ELL888 Indian Institute of Technology Delhi
MCM Kernel Optimization
Scribed by: *Mridul Gupta (2021AIZ8322)*
Instructors: Sandeep Kumar and Jayadeva

Contents

1 Structure

Structure these points later

☒ Class separability might be worse in the feature space.

- Start with we want to get a bound on the risk on the test data. This requires knowing the distribution from which the data is being generated.
- We cannot know the distribution parameters. So, we estimate the actual risk using empirical risk which is the risk given some (training) samples from the distribution.
- Risk bound equation in VC slt.pdf
- Highlight two parts, the empirical risk and the structural risk
- make note that to reduce risk, we must reduce h (VC-Dimension)
- Mention VC-Dimension bound for Δ-margin classifiers.
- Mention that in general it will be $n+1$ but it can be bound by reducing $\frac{R^2}{\Delta^2}$

- Show the simplified graph
- This transformation might not take to same dimensional space, might be higher dimensional. This function is a Kernel, that needs to be optimized to optimize structural risk.
- Also mention examples why sometimes even an infinite dimensional kernel might not work. And why we need Kernel Optimization.
- Go into the details of how to do this.
- Mention appropriate theorems, definitions and lemmas in between.
- Add a key takeaways section.

Refined list to structure

- A little intro to MCM (slt, minimize vc dim bound, structural risk minimization).
- What is kernel optimization, why it is needed
- Amari's idea, magnifying around support vectors
- Xiong's idea of using a scatter matrix, Fisher discriminant
- Xiong's idea of creating a kernel scatter matrix, how it's equal to normal scatter
- Empirical feature space, matrix equations
- solving generalized eigenvalue problems
- Getting the MCM with optimized kernel

2 Introduction

Support Vector Machines (SVMs) are older than the state of Haryana (the linear variant at least), and the kernel version of SVMs were proposed by Vapnik *et al.* in 1992. Vapnik and Chernvonenkis along with others also developed theory key thoeretical concepts that constitute statistical learning theory. In what follows, we will discuss the theoretical and practical advancements since then that led to the work titled "Kernel optimization using conformal maps for the minimal complexity machine" by Jayadeva *et al.*

Since this is the congruence of two paths, one leading to MCMs and the other leading to kernel optimization in a data dependent way, in the following we first discuss MCMs, and then kernel optimization, and finally how the two fit together.

3 Background

The key concepts that lead to MCMs were available in statistical learning theory long ago, but weren't applied until 2015. So let's discuss the key concepts that naturally lead to minimal complexity machines.

3.1 Statistical Learning Theory

It all starts by stating the objective of Machine Learning. The model of learning from examples can be descibed using three components [**?**]:

1. a generator of random vectors x, drawn independently from a fixed but unknown distribution $P(x)$;

2. a supervisor that returns an output vector y for every input vector x, according to a conditional distribution function $P(y \mid x)$, also fixed but unknown;

3. a learning machine capable of implementing a set of functions $f(x, \alpha), \alpha \in \Lambda$.

The problem of learning then is to choose the "right" function from the family of functions $f(x, \alpha), \alpha \in \Lambda$. We want to choose this function so that it predicts the supervisor's response on seen as well as previously unseen data in the best possible way. The function choice has to be made based on a set of pairs $\{(x_i, y_i)\}_{i=1}^{M}$ called the training set. These samples are drawn from the distribution $P(x)P(y \mid x)$ which is the joint distribution $P(x, y)$. The M samples are assumed to be *independent and identically distributed (i.i.d.)*.

Next we need to define what "predicting the response in the best possible way means". For this we need to define how correct is $f(x, \alpha)$ is compared to y. This is done by defining an error function (also called loss function or cost function) that assigns some real valued number to two objects (may be vectors, matrices, scalars, or general mathematical objects). This number represents the "cost" incurred by predicting $f(x, \alpha)$ given y.

