SMM conditional probability

Chanwoo Lee, April 20, 2020

1 Simulation of SMM Conditional Probability

I generate feature data matrix with the following rule.

$$\begin{aligned}
&\mathbb{P}(X_{\cdot 1}|y=1) \quad \sim N((1,1)^T, I_2), \quad \mathbb{P}(X_{\cdot 2}|y=1) \quad \sim N((-1,-1)^T, I_2),\\
&\mathbb{P}(X_{\cdot 1}|y=-1) \sim N((0,0)^T, I_2), \quad \mathbb{P}(X_{\cdot 2}|y=-1) \sim N((0,0)^T, I_2),\\
&X = (X_{\cdot 1}, X_{\cdot 2}) \in \mathbb{R}^{2\times 2}.
\end{aligned}$$

With this rule, we have a test data set $(X^1,1),\ldots,(X^{20},1),(X^{21},-1),\ldots,(X^{40},-1)$. Define

$$Z_1 = X_{11} + X_{21} \text{ and } Z_2 = X_{12} + X_{22}.$$

We can plot the those feature matrices into two dimension transforming the matrices as $\left((Z_1^1, Z_2^1)^T, 1\right), \ldots, \left((Z_1^{20}, Z_2^{20})^T, 1\right), \left((Z_1^{21}, Z_2^{21})^T, -1\right), \ldots, \left((Z_1^{40}, Z_2^{40})^T, -1\right)$. Figure 1 shows the data distribution.

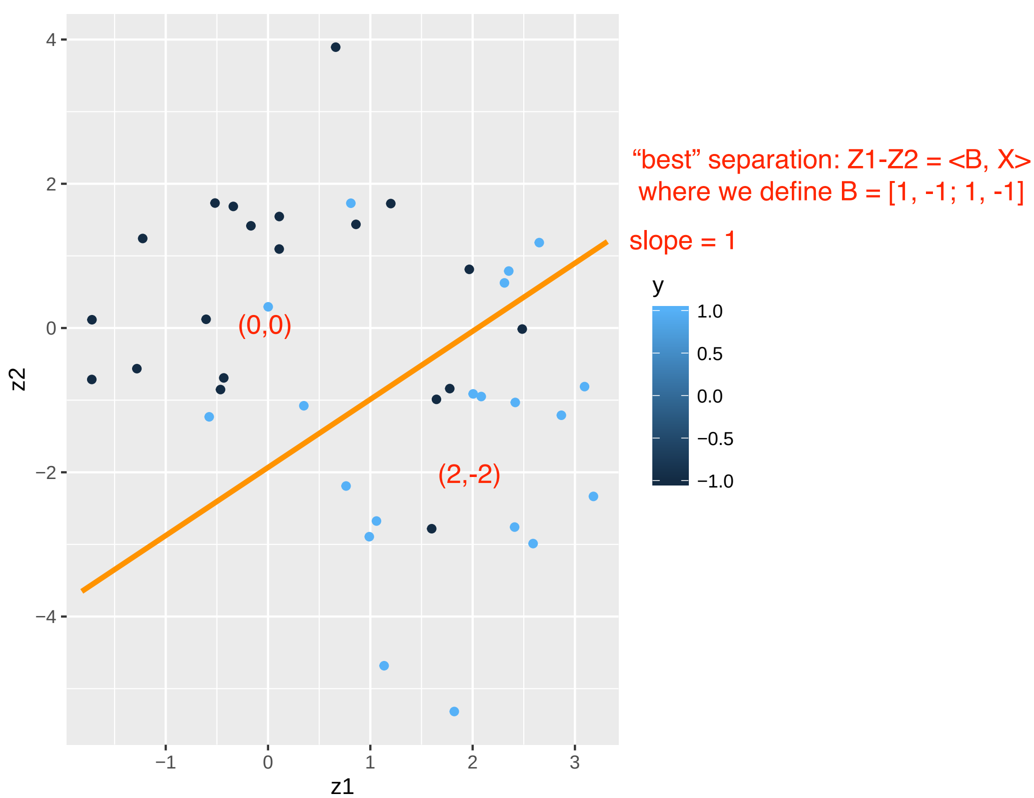

Figure 1: Visualization of the feature matrices. $Z1$ is the sum of the first column and $Z1$ is of the second one.

