Machine Learning, Spring 2018
Homework 5

Due on 23:59 May 16, 2018
Compress all your materials in one file, and send to $cs282\_01@163.com$
with subject "Chinese name+student number+HW5" (In this format please!)

Understanding VC dimension (20 points)

1. In this proof, we assume the labels (y_1, y_2, y_3, y_4) are: -1,-1,+1,-1. Then we get: $sin(\alpha) < 0$, $sin(2\alpha) < 0$, $sin(3\alpha) \geq 0$, $sin(4\alpha) < 0$. In the following, we will show that this implies $sin^2(\alpha) < \frac{1}{2}$ and $sin^2(\alpha) \geq \frac{3}{4}$, which is a contradiction.
   Using the theorem: $sin(2\theta) = 2sin(\theta)cos(\theta)$ and the fact: $sin(4\alpha) < 0$, we have

   $$2sin(2\alpha)cos(2\alpha) = sin(4\alpha) < 0$$

   Since $sin(2\alpha) < 0$ we can divide both sides of this inequality by $2sin(2\alpha)$ to conclude $cos(2\alpha) > 0$. Applying the identity $cos(2\theta) = 1 - 2sin^2(\theta)$ yields: $1 - 2sin^2(\alpha) > 0$, i.e., $sin^2(\alpha) < \frac{1}{2}$.
   However, using the identity $sin(3\theta) = 3sin(\theta) - 4sin^3(\theta)$ and the fact that $sin(3\alpha) \geq 0$ we have:

   $$3sin(\alpha) - 4sin^3(\alpha) = sin(3\alpha) \geq 0$$

   Since $sin(\alpha) < 0$ we can divide both sides of this inequality by $sin(\alpha)$ to conclude $3 - 4sin^3(\alpha) \leq 0$, i.e., $sin^2(\alpha) \geq \frac{3}{4}$.
   Then we get the contradiction, which means the four samples: (1,-1), (2,-1), (3,+1), (4,-1) cannot be shattered.

2. In this proof, we find a construction: α and x_i s.t. the dimension is ∞:
   Consider the set of points given by $x_i = 10^{-i}$, choose any label $\{y_1, \cdots, y_m\}$, then let

   $$\alpha = \pi(1 + \sum_{i=1}^{m} \frac{(1 - y_i)10^i}{2})$$

   Then we get

   $$f(x_j) = sign(sin(10^{-j} \times \pi(1 + \sum_{i=1}^{m} \frac{(1 - y_i)10^i}{2}))) = sign(sin(10^{-j}\pi + \sum_{i=1}^{m}(1 - y_i)10^{i-j}\frac{\pi}{2}))$$

   For any $y_i = 1$, the summation is 0. Also, for any $i > j$, we will be adding an integral number of $\frac{\pi}{2}$ terms to a sin function, which cause no change in final value $f(x_j)$. So we rewrite the formula as follows:

$$
\begin{aligned}
f(x_j) &= sign(sin(10^{-j}\pi + \sum_{i:i<j, y_i=-1}(1-y_i)10^{i-j}\tfrac{\pi}{2})) \\
&= sign(sin(10^{-j}\pi + (1-y_j)\tfrac{\pi}{2} + \sum_{i:i<j, y_i=-1} 2\times 10^{i-j}\tfrac{\pi}{2})) \\
&= sign(sin((1-y_j)\tfrac{\pi}{2} + 10^{-j}\pi + \pi \sum_{i:i<j, y_i=-1} 10^{i-j}))
\end{aligned}
$$

Where we use the fact that $sin(\pi + x) = -sin(x)$. It is easy to show that the summation in the last term and the second term is always less than 1. Therefore, if $y_j = 1$, the argument of the sin function is between 0 and π. Therefore the sin function takes positive values and $f(x_j) = 1 = y_j$. If $y_j = -1$, then the first term becomes π and the argument to the sin function is between π and 2π. The sin function takes a negative value and $f(x_j) = -1 = y_j$. Thus for $\forall j$, $f(x_j) = y_j$. So the set $\{10^{-1}, \cdots 10^{-m}\}$ can be shattered for any value of m. Therefore the VC dimension is ∞.

