Project 1: Minimax Probability Machine

April 2023

1 Introduction

In the class, we have learned many linear methods including LDA (linear discriminative analysis), Logistic Regression, SVM, etc. In this project, you will learn a new variant of the linear model called **Minimax Probability Machine** [1] (or MPM in abbreviation). We will take you through step by step about the deduction, and you need to write a program to implement it, and compare its performance with vanilla linear methods.

2 Preliminaries

All the following deductions can be found in the original paper[1, 2]. Here we present the main structure of the deduction and ask you questions to help understand it. Note that you need to answer in *this* document's notation system. (i.e. parameter $\boldsymbol{w}$ instead of $\boldsymbol{a}$)

Suppose we want to use a linear plane to separate two classes of samples, namely we are solving a binary classification problem. The distribution of positive class $\mathcal{D}_p$ and negative class $\mathcal{D}_n$ are both unknown distributions supported on $\mathbb{R}^d$. But we know the mean and covariance for both distributions as

$$\boldsymbol{\mu}_p = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}_p}[\boldsymbol{x}], \quad \boldsymbol{\Sigma}_p = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}_p}[(\boldsymbol{x}-\boldsymbol{\mu}_p)(\boldsymbol{x}-\boldsymbol{\mu}_p)^\top],$$
$$\boldsymbol{\mu}_n = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}_n}[\boldsymbol{x}], \quad \boldsymbol{\Sigma}_n = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}_n}[(\boldsymbol{x}-\boldsymbol{\mu}_n)(\boldsymbol{x}-\boldsymbol{\mu}_n)^\top].$$

2.1 Part 1.

We want to determine a linear classifier parametrized by $\boldsymbol{w} \in \mathbb{R}^d$ and $b \in \mathbb{R}$ to achieve the best performance. Denote the error rate as ϵ, this means we want to solve the following optimization problem

$$\min_{\epsilon,\boldsymbol{w},b} \quad \epsilon \tag{1}$$
$$s.t. \quad \sup_{\mathcal{D}_p} \mathcal{P}_{\boldsymbol{x}\sim\mathcal{D}_p}(\boldsymbol{w}^\top\boldsymbol{x} \le b) \le \epsilon \tag{2}$$
$$\sup_{\mathcal{D}_n} \mathcal{P}_{\boldsymbol{x}\sim\mathcal{D}_n}(\boldsymbol{w}^\top\boldsymbol{x} \ge b) \le \epsilon. \tag{3}$$

Here, the operator sup is taken over all probability distributions $\mathcal{D}_p$ that has mean as $\boldsymbol{\mu}_p$ and covariance as $\boldsymbol{\Sigma}_p$, similarly for $\mathcal{D}_n$.

Problem 1 (10 pt). Please prove the following equation

$$\sup_{\mathcal{D}_n} \mathcal{P}_{\boldsymbol{x}\sim\mathcal{D}_n}(\boldsymbol{w}^\top \boldsymbol{x} \geq b) = \frac{1}{1+d^2},$$

where

$$d^2 = \inf_{\boldsymbol{x}:\boldsymbol{w}^\top \boldsymbol{x}\geq b} (\boldsymbol{x}-\boldsymbol{\mu}_n)\boldsymbol{\Sigma}_n^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_n) = \frac{(b-\boldsymbol{w}^\top \boldsymbol{\mu}_n)^2}{\boldsymbol{w}^\top \boldsymbol{\Sigma}_n \boldsymbol{w}}. \tag{4}$$

2.2 Part 2.

Using the conclusion from problem 1, we can boil (3) down into

$$\epsilon \geq \frac{1}{1+d^2} \Longrightarrow d \geq \sqrt{\frac{1-\epsilon}{\epsilon}}.$$

Plug in (4), we know

$$b - \boldsymbol{w}^\top \boldsymbol{\mu}_n = d\sqrt{\boldsymbol{w}^\top \boldsymbol{\Sigma}_n \boldsymbol{w}} \geq \kappa(\epsilon)\sqrt{\boldsymbol{w}^\top \boldsymbol{\Sigma}_n \boldsymbol{w}}, \quad \kappa(\epsilon) = \sqrt{\frac{1-\epsilon}{\epsilon}} \tag{5}$$

we can take the negative of $\boldsymbol{w}$ and b to get similar condition for positive samples. This further transforms our optimization problem into

$$\min_{\epsilon,\boldsymbol{w},b} \quad \epsilon \quad s.t. \quad -b + \boldsymbol{w}^\top \boldsymbol{\mu}_p \geq \kappa(\epsilon)\sqrt{\boldsymbol{w}^\top \boldsymbol{\Sigma}_p \boldsymbol{w}} \tag{6}$$

$$b - \boldsymbol{w}^\top \boldsymbol{\mu}_n \geq \kappa(\epsilon)\sqrt{\boldsymbol{w}^\top \boldsymbol{\Sigma}_n \boldsymbol{w}} \tag{7}$$

Problem 2 (20 pt). Please prove that the optimization above is equivalent to all the problems below. (hint: transform one by one)

- $$\max_{\kappa,\boldsymbol{w},b} \quad \kappa \quad s.t. \quad \boldsymbol{w}^\top \boldsymbol{\mu}_p - \kappa\sqrt{\boldsymbol{w}^\top \boldsymbol{\Sigma}_p \boldsymbol{w}} \geq b \geq \boldsymbol{w}^\top \boldsymbol{\mu}_n + \kappa\sqrt{\boldsymbol{w}^\top \boldsymbol{\Sigma}_n \boldsymbol{w}}. \tag{8}$$