The next important quantity is the *risk functional* which is the expectation of loss. Since, x is a random vector, so the output of $f(\cdot, \alpha)$ is also random, so is the loss $\mathcal{L}$. Thus, it makes sense to take the expectation of it.

$$R(\alpha) = \mathbb{E}_{(x,y)\sim P(x,y)}\Big[\mathcal{L}\big(y, f(x, \alpha)\big)\Big] = \int \mathcal{L}\big(y, f(x, \alpha)\big) dP(x, y) \tag{1}$$

The goal is then to find α^* that minimizes the risk functional $R(\alpha)$. The problem is that $P(x, y)$ is unknown, all that's known is the training set $\{(x_i, y_i)\}_{i=1}^{M}$.

What we do have then is:

$$R_{\text{emp}}(\alpha) = \frac{1}{M} \sum_{i=1}^{M} \mathcal{L}\big(y, f(x, \alpha)\big) \tag{2}$$

R_{emp} is called the *empirical risk*. Given these quantities, we have the following result from statistical learning theory:

Theorem 3.1 *(Vapnik) If* $0 \leq \mathcal{L}\big(y, f(x, \alpha)\big) \leq B, \alpha \in \Lambda$, *that is the loss is totally bounded, then with probability at least* $1 - \eta$ *the inequality*

$$R(\alpha) \leq R_{emp}(\alpha) + \frac{B\varepsilon}{2}\left(1 + \sqrt{1 + \frac{4R_{emp(\alpha)}}{B\varepsilon}}\right) \tag{3}$$

holds true simultaneously for all functions of the set $\mathcal{L}\big(y, f(x,\alpha)\big)$. *where*

$$\varepsilon = 4\frac{h\left(\ln\frac{2M}{h}+1\right)-\ln\eta}{M} \tag{4}$$

and h *is the VC-dimension of the set of functions.*

On the RHS of equation 3, the term other than the empirical risk $R_{\text{emp}}(\alpha)$ is called the *structural risk*.

Another important result from Vapnik that will be needed to continue our discussion of MCMs is:

Theorem 3.2 *(Vapnik) Let vectors* $x \in X$ *belong to a sphere of radius* R. *Then the set of* Δ*-margin separating hyperplanes has the VC-dimension* h *bounded by the inequality*

$$h \leq \min\left(\left[\frac{R}{\Delta}\right]^2, n\right) + 1. \tag{5}$$

where n is the feature dimension of the data. For more, refer [**?**], but for our discussion this is sufficient.

3.2 Minimal Complexity Machines

3.2.1 Motivation

We want to minimize the actual **risk**, $R(\alpha)$, in order to generalize prediction on future samples. We can do this by tightening the bound on the RHS of equation 3. We can optimize the empirical risk, and we can optimize the structural risk. Note that in structural risk, the only thing we can alter is ε. We need to minimize ε to tighten the bound on the risk. And in the expansion of ε we can only alter h assuming the training sample size is as large as we can get. $\varepsilon \propto h$. Thus to reduce ε, we want to minimize h.

Now let's look at figure 1. Imagine we have the data points in $\mathbb{R}^1$. The blue dot shows the optimal hyperplane that classifies the points with maximum margin. R is the radius of the hypersphere and Δ is the margin. According to equation 5, we'd like to find the hyperplane such that R is minimized while Δ is maximized simultaneously.

3.2.2 MCM formulations

The mathematical formulations of MCM [**?**] starts by defining a mathematical quantity h_{MCM} as

$$h_{MCM} = \frac{\max_{i=1,2,\ldots,M}\|u^T x_i + v\|}{\min_{i=1,2,\ldots,M}\|u^T x_i + v\|} \tag{6}$$

where it is assumed that the separating hyperplane is $u^T x + v = 0$. It is assumed that the data is linearly separable.

