Questions and my answers

Chanwoo Lee, June 3, 2020

algorithmic accuracy vs. statistical accuracy.

1. **Why do you choose SVM instead of other non-parametric estimation?:**
ans) SVM works the best when $\dim(\boldsymbol{X}) \gg \#$ of data points case. Especially in matrix feature (or higher dimensional) setting, feature dimension usually is greater than the number of data available. SVM works very well for this setting. One of the reason for this is that algorithm complexity of dual problem is fixed by the number of data points. Another benefit using SVM is that it is robust. Our obtained function from SVM has the form as

vs. K-NN, vs. neural network

$$\hat{f}(\boldsymbol{x}) = \sum_{i=1}^{N} \hat{\alpha}_i y_i \langle \boldsymbol{x}, \boldsymbol{x}_i \rangle,$$

where $\boldsymbol{\alpha} \in \mathbb{R}^N$ is solution of dual problem. We can check this solution of dual problem is sparse having many 0 entries. This means that points around boundary only have impact on the classifier but others not, which gives SVM robustness.

independence among (X,Y) \sim unkonwn distribution (\tX, \txY); bernoulli distribution on YIX ~ Bernoulli (f(X))?

2. **Why is our method called non-parametric?**
ans) First of all, we do not assume any distribution on the model i.e. it is distribution free model which is non-parametric. Secondly, if we check our classifier it has the form (including SMM) as

$$\hat{f}(\boldsymbol{x}) = \sum_{i=1}^{N} \hat{\alpha}_i y_i \langle \boldsymbol{x}, \boldsymbol{x}_i \rangle.$$

agree with this.

Therefore, as $N \to \infty$, we can think that our method has infinity parameter to estimate.

3. **What is the practical benefit of using low rank structure in SMM?**
ans) If we are using the full rank, we have too many parameters which can result in overfitting problem. Low rank structure prevent us from overfitting the data. Secondly, low rank consider matrix structure of feature matrix $\boldsymbol{X}$. Our classifier with this low rank has the form as

$$\hat{f}(\boldsymbol{X}) = \sum_{i=1}^{N} \hat{\alpha}_i y_i \langle \hat{\boldsymbol{P}}_r \boldsymbol{X}, \hat{\boldsymbol{P}}_r \boldsymbol{X}_i \rangle = \sum_{i=1}^{N} \hat{\alpha}_i y_i \langle \boldsymbol{X} \hat{\boldsymbol{P}}_c, \boldsymbol{X}_i \hat{\boldsymbol{P}}_c \rangle = \sum_{i=1}^{N} \hat{\alpha}_i y_i \langle \hat{\boldsymbol{P}}_r \boldsymbol{X} \hat{\boldsymbol{P}}_c, \hat{\boldsymbol{P}}_r \boldsymbol{X}_i \hat{\boldsymbol{P}}_c \rangle. \quad (1)$$

Joint dimension reduction + supervised learning. can be viewed as a supervised PCA.

(1) shows that low rank structure consider column space and row space of the feature matrix.

4. **What is intuition of (1)?, does it have something fundamental?**
ans) I checked that this really happens from simulation i.e.

<B, X> = <QR, X>, then solve for Q and R

<B, X> = <PUR, X>, then solve for P, U, R (matrix SVD parameterization)

alpha are all the same.

$$\hat{\boldsymbol{B}} = \sum_{i=1}^{N} \hat{\alpha}_i y_i \hat{\boldsymbol{P}}_r \boldsymbol{X}_i = \sum_{i=1}^{N} \hat{\alpha}_i y_i \hat{\boldsymbol{X}}_i \hat{\boldsymbol{P}}_c = \sum_{i=1}^{N} \hat{\alpha}_i y_i \hat{\boldsymbol{P}}_r \boldsymbol{X}_i \hat{\boldsymbol{P}}_c. \quad (2)$$

Since this formula is derived from the first order condition in dual problem and linearity plays crucial role for this formula, (2) holds when the function is linear to feature matrix. Intuitively, we are looking for a linear coefficient $\boldsymbol{B}$ that extract the information from $\boldsymbol{X}_i$ the best. Since $\{\text{row}(\boldsymbol{X}_i)_j\}_{j=1}^{d_1}$ and $\{\text{column}(\boldsymbol{X}_i)_j\}_{j=1}^{d_2}$ contain the same information, the coefficient $\boldsymbol{B}$ should be the same. But I am not clear what kinds of fundamental theorems are there mathematically.

reason: identifiability due to linearity in both d.r. and s.l

dimension reduction + supervised learning, 4 possible combinations:
linear d.r. + linear supervised learning
nonlinear d.r+ linear s.l.
linear d.r + nonlinear s.l
nonlinear d.r + nonlinear s.l

<PXP, B> = <XP, PB> = <X, PBP>
<P h(X) P, B>
<g(PXP), B>
g(<Ph(X)P, B>)

complexity of SVM: number of sample size required for classification.

