Active Sampling for Similarity Learning with its Applications in Carpet Dyelot Merge

1. Introduction:

This research is motivated from the real-world applications in carpet manufacturing where extensive sensor data are available to measure various characteristics of carpets. During the manufacturing process, carpets are produced in different dyelots at different times depending on the customer's orders with respect to the style, color, and raw materials of carpets. In each dyelot, different rolls might be produced, often during the same time. It is expected that different rolls with the same dyelot can be merger together to sell together to the customers. An important business opportunity is when two dye lots of carpets are indistinguishable, which will allow the company to reduce cost by combining carpets from different dyelots to fill the customer's orders.

One way to evaluate whether we could match two rolls of carpets produced in different dye lots or not is to ask for expert's assessments. Expert-inspections, however, is labor-intensive and very costly. Moreover, there are often thousands of dyelots in a month's production, and it is unclear for human being to go through all possible pairwise comparisons. This leads our research problem how to combine the expert's opinion with the sensor data to assess the similarity of different dyelots of carpets.

To be more specific, we propose to actively select only a few rolls of carpets which probably match with each other, for expert inspections. Next, based on the expert's feedback, we will develop a similarity learning method based on the sensing data, which yields a carpet dyelot merger policy that should be much cheaper than the human-inspection method.

There are two challenging tasks. The first one is the active sampling problem regarding which rolls should be selected for expert inspections. We propose a greedy algorithm to find only a few rolls with similar features to be checked by experts, as most dyelots might not be merged. The second is the similarity learning algorithm, and we propose to apply Gaussian process classifier to evaluate whether we could merge two carpets or not.

The technical details are as follows.

2. Model and problem formulation

In the manufacturing recording system, the p-dimensional features $X_i = \left(X_{i1}, X_{i2}, \dots, X_{ip}\right)^T$, $i = 1, \dots, n$, represents the sensor readings of each roll. We first consider a decision rule on whether we could merge two carpets or not. For two carpets with sensor readings X_i and X_j, we assume that they can be merged if and only if

$$D\left(X_i, X_j\right) \leq C,$$

for some unknown constant C, which represents the detection threshold. In general, we do not know how to define the distance, since the components of X_i might be correlated. Furthermore, responses of matching are correlated since if we can math two pairs of carpet, then the three carpets are probably match with each other. In other words, this is the problem of classification with correlated response. since we do not have any information about the distance function D, we let D as a weighted Euclidean distance:

$$D(X_1, X_2; w) = \sum_{i=1}^{p} w_i (X_{i1} - X_{i2})^2 = w(X_1 - X_2)^2$$

where the weights $w = (w_1, \dots, w_p)$ are unknown. For estimating w, we transform the original data into a new representation, which is called distance data, denoted $X^d = (X^d_{(1,2)}, \dots, X^d_{(n,n-1)})$. In the distance data, each column is standardized, and each row of data indicates the square-difference of each feature. That is

$$X^d_{(i,j)} = \left(X_i - X_j\right)^2$$

Moreover, according to the model assumption, the validation of merging is determined by whether the weighted Euclidean distance is enough short or not. The binary outcome of whether we could merge ith and jth rolls is $Y^d = (Y^d_{(1,2)}, \dots, Y^d_{(n,n-1)})$, and components of Y^d is

$$Y^d_{(i,j)} = \mathrm{Sign}(\, \mathrm{C} - D(X_1, X_2)\,) = \begin{cases} 1 & \text{if the ith and the jth rolls can be merged} \\ -1 & \text{if the ith and the jth rolls cannot be merged} \end{cases}$$

Thus, we are able to estimate w via supervised learning techniques if we do have distance data X^d and responses Y^d. Unfortunately, since the expert has not evaluated Y^ds yet, we cannot do supervise learning now. We have to select a few rolls to be investigated by experts and then obtain responses. Since human-inspection is expensive, we cannot send too many rolls to experts. Therefore, we would like to select a few representative samples to help us estimate w. Suppose we try to select m samples to be investigated if $\binom{m}{2}$ times comparison is available, our target is to find a subset with m elements, denoted S_m, in the distance data such that we could merge all the rolls in the subset. In other words, we try to find a subset S_m, which has the weighted Euclidean distance of all of the elements in S_m are less than C. If we find such S_m, we cannot do supervised learning since all responses are the same. However, the carpet company would prefer this result, because it helps them to select the subset of carpets that all elements are matched. On the other hand, if S_m is not the all-matching subset, then we are able to do supervised learning to estimate w.