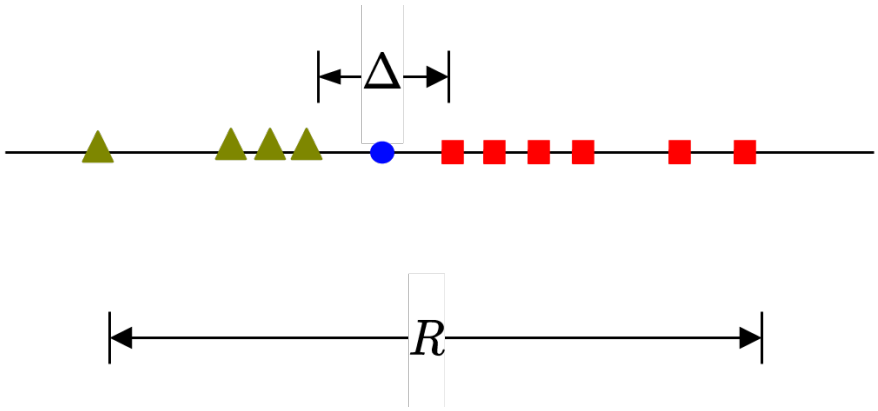

Figure 1: Data points $\in \mathbb{R}^1$

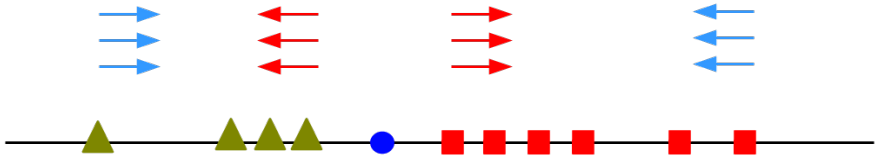

Figure 2: The red arrows are operations on margin and blue arrows on radius

Augmenting the data vectors to have a feature whose value is always 1, and concatenating the weight vector, we have $\hat{x}_i \leftarrow \{x_i; 1\}$ and $\hat{u} \leftarrow \{u; v\}$. Then the hyperplane goes through the origin in the $\mathbb{R}^{(n+1)}$ space.

Then, the margin Δ is given by

$$\Delta = \min_{i=1,2,\ldots,M} \frac{\|\hat{u}^T \hat{x}_i\|}{\|\hat{u}\|} \tag{7}$$

and the radius is just $R = \max_{i=1,2,\ldots,M} \|\hat{x}_i\|$. So, the ration of interest $\frac{R}{\Delta}$ is given by:

$$\frac{R}{\Delta} = \frac{\max_{i=1,2,\ldots,M} \|\hat{x}_i\|}{\min_{i=1,2,\ldots,M} \frac{\|\hat{u}^T \hat{x}_i\|}{\|\hat{u}\|}} = \frac{\max_{i=1,2,\ldots,M} \|\hat{u}\| \|\hat{x}_i\|}{\min_{i=1,2,\ldots,M} \|\hat{u}^T \hat{x}_i\|} \tag{8}$$

And using the Cauchy-Schwarz inequality ($\|a^T b\| \leq \|a\| \|b\|$)

$$\frac{R}{\Delta} \geq \frac{\max_{i=1,2,\ldots,M} \|\hat{u}^T \hat{x}_i\|}{\min_{i=1,2,\ldots,M} \|\hat{u}^T \hat{x}_i\|} = \frac{\max_{i=1,2,\ldots,M} \|u^T x_i + v\|}{\min_{i=1,2,\ldots,M} \|u^T x_i + v\|} \tag{9}$$

Thus

$$h_{MCM} \leq \frac{R}{\Delta} \tag{10}$$

$$\Rightarrow h^2_{MCM} \leq \left(\frac{R}{\Delta}\right)^2 < 1 + \left(\frac{R}{\Delta}\right)^2 \tag{11}$$