Notice that we can calculate the distribution of $(Z_1, Z_2)^T$ as,

$$\mathbb{P}((Z_1, Z_2)^T|y=1) \quad \sim N((2,-2), 2I_2), \tag{1}$$

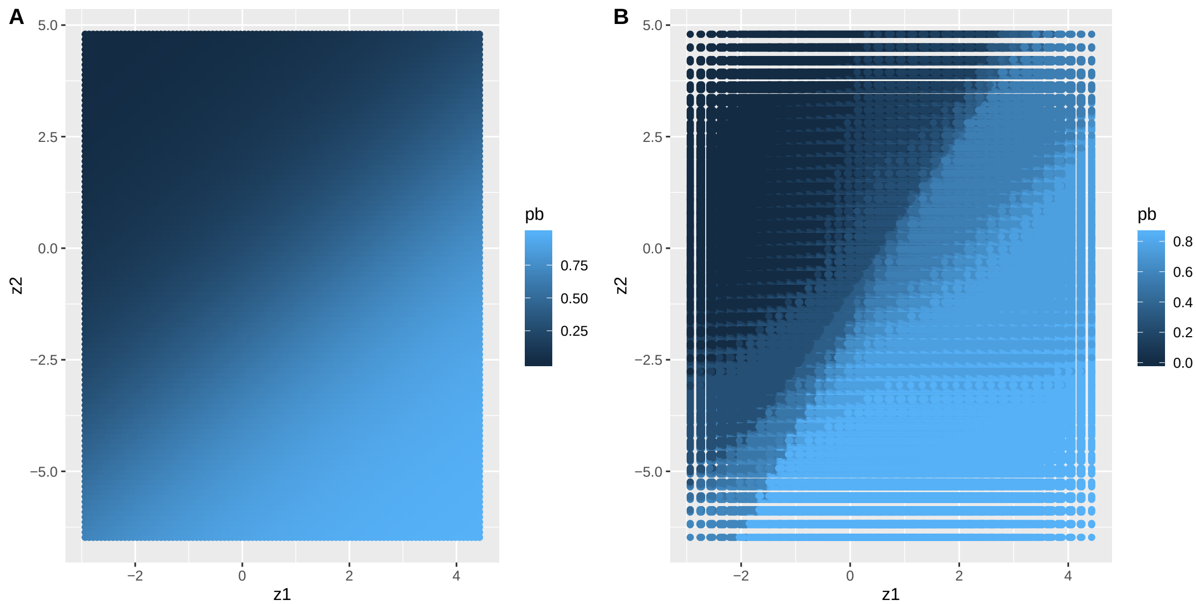

Figure 2: Figure A shows the true conditional probability of y given (Z_1, Z_2). Figure B shows the estimated probability using SMM weighted hinge loss approach.

$$\mathbb{P}((Z_1, Z_2)^T | y = -1) \sim N((0,0), 2I_2).$$

From the Bayes rule, we calculate the conditional probability $\mathbb{P}(y = 1|(Z_1, Z_2))$ with the assumption $\mathbb{P}(y = 1) = \mathbb{P}(y = -1)$ as

$$\begin{aligned}\mathbb{P}(y = 1|(Z_1, Z_2)^T) &= \frac{\mathbb{P}((Z_1, Z_2)^T | y = 1)\mathbb{P}(y = 1)}{\mathbb{P}((Z_1, Z_2)^T | y = 1)\mathbb{P}(y = 1) + \mathbb{P}((Z_1, Z_2)^T | y = -1)\mathbb{P}(y = -1)} \\ &= \frac{1}{1 + \exp\left(-(2, -2)\left((Z_1, Z_2) - (1, -1)\right)^T / 4\right)}.\end{aligned} \tag{2}$$

Sufficient statistics: Z1-Z2=X11+X21-X12-X22

In the last equality in (2), we use the distribution in (1).

As a by-product, we learn the dimension reduction from original X-space (dim = 4) to a reduced space (dim = 1)

Using the weighted hinge loss approach, I calculate the conditional probability using SMM method (with modified algorithm for conditional probability). I set the rank as one when fitting SMM model to the data set. Figure 2 shows the true conditional probability $\mathbb{P}(y = 1|(Z_1, Z_2))$ and the estimated one. We can see that SMM conditional probability estimation works well since the true distribution is linear.

SMM achieves two goals simultaneously:
1. estimate the probability
2. find sufficient dimension reduction in X-space.

2 Kernel method

I am suggesting new kernel method which makes optimization easier. Define feature mapping $h : \mathbb{R}^{m\times n} \to \mathbb{R}^{m'\times n}$ where $m < m'$. Kernels for matrix case can be define as

$$K(X, X') = h(X)^T h(X') \in \mathbb{R}^{n\times n}.$$ (column-wise kernel)

Our objective primal problem is

$$\min_{U\in\mathbb{R}^{m'\times r}, V\in\mathbb{R}^{n\times r}, \boldsymbol{\xi}} \frac{1}{2}\|UV^T\|^2 + c\sum_{i=1}^{N} \xi_i \tag{3}$$

$$\text{subject to } y_i(\langle UV^T, h(X_i)\rangle + b) \le 1 - \xi_i$$
$$\xi_i \ge 0, \quad i = 1, \ldots, N.$$

How to update U while fixing V?

Then fixing V we have the dual problem of (3).

(you may want to extend h: m-by-n to m'-by-n', where m'>=m, n'>=n
h induces both row-wise kernel and column-wise kernel)

$$\begin{aligned}
&\min_{\boldsymbol{\alpha}} -\sum_{i=1}^{N} \alpha_i + \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i\alpha_j y_i y_j \langle h(X_i)H_V, h(X_j)H_V\rangle \\
&\text{subject to } \sum_{i=1}^{N} y_i\alpha_i = 0 \\
&\qquad\qquad 0 \le \alpha_i \le C, \quad i = 1, \ldots, N.
\end{aligned}$$

Notice $\langle h(X_i)H_V, h(X_j)H_V\rangle = \mathrm{tr}\left(H_V h(X_i)^T h(X_j)\right) = \mathrm{tr}\left(H_V K(X_i, X_j)\right)$. Therefore, we can update U and V without the information of the feature map h. We can define the following kernels.

$$\begin{aligned}
\text{Linear:} \quad & K(X, X') = X^T X' \\
\text{Polynomial:} \quad & K(X, X') = (X^T X' + I_n)^d \\
\text{Radial:} \quad & K(X, X') = \exp\left((X - X')^T (X - X')/\sigma\right),
\end{aligned}$$

where $\exp(A) = \sum_{n=0}^{\infty} \frac{1}{n!} A^n$. Notice that when $X \in \mathbb{R}^{m\times 1}$ i.e. X is a vector, all those definitions are reduced to SVM case. From this way, we can generalize linear SMM method to Kernel SMM with tractable algorithm.