- $$\max_{\kappa,\boldsymbol{w}} \quad \kappa \quad s.t. \quad \boldsymbol{w}^\top \boldsymbol{\mu}_p - \kappa\sqrt{\boldsymbol{w}^\top \boldsymbol{\Sigma}_p \boldsymbol{w}} \geq \boldsymbol{w}^\top \boldsymbol{\mu}_n + \kappa\sqrt{\boldsymbol{w}^\top \boldsymbol{\Sigma}_n \boldsymbol{w}}. \tag{9}$$

- $$\max_{\kappa,\boldsymbol{w}} \quad \kappa \quad s.t. \quad \frac{1}{\kappa} \geq \sqrt{\boldsymbol{w}^\top \boldsymbol{\Sigma}_p \boldsymbol{w}} + \sqrt{\boldsymbol{w}^\top \boldsymbol{\Sigma}_n \boldsymbol{w}} \tag{10}$$

  $$\boldsymbol{w}^\top (\boldsymbol{\mu}_p - \boldsymbol{\mu}_n) = 1. \tag{11}$$

- $$\min_{\boldsymbol{w}} \quad \sqrt{\boldsymbol{w}^\top \boldsymbol{\Sigma}_p \boldsymbol{w}} + \sqrt{\boldsymbol{w}^\top \boldsymbol{\Sigma}_n \boldsymbol{w}} \quad s.t. \quad \boldsymbol{w}^\top (\boldsymbol{\mu}_p - \boldsymbol{\mu}_n) = 1. \tag{12}$$

And give the equation for how to compute the optimal b^* in (8) from optimal solution $\boldsymbol{w}^*$ in (12).

3 Solving Problem

Until (12), we finally transform our problem into a single variable optimization problem.

$$\min_{\boldsymbol{w}} \quad \sqrt{\boldsymbol{w}^\top \boldsymbol{\Sigma}_p \boldsymbol{w}} + \sqrt{\boldsymbol{w}^\top \boldsymbol{\Sigma}_n \boldsymbol{w}} \quad s.t. \quad \boldsymbol{w}^\top (\boldsymbol{\mu}_p - \boldsymbol{\mu}_n) = 1.$$

This is a famous convex optimization problem known as SOCP (second order cone program). All the variables are either known or parameters to learn.

Problem 3 (50 pt). Find ways to solve this SOCP optimization problem. (Hint: there are already some packages to solve this problem, or you can implement the iterative least square approach from table 1 in [2].) Then use the solution $\boldsymbol{w}^*, b^*$ from this optimization to form a classifier.

Compare its performance with the original LDA, Logistic Regression, and SVM on the datasets we provide (including breastcancer, pima, sonar). Each time you should randomly partition it into 90% training set and 10% test set, and repeat that procedure 10 times to get the average test accuracy. Also report the average MPM's guaranteed error ϵ in line (1). You should achieve comparable results as in paper[1].

Also, measure its running speed compared with these methods. What do you find?

4 Understanding

Suppose $\mathcal{D}_p \sim \mathcal{N}(\boldsymbol{\mu}_p, \boldsymbol{\Sigma}_p), \mathcal{D}_n \sim \mathcal{N}(\boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n)$ are both Gaussian distribution, and

$$\boldsymbol{\mu}_1 = (0,0), \boldsymbol{\mu}_2 = (4,2), \boldsymbol{\Sigma}_1 = \begin{bmatrix} 4 & -1 \\ -1 & 1 \end{bmatrix}, \boldsymbol{\Sigma}_2 = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}.$$

Problem 4 (20 pt). Read section 2.4 of paper[2], draw ellipsoids

$$\mathcal{E}_p(\kappa) = \{\boldsymbol{x} = \boldsymbol{\mu}_p + \boldsymbol{\Sigma}_p^{1/2}\boldsymbol{u} : \|\boldsymbol{u}\| \leq \kappa\},$$

$$\mathcal{E}_n(\kappa) = \{\boldsymbol{x} = \boldsymbol{\mu}_n + \boldsymbol{\Sigma}_n^{1/2}\boldsymbol{u} : \|\boldsymbol{u}\| \leq \kappa\}$$

with different κ. What is the smallest κ^* that makes $\mathcal{E}_p(\kappa^*) \cap \mathcal{E}_p(\kappa^*) \neq \varnothing$? Compare it with the solution of (9). Read section 2.7 of paper[2], briefly explain the relation between MPM and FDA (Fisher Discriminative analysis).

5 Bouns: Kernel Version.

It is interesting to extend the linear model's capacity by simply changing a kernel. We always want to use a linear model with a non-linear kernel to leverage the model's capacity.

Problem 4 (Bonus 20 pt). Extend the plain version Minimax Probability Machine into kernel version to let it support non-linear kernels like Gaussian kernel. And compare its performance with the same kernel SVM method. Report MPM's guaranteed error ϵ in line (1) on all possible distributions.

References

[1] Gert Lanckriet, Laurent Ghaoui, Chiranjib Bhattacharyya, and Michael Jordan. Minimax probability machine. *Advances in neural information processing systems*, 14, 2001.

[2] Gert RG Lanckriet, Laurent El Ghaoui, Chiranjib Bhattacharyya, and Michael I Jordan. A robust minimax approach to classification. *Journal of Machine Learning Research*, 3(Dec):555–582, 2002.