From equation **??** we have for large dimensional data:

$$h \leq 1 + \left(\frac{R}{\Delta}\right)^2 \tag{12}$$

Thus $\exists \beta \in \mathbb{R}^+$, such that $h \leq \beta h^2_{MCM}$. Also since, $h^2_{MCM} \geq 1$ and VC-dimension satisfies $h \geq 1$, $\exists \alpha \in \mathbb{R}, \alpha > 0$ such that $\alpha h^2_{MCM} \leq h$. Combining the two we have $\exists \alpha, \beta > 0, \alpha, \beta \in \mathbb{R}$ such that

$$\alpha h^2_{MCM} \leq h \leq \beta h^2_{MCM} \tag{13}$$

That is h^2_{MCM} is an exact bound on the VC dimension h. And since the data is linearly separable, $u^T x_i + v \geq 0$ if $y_i = 1$ and $u^T x_i + v \leq 0$ if $y_i = -1$. Thus $\|u^T x_i + v\|$ can be written as $y_i(u^T x_i + v)$. Thus the machine capacity can be minimized by keeping h^2_{MCM} as small as possible.

$$\underset{u,v}{\text{minimize}}\ h_{MCM} = \frac{\max_{i=1,\ldots,M} y_i(u^T x_i + v)}{\min_{i=1,\ldots,M} y_i(u^T x_i + v)} \tag{14}$$

Further the authors show that the above formulation can be further simplified by writing:

$$h_{MCM} = \frac{g}{l} \tag{15}$$

$$\min_{u,v,g,l} \frac{g}{l} \tag{16}$$

$$g \geq y_i(u^T x_i + v), \quad i = 1, \ldots, M \tag{17}$$

$$l \leq y_i(u^T x_i + v), \quad i = 1, \ldots, M \tag{18}$$

Using Charnes-Cooper transformation, introducing $p = \frac{1}{l}$

$$\min_{u,v,g,l,p} g \cdot p \tag{19}$$

$$g \cdot p \geq y_i(p \cdot u^T x_i + p \cdot v), \quad i = 1, \ldots, M \tag{20}$$

$$l \cdot p \leq y_i(p \cdot u^T x_i + p \cdot v), \quad i = 1, \ldots, M \tag{21}$$

$$p \cdot l = 1 \tag{22}$$

Denoting $w \triangleq p \cdot u, b \triangleq p \cdot v$ and noting that $p \cdot l = 1$

$$\min_{w,b,h} h \tag{23}$$

$$h \geq y_i(w^T x_i + b), \quad i = 1, \ldots, M \tag{24}$$

$$1 \leq y_i(w^T x_i + b), \quad i = 1, \ldots, M \tag{25}$$

Equations 23-25 define the Minimal Complexity Machine (MCM). And it is trained by solving the Linear Programming Problem defined above.

3.2.3 Generalizing the MCM

The MCM above is further generalized to allow for classification errors by introducing slack variables.

$$\begin{aligned}
&\min_{w,b,h,q} \; h + C \cdot \sum_{i=1}^{M} q_i \\
&h \geq y_i(w^T x_i + b) + q_i, \quad i = 1, \ldots, M \\
&1 \leq y_i(w^T x_i + b) + q_i, \quad i = 1, \ldots, M \\
&q_i \geq 0 \quad i = 1, \ldots, M
\end{aligned}$$

And for the non-linear case using Kernels

$$\min_{w,b,h,q} \; h + C \cdot \sum_{i=1}^{M} q_i \tag{26}$$

$$h \geq y_i(w^T \phi(x)_i + b) + q_i, \quad i = 1, \ldots, M \tag{27}$$

$$1 \leq y_i(w^T \phi(x)_i + b) + q_i, \quad i = 1, \ldots, M \tag{28}$$

$$q_i \geq 0 \quad i = 1, \ldots, M \tag{29}$$

References

[1] Cormen TH, Leiserson CE, Rivest RL, Stein C. *Introduction to algorithms.*. MIT press. 2009 Jul 31