Understanding Lasso (30 points)

1. the update function:

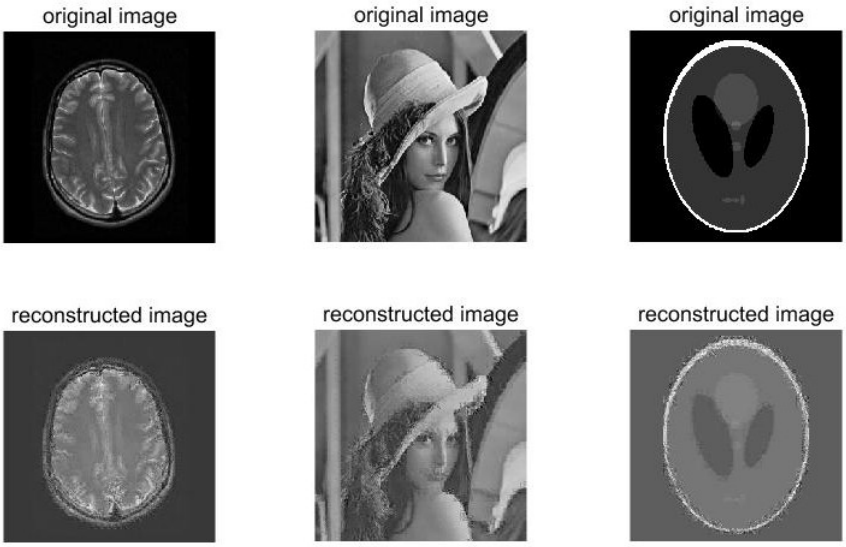
   - x: $x_k = (A^TA + \rho F^TF)^{-1}(A^Tb + F^T(\rho z_{k-1} - y_{k-1}))$
   - z: $z_k = \mathbb{S}_{\frac{\lambda}{\rho}}(Fx_k + \frac{1}{\rho}y_{k-1})$

   derivation: see Fig 1.

2. Complete `glasso.m` as Fig 2, the running result in `testglasso.m` is shown in Fig 3.

3. In demo, the running result is shown as follows:

brain: MSE=0.0025 PSNR=26.0438dB
Lena: MSE=0.0049722 PSNR=23.0346dB
phantom: MSE=0.01004 PSNR=19.9825dB



