Project 1: Minimax Probability Machine

周亦涵 2020012853 未央-水木01

Problem1

> *Prove the following equation:*
>
> $$\sup_{D_n} P_{x\sim D_n}(w^T x \geq b) = \frac{1}{1+d^2} \tag{74}$$
>
> *where*
>
> $$d^2 = \inf_{x:w^T x\geq b}(x-\mu_n)^T\Sigma_n^{-1}(x-\mu_n) = \frac{(b-w^T\mu_n)^2}{w^T\Sigma_n w} \tag{75}$$

We first prove equation(2):

Denote
$\tilde{w} = \Sigma_n^{-1/2}(x-\mu_n),\ c^T = w^T\Sigma_n^{1/2}, a = b - w^T\mu_n,\ d^2 = \inf_{x:w^Tx\geq b}(x-\mu_n)\Sigma_n^{-1}(x-\mu_n)$
. Then, we have

$$d^2 = \inf_{c^T\tilde{w}\geq a}\tilde{w}^T\tilde{w} \tag{76}$$

So we can corm the Lagrangian:

$$L(\tilde{w},\lambda) = \tilde{w}^T\tilde{w} + \lambda(a - c^T\tilde{w}) \tag{77}$$

which is to be maximized with respect to $\lambda \geq 0$ and minimized with respect to $\tilde{w}$.

$$\because \frac{\partial L(\tilde{w},\lambda)}{\partial\tilde{w}} = 2\tilde{w} - \lambda c = 0 \tag{78}$$
$$\therefore 2\tilde{w} = \lambda c \tag{79}$$
$$\because \frac{\partial L(\tilde{w},\lambda)}{\partial\lambda} = 0 \tag{80}$$
$$\therefore a = c^T\tilde{w} \tag{81}$$
$$\therefore 2c^T\tilde{w} = \lambda c^T c = 2a \tag{82}$$
$$\therefore \lambda = \frac{2a}{c^Tc} \tag{83}$$
$$\therefore \tilde{w} = \frac{\lambda c}{2} = \frac{ac}{c^Tc} \tag{84}$$

Under these circumstances, we have

$$
\begin{aligned}
d^2 = \inf_{c^T\tilde{w}\geq a} \tilde{w}^T\tilde{w} = \frac{a^2}{c^Tc} \\
\because c^T = w^T\Sigma_n^{1/2},\ a = b - w^T\mu_n \\
\therefore c^Tc = w^T\Sigma_n^{1/2}\Sigma_n^{1/2}w = w^T\Sigma_n w \\
\therefore d^2 = \frac{(b-w^T\mu_n)^2}{w^T\Sigma_n w}
\end{aligned} \tag{85}
$$

Then let's prove equation(1):

D_n has mean μ_n and covariace matrix Σ_n, so d^2 is the expression of the squared distance from μ_n to the set of negative class. This is a multivariate generalization of Chebyshev's inequality, with the tight multivariate one-sided Chebyshev bound being:

$$
\sup_{X\sim(M,\Gamma)} P(X > M_{e+\delta}) = \frac{1}{1+d^2} \tag{86}
$$

where $d^2 = minimize\ x^T\Gamma^{-1}x$, subject to $x \geq M_\delta$. Here, $M = \mu_n, \Gamma = \Sigma_n$

If $\Gamma^{-1}M_\delta \geq 0$, then the tight bound is expressible in closed form:
$\sup_{X\sim(M,\Gamma)} P(X > M_{e+\delta}) = \frac{1}{1+M_\delta^T\Gamma^{-1}M_\delta}$.

Problem2

> *Prove 1:*
>
> $$
> \max_{\kappa,w,b} \kappa\ s.t.\ \ w^T\mu_p - \kappa\sqrt{w^t\Sigma_p w} \leq b \leq w^T\mu_n + \kappa\sqrt{w^T\Sigma_n w} \tag{87}
> $$

Proof 1:

We have our optimization goal:

$$
\begin{aligned}
&\min_{\epsilon,w,b} \epsilon && (88) \\
s.t.\ \ &\sup_{D_p} P_{x\sim D_p}(w^Tx \leq b) \leq \epsilon && (89) \\
&\sup_{D_n} P_{x\sim D_n}(w^Tx \geq b) \leq \epsilon. && (90)
\end{aligned}
$$

Using the conclusion from problem 1, our optimization goal becomes:

$$
\begin{aligned}
\min_{\epsilon,w,b} \epsilon\ s.t.\ \ -b + w^T\mu_p \geq \kappa(\epsilon)\sqrt{w^T\Sigma_p w} && (91) \\
b - w^T\mu_n \geq \kappa(\epsilon)\sqrt{w^T\Sigma_n w} && (92)
\end{aligned}
$$