According to the assumption of merging criterion, the classes of 1 and -1 are separated by a hyperplane $\{\mathrm{C} = wx^d\}$. To estimate w, (SVM) is a method to find a hyperplane, which has the largest margin that separates two classes. Moreover, the estimation of w in SVM relates only to support vectors which are the data nearest the hyperplane. That is, if S_m contains all support vectors in the full data, then the estimation result of using full data will be as same as if we merely use S_m to estimate the parameter of the hyperplane.

3. Methodology

We develop an algorithm to select a subset with m elements, denoted S_m, which has the minimal full-connected distance. That is

$$S_m = \arg\min_{x_1,\ldots,x_m \in X} \textstyle\sum_{x_i \neq x_j} D(x_1, x_2; w = 1),$$

Suppose S_m^* is the set of dropping one element, denoted s^*, out of S_m. A characteristic of S_m is

$$s^* = \arg\min_{x \notin S_m^*} \textstyle\sum_{s \in S_m^*} D(x, s),$$

that is the element has the smallest distance between all elements in S_m^*: s^* has the smallest distance of joining. According to the characteristic of smallest distance of joining, we are able to develop a greedy algorithm to approach S_m. We start at a specific element x and set $S_{x,1} = x$, then recruit the element, which has the smallest distance of joining $S_{x,2}$. Until the set has m elements, denoted $S_{x,m}$, the set is thus a candidate set with m elements, which has minimal full-connected distance. Therefore, S_m is the subset, which has minimal full-connected distance through all candidate subsets with different initial elements in X.

Algorithm:

1. $S_{x,1} = x$
2. for j=1,…,m-1,

$$S_{x,j+1} = S_{x,j} \cup \arg\min_{x \notin S_{x,j}} \sum_{s \in S_{x,j}} D(x, s)$$

3. $S_m = \min_{x \in X} \sum_{s_1 s_2 \in S_{x,m}} D(s_1, s_2)$

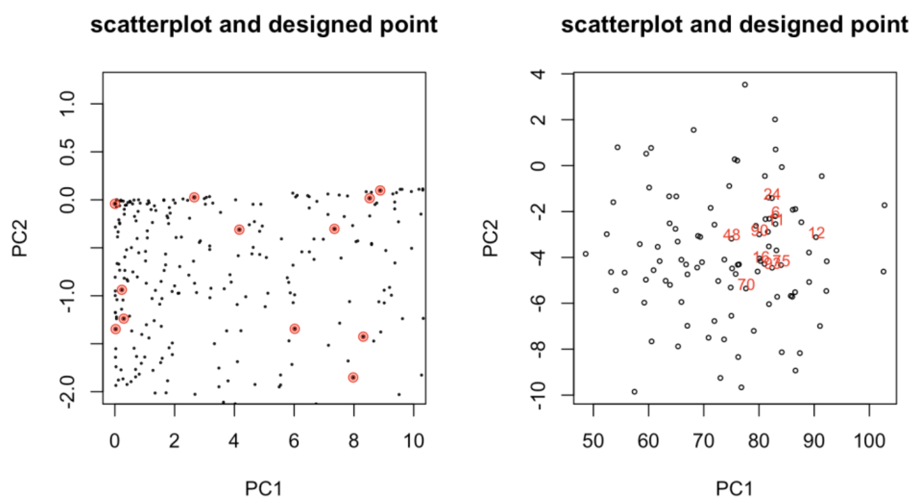

Figure 1: Right penal is the scatter plot that represents S_m by red dots in the original data. Left penal is the scatter plot that represents the pairwise combination of S_m in the distance data. We transfer both figures into the first and the second