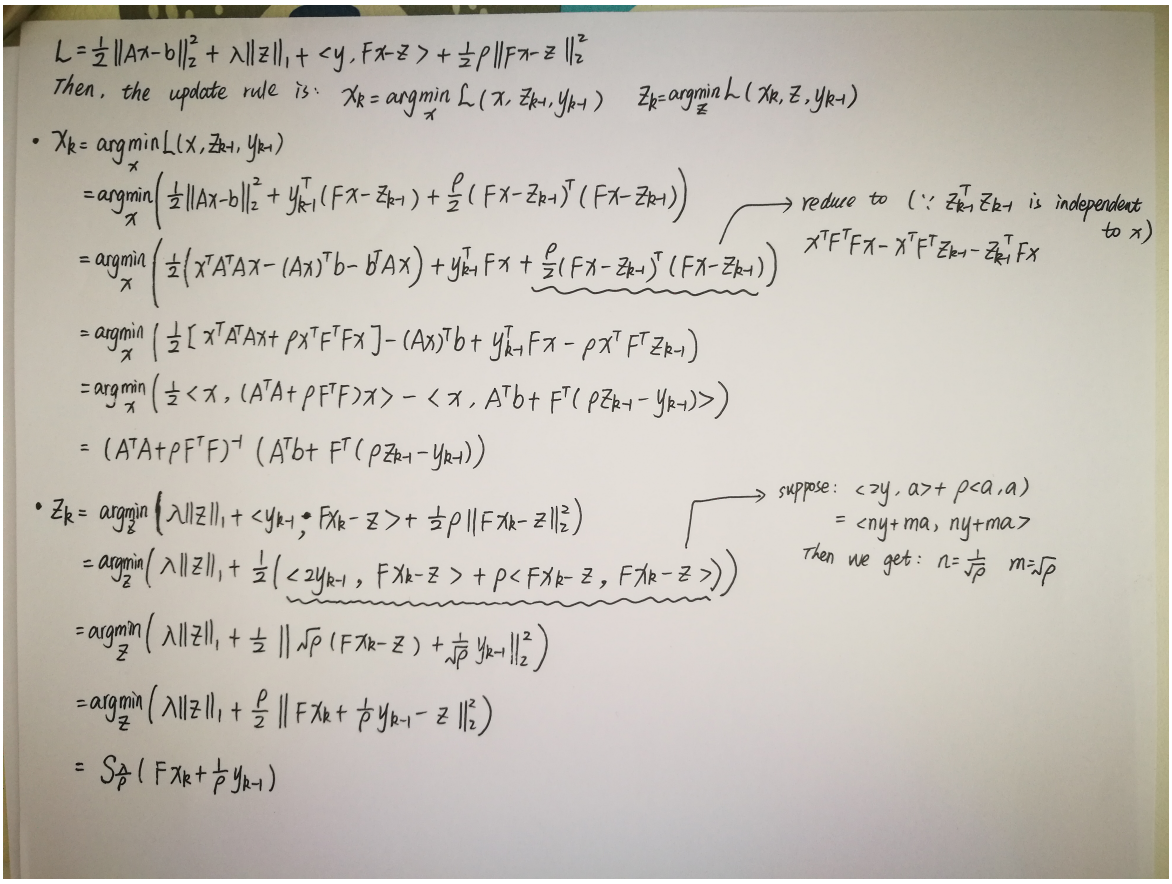


Figure 1: derivation of $\boldsymbol{x}$, $\boldsymbol{z}$

Dual Formulation of the SVM (25 points)

1. The induction is shown in Fig 4.

2. (a) SMO is an iterative algorithm for solving the optimization problem described above. SMO breaks this problem into a series of smallest possible sub-problems, which are then solved analytically. Because of the linear equality constraint involving the Lagrange multipliers α_i, the smallest possible problem involves two such multipliers. Then, for any two multipliers α_1 and α_2, the constraints are reduced to:

$$0 \leq \alpha_1, \alpha_2 \leq C, y_1\alpha_1 + y_2\alpha_2 = k$$

And this reduced problem can be solved analytically: one needs to find a minimum of a one-dimensional quadratic function. k is the negative of the sum over the rest of terms in the equality constraint, which is fixed in each iteration.

So the algorithm proceeds as follows:

i. Find a Lagrange multiplier α_1 that violates the KarushKuhnTucker (KKT) conditions for the optimization problem.

ii. Pick a second multiplier α_2 and optimize the pair α_1, α_2.

iii. Repeat steps 1 and 2 until convergence.

```
% update x, write your formulation
x1 = (A'*A + rho* (F'*F))\(A'* b+F'*(rho*z-y));%x_k+1, z_k, y_k
% update z, write your formulation
z = soft(F*x1+y/rho, lambda/rho); %z_k+1, x_k+1, y_k
```

Figure 2: complete `glasso.m`

```
命令行窗口
stopping criteron: deltaX0.000000, constraint 0.000000
==================================================

relative_error_x =

   4.5290e-04


relative_error_z =

   4.5290e-04

fx >>
```

Figure 3: `testglasso.m` result

When all the Lagrange multipliers satisfy the KKT conditions (within a user-defined tolerance), the problem has been solved. Although this algorithm is guaranteed to converge, heuristics are used to choose the pair of multipliers so as to accelerate the rate of convergence. This is critical for large data sets since there are $\frac{n(n-1)}{2}$ possible choices for α_i and α_j.

The pseudocode of the SMO algorithm is shown in another pdf file named 'pseudocode of the SMO'. And I have implemented(write code) SMO in matlab. The whole source code files is in the folder: ***p3/source code of SMO***. (To test, you only need to run `test.m`)

(b) The accuracy graph is shown in Fig 5. x-axis is passes, y-axis is the accuracy. Notice that, we test 35 cases totally, and for each $i \in \{1, \cdots, 35\}$, the size of training data is $91i$. The last case with $x = 35$ show the accuracy using the whole training data set whose size is 3185. The source code is in folder 'p3/source code of SMO'. (To test, you only need to run `test.m`)

