v) Using a very large value of regularization parameter λ cannot hurt the perfor-mance of your hypothesis.

False.

In the problem of logisitc regression with regularization above, we can find $\lambda \geq 100$ results in the divergence of the gradient with direct gradient descent method.

References

1. 《机器学习》 - 周志华
2. Website:http://blog.csdn.net/itplus/article/details/21896981 (DFP method introduction)
3. Discuss with Chaoqi Wang about the divergence of the gradient problem