Since $\kappa = \sqrt{\frac{1-\epsilon}{\epsilon}} = \sqrt{\frac{1}{\epsilon} - 1}$ decreases as ϵ increases, so $\min_{\epsilon,w,b}\epsilon$ is equavalent to $\max_{w,b}\kappa$

From (18):

$$w^T\mu_p - \kappa\sqrt{w^T\Sigma_p w} \geq b \tag{93}$$

From (19):

$$b \geq w^T\mu_n + \kappa\sqrt{w^T\Sigma_n w} \tag{94}$$

Consider (20) & (21) simultaneously:

$$w^T\mu_p - \kappa\sqrt{w^T\Sigma_p w} \geq b \geq w^T\mu_n + \kappa\sqrt{w^T\Sigma_n w} \tag{95}$$

So our goal becomes:

$$\max_{\kappa,w,b} \ \kappa \ s.t. \ w^T\mu_p - \kappa\sqrt{w^T\Sigma_p w} \geq b \geq w^T\mu_n + \kappa\sqrt{w^T\Sigma_n w} \tag{96}$$

Proof 2:

In equation (23), since we have to find a b that maximizes κ under constraints $f_1(\kappa, w) \leq b \leq f_2(\kappa, w)$, with f_1 and f_2 independent of b, so b doesn't affect the value of κ, we can eliminate the parameter b. Also, because we want to maximize κ, the inequalities will become equalities at the optimum, so the optimal value of b:

$$b_* = w_*^T\mu_p - \kappa_*\sqrt{w_*^T\Sigma_p w_*} = w_*^T\mu_n + \kappa_*\sqrt{w_*^T\Sigma_n w_*} \tag{97}$$

Thus our goal becomes:

$$\max_{\kappa,w} \ \kappa \ s.t. \ w^T\mu_p - \kappa\sqrt{w^T\Sigma_p w} \geq w^T\mu_n + \kappa\sqrt{w^T\Sigma_n w} \tag{98}$$

Proof 3:

From (24): consider the constraint:

$$w^T\mu_p - \kappa\sqrt{w^t\Sigma_p w} \geq w^T\mu_n + \kappa\sqrt{w^T\Sigma_n w} \tag{99}$$

$$\Leftrightarrow w^T(\mu_p - \mu_n) \geq \kappa(\sqrt{w^T\Sigma_n w} + \sqrt{w^T\Sigma_p w}) \tag{100}$$

$$\Leftrightarrow \frac{w^T(\mu_p - \mu_n)}{\kappa} \geq \sqrt{w^T\Sigma_n w} + \sqrt{w^T\Sigma_p w} \tag{101}$$

(28) implies that $w^T(\mu_p - \mu_n) \geq 0$. Also, let's multiply the formula by a positive value $s \in R_+$, we have:

$$\frac{(sw)^T(\mu_p - \mu_n)}{\kappa} \geq \sqrt{(sw)^T\Sigma_n(sw)} + \sqrt{(sw)^T\Sigma_p(sw)} \tag{102}$$

So if w satisfies the inequality, sw also does. For convenience, we can restrict w to satisfy $w^T(\mu_p - \mu_n) = 1$, and our goal becomes:

$$\begin{gathered}\max_{\kappa,w} \ \kappa \ s.t. \ \frac{1}{\kappa} \geq \sqrt{w^T\Sigma_n w} + \sqrt{w^T\Sigma_p w} \\ w^T(\mu_p - \mu_n) = 1\end{gathered} \tag{103}$$

Proof 4:

From (35), we can eliminate κ: when κ is maximized, $\frac{1}{\kappa}$ is minimized, so $\sqrt{w^T\Sigma_n w} + \sqrt{w^T\Sigma_p w}$ is also minimized. Similarly, κ is optimal when the formula becomes an equation, that is

$$\kappa_* = 1/(\sqrt{w_*^T\Sigma_n w_*} + \sqrt{w_*^T\Sigma_p w_*}) \tag{104}$$

And our goal becomes:

$$\min_w \ \sqrt{w^T\Sigma_n w} + \sqrt{w^T\Sigma_p w} \ s.t. \ w^T(\mu_p - \mu_n) = 1 \tag{105}$$

Compute the optimal b_*:

From (31), we can calculate w_* by solving a convex optimization problem. Denote $w = w_o + Fu$, where $u \in R^{n-1}$, $w_0 = (x-y)/||(x-y)||^2$,$x \sim D_p$, $y \sim D_n$, $F \in R^{n\times(n-1)}$ being an orthogonal matrix whose columns span the subspace of vectors orthogonal to $\mu_p - \mu_n$. Then our goal becomes:

$$\min_u ||\Sigma_x^{1/2}(w_0 + Fu)||_2 + ||\Sigma_y^{1/2}(w_o + Fu)||_2 \tag{106}$$