5. **$\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}') = h(\boldsymbol{X})h^T(\boldsymbol{X}')$ is not really kernel. How do we call this?**
ans) asymmetric kernel? lifted matrix?

6. **Intuitive way to explain why SVM based probability estimation is better than histogram method.** convergence rate: n^(-{2 alpha/ 2alpha+d}): d is dimension of the feature, and n is the number of sample points, alpha = smooth parameter, e.g. for Holder function or lipschitz function
ans) The histogram probability estimation method is not working when the dimension of feature is large. Notice that histogram is making grid on feature map space for estimation. Histogram method cannot estimate probability well in the area of which no data point is observed especially in high dimensional setting. SVM based probability estimation have advantage when dimension is large but have small data size. reasonable.

7. **how $p(\boldsymbol{X}) = \mathbb{P}(Y = 1|\boldsymbol{X})$ affects the accuracy in probability estimation?**
The condition $p(\boldsymbol{X})$ is monotonic function of $\langle \boldsymbol{X}, \cdot \rangle$ is enough for consistent estimation. Consider the following probability estimation process

$$\frac{1}{H}\sum_{i=1}^{N} \mathbb{1}\left\{\boldsymbol{X} : \operatorname{sign}(\hat{f}_h(\boldsymbol{X})) = -1\right\} \to \frac{1}{H}\sum_{i=1}^{N} \mathbb{1}\left\{\boldsymbol{X} : p(\boldsymbol{X}) \le \frac{h}{H}\right\} \approx p(\boldsymbol{X})$$

Lebesgue/Riemann

In the last approximation we can get a insight how geometric figure of $p(\boldsymbol{X})$ impact on our estimation. If $p(\boldsymbol{X})$ is smooth function that the last approximation works well has accurate estimator. On the other hand $p(\boldsymbol{X})$ is a wiggly function, the last approximation perform poorly resulting in bad estimation. As long as $p(\boldsymbol{X})$ is Lebesgue measurable function, the last approximation is consistent as $H \to \infty$. But in the finite approximation case, the geometric features of $p(\boldsymbol{X})$ do matter in estimation accuracy.

8. **Why whitening step is needed for SDR but not for prob.est?**
ans) We obtain prob.est from solving
Claim: does the prob. estimation theory go through with either formulation (3) or formulation (4)?

$$\operatorname{Vec}(\boldsymbol{B})^T \operatorname{Vec}(\boldsymbol{B}) + \frac{\lambda}{n}\sum_{i=1}^{n} \omega_\pi(y_i) \left|1 - Y_i \hat{f}_n(\boldsymbol{X}_i; \boldsymbol{B})\right|_+, \tag{3}$$

p(Y=1|X)

where $\hat{f}_n(\boldsymbol{X}_i; \boldsymbol{B}) = \langle \boldsymbol{B}, \boldsymbol{X}_i \rangle$.
However, we obtain SDR estimator from solving
we have to use (4). (3) is invalid for SDR

$$\operatorname{Vec}(\boldsymbol{B})\hat{\Sigma}_n \operatorname{Vec}(\boldsymbol{B}) + \frac{\lambda}{n}\sum_{i=1}^{n} \omega_\pi(y_i) \left|1 - Y_i \hat{f}_n(\boldsymbol{X}_i; \boldsymbol{B})\right|_+, \tag{4}$$

where $\hat{f}_n(\boldsymbol{X}_i; \boldsymbol{B}) = \langle \boldsymbol{B}, \boldsymbol{X}_i - \bar{\boldsymbol{X}}_n \rangle$, $\bar{\boldsymbol{X}}_n = n^{-1}\sum_{i=1}^{n} \boldsymbol{X}_i$, and $\hat{\Sigma}_n$ denotes the sample covariance matrix of $\{\operatorname{Vec}(\boldsymbol{X}_i)\}_{i=1}^{n}$. Therefore, by whitening step, we simplify (4) to (3) where we can use SMM algorithm.

Probability estimation: (X, y). 1 whiten X <— X \Sigma^{-1/2}
2. weighted SVM/SMM —> function estimation. p(Y=1|X\Sigma^{-1/2})