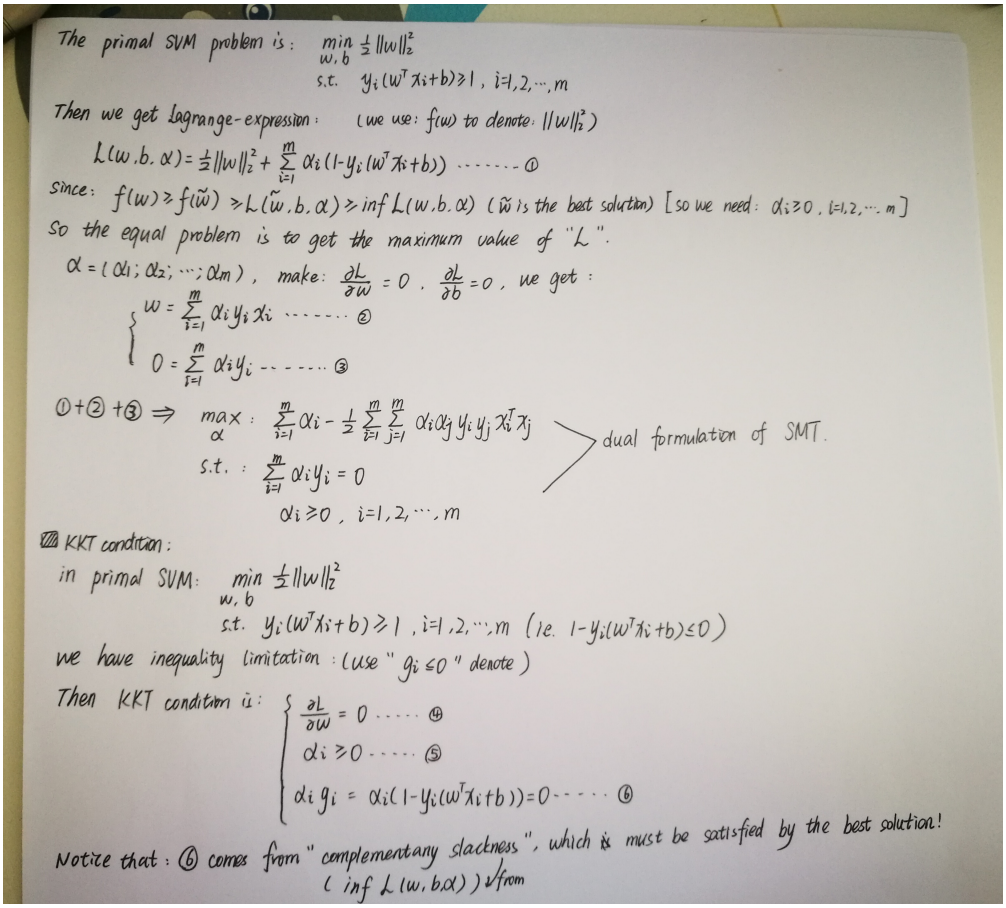


Figure 4: induction of dual and KKT conditions

Kernel function (25 points)

1. The mapping function from $\mathbb{R}^2$ to $\mathbb{R}^3$ is: $\phi(\boldsymbol{x}) = \phi(x^1, x^2) = (x^1, x^2, \frac{(x^1)^2+(x^2)^2-5}{3})$. Then the kernel function is:

$$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \phi(\boldsymbol{x}_i)^T \phi(\boldsymbol{x}_j) = < \boldsymbol{x}_i, \boldsymbol{x}_j > + \frac{(\|\boldsymbol{x}_i\|_2^2 - 5) \cdot (\|\boldsymbol{x}_j\|_2^2 - 5)}{9}$$

2. Using this alternative mapping function, the data in the new feature space $\mathbb{R}^3$ looks like:
data points labeled negatively ("-1")

$$\left\{ \begin{pmatrix} 1 \\ 1 \\ -1 \end{pmatrix}, \begin{pmatrix} 1 \\ -1 \\ -1 \end{pmatrix}, \begin{pmatrix} -1 \\ 1 \\ -1 \end{pmatrix}, \begin{pmatrix} -1 \\ -1 \\ -1 \end{pmatrix} \right\},$$

data points labeled positively ("$+1$")

$$\left\{ \begin{pmatrix} 2 \\ 2 \\ 1 \end{pmatrix}, \begin{pmatrix} 2 \\ -2 \\ 1 \end{pmatrix}, \begin{pmatrix} -2 \\ 2 \\ 1 \end{pmatrix}, \begin{pmatrix} -2 \\ -2 \\ 1 \end{pmatrix} \right\},$$