By taking iterative least-squares approach, we can get our optial solution w_*, so from (31), we have $\kappa_* = 1/(\sqrt{w_*^T\Sigma_n w_*} + \sqrt{w_*^T\Sigma_p w_*})$, and substitute w_* and κ_* into (24), we can compute the optimal b_*:

$$b_* = w_*^T\mu_n + \sqrt{w_*^T\Sigma_n w_*}/(\sqrt{w_*^T\Sigma_n w_*} + \sqrt{w_*^T\Sigma_p w_*}) \tag{107}$$

which is equalavent to:

$$b_* = w_*^T\mu_p - \sqrt{w_*^T\Sigma_p w_*}/(\sqrt{w_*^T\Sigma_n w_*} + \sqrt{w_*^T\Sigma_p w_*}) \tag{108}$$

Problem3

The code is submitted in `Code.ipynb`, `Problem3` part.

The SOCP problem we want to solve:

$$\min_w \ \sqrt{w^T\Sigma_p w} + \sqrt{w^T\Sigma_n w} \ s.t. \ w^T(\mu_p - \mu_n) = 1 \tag{109}$$

We use the package `cvxpy` to find the solution w_*, b_*. Then we define the MPM classifier. The accuracies and average errors for three datasets over 10 runs are listed below:

```
Dataset: pima
    average accuracy: 0.7922077922077921
    average error: 0.6824541566170195
Dataset: breast_cancer
    average accuracy: 0.9768115942028986
    average error: 0.15646313766701464
Dataset: sonar
    average accuracy: 0.7761904761904763
    average error: 0.35455981677988435
MPM running time: 1.7712400830005208
```

Dataset	Average Test Accuracy	Average Error
pima	0.7922	0.6825
breast_cancer	0.9768	0.1565
sonar	0.7762	0.3546

Next, we run LDA, Logistic Regression (LR) and SVM on the three datasets and compare the average test accuracy and the total running time:

```
Logistic Regression
    Dataset: pima
        average accuracy: 0.7597402597402597
    Dataset: breast_cancer
        average accuracy: 0.9695652173913045
    Dataset: sonar
        average accuracy: 0.7761904761904761
    Running time: 1.7932287089988677
LDA
    Dataset: pima
        average accuracy: 0.7909090909090909
    Dataset: breast_cancer
        average accuracy: 0.9623188405797102
    Dataset: sonar
        average accuracy: 0.7571428571428572
    Running time: 0.48981866699978127
SVM
    Dataset: pima
        average accuracy: 0.7740259740259741
    Dataset: breast_cancer
        average accuracy: 0.955072463768116
    Dataset: sonar
        average accuracy: 0.8
    Running time: 1.161992499999542
```

Dataset	MPM	LDA	LR	SVM
`pima`	0.7922	0.7909	0.7597	0.7740
`breast_cancer`	0.9768	0.9623	0.9696	0.9551
`sonar`	0.7762	0.7571	0.7762	0.8000
Total Running Time	1.7712	0.4898	1.7932	1.1620

From the above results, we can conclude that MPM has the highest accuracy in the first two datasets! For `sonar` particularly, SVM has a prominent performance, while MPM still performs better than LDA and Logistic Regression.

In terms of the running speed, LDA > SVM > MPM > LR. MPM's running speed is similar with LDA, while it has a better performance. LDA is fast in cost of some accuracy. We can't have the highest speed and the highest accuracy simultaneously, it's our choice to balance the two norm.

Problem4

- **Find the best κ:**

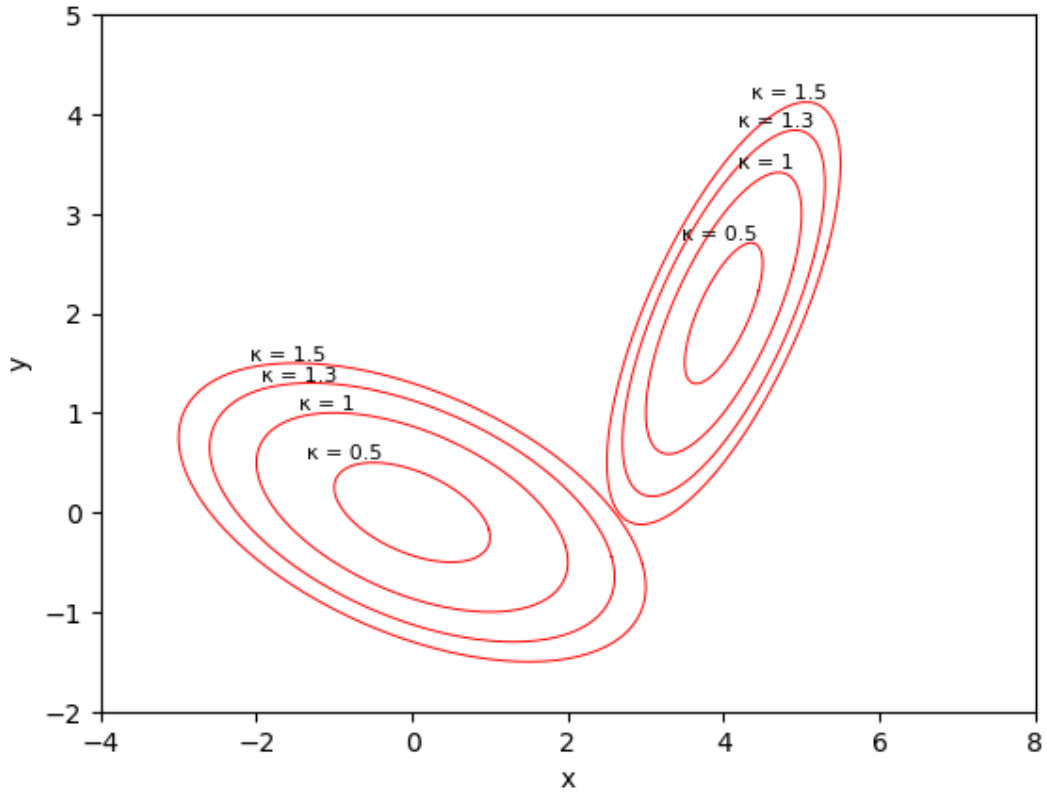
Solution of κ_* above is: $\kappa_* = 1/(\sqrt{w_*^T \Sigma_n w_*} + \sqrt{w_*^T \Sigma_p w_*})$.

First let's use the same `cvxpy` algorithm to calculate the optimal theoretical κ with the code below:

```
# Define the parameters
mu_1 = np.array([0,0])
mu_2 = np.array([4,2])
Sigma_1 = np.array([[4,-1],[-1,1]])
Sigma_2 = np.array([[1,1],[1,2]])
dim = 2

# Define the variables
w = cp.Variable(dim)

# Define the objective function
## 计算协方差阵的1/2
evalue_1, evector_1 = np.linalg.eig(Sigma_1)
half_Sigma_1 = evector_1 @ np.diag(np.sqrt(evalue_1)) @ evector_1.T
evalue_2, evector_2 = np.linalg.eig(Sigma_2)
half_Sigma_2 = evector_2 @ np.diag(np.sqrt(evalue_2)) @ evector_2.T
## 送入目标函数
goal = cp.norm(half_Sigma_1 @w,2)+cp.norm(half_Sigma_2@w,2)

# Define the constraint
soc_constraints = [w @ (mu_2-mu_1) == 1]

# Define the problem
prob = cp.Problem(cp.Minimize(goal),soc_constraints)

# Solve the problem
prob.solve()
# Print the result
w_opt = w.value
k_opt = 1/(np.sqrt(w_opt @ Sigma_1 @ w_opt.T)+np.sqrt(w_opt @ Sigma_2 @ w_opt.T))
print("Optimal Kappa:",k_opt)
```

And the result is $\kappa = 1.512154700499445$.

Now let's draw different ellipsoids with different κ to find the best one. The output image is as following:



After trying different κs, 1.5 is the optimal value, which is very close to the κ_* we have calculated!

- **The relation between MPM and FDA**

MPM and FDA both want to find a discriminant hyperplane, but they have different optimization goals. MPM is a binary classification algorithm that aims to minimize the maximum misclassification probability, while FDA is a statistical technique that aims to reduce dimension by finding a linear combination of features that maximizes the separation between classes while minimizing the within-class scatter, so that classification isn't its main goal. Specifically speaking, their optimization goals are:

$$\begin{aligned} \max_a \ \kappa_{FDA}(\mathbf{a}) &= \frac{|a^T(\bar{x}-\bar{y})|}{\sqrt{a^T\Sigma_x a + a^T\Sigma_y a}} \\ \max_a \ \kappa_{MPM}(\mathbf{a}) &= \frac{|a^T(\bar{x}-\bar{y})|}{\sqrt{a^T\Sigma_x a} + \sqrt{a^T\Sigma_y a}} \end{aligned} \tag{110}$$

The FDA criterion function is built upon the intuition that “separation” is useful, and upon the desideratum of computational efficiency by simpling solving a generalized eigenvalue problem. Although the goal is different, MPM has a similar form with FDA, so its algorithm has similar complexity.