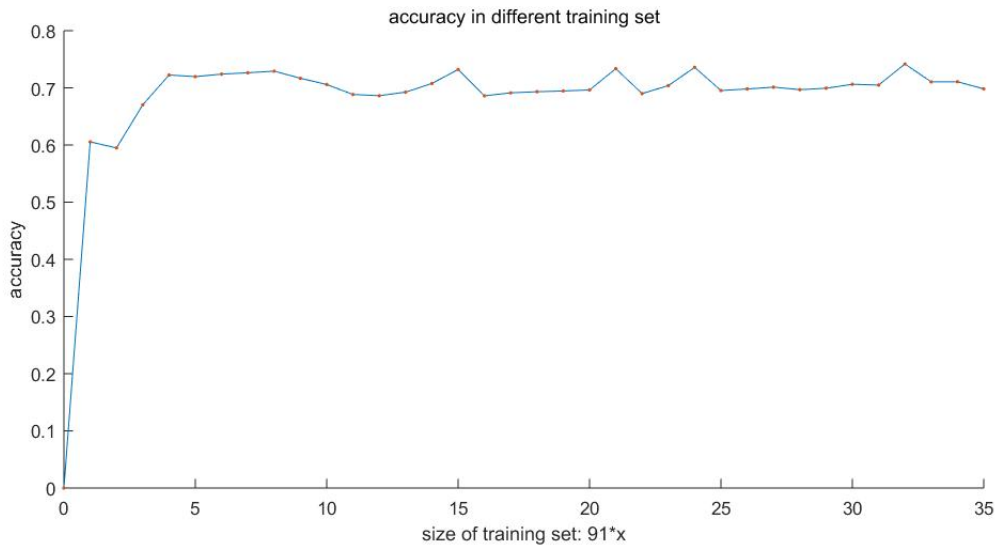


Figure 5: the accuracy graph

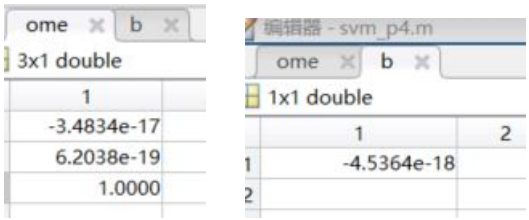
Then the SVM problem is:

$$\begin{aligned} &\min_{\omega,b} \frac{1}{2}\|\omega\|_2^2 \\ &\text{s.t. } y_i(\omega^T\phi(\boldsymbol{x}_i)+b) \geq 1, \quad i=1,\cdots,m. \end{aligned} \tag{1}$$

And its equal dual problem is:

$$\begin{aligned} &\max_{\alpha} \sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j \kappa(\boldsymbol{x}_i,\boldsymbol{x}_j) \\ &\text{s.t.} \sum_{i=1}^{m}\alpha_i y_i = 0, \\ &\quad \alpha_i \geq 0, \quad i=1,\cdots,m. \end{aligned} \tag{2}$$

When we use matlab to solve this convex problem (the source code I wrote can be found in the folder 'p4/smv p4.m'), we get such results of ω and b:



It's more like $\omega = (0,0,1)$ and $b = 0$. Then we get the final separating hyperplane expression:

$$h = sign(\|\boldsymbol{x}\|_2^2 - 5), \boldsymbol{x} \in \mathbb{R}^2$$

The graph is shown in Fig 6.

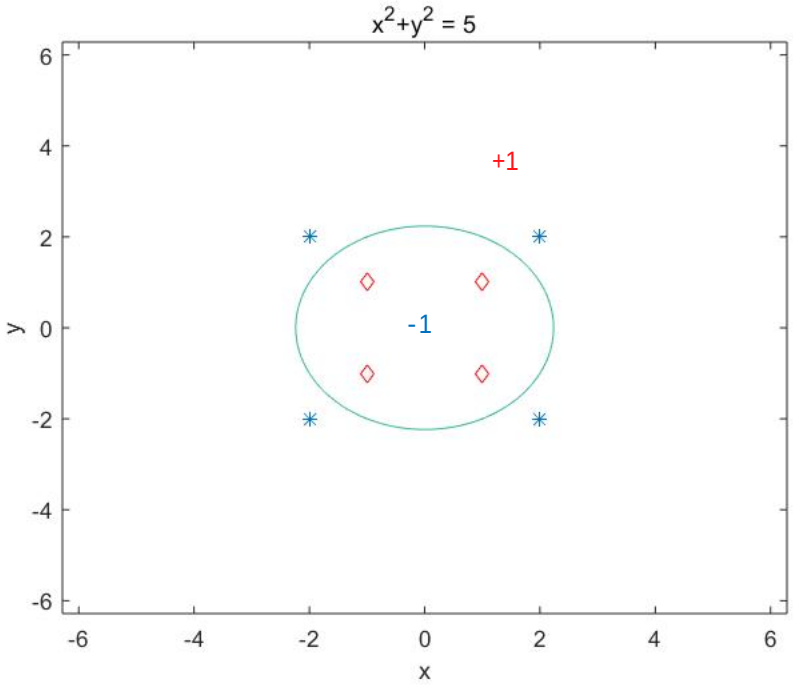


Figure 6: separating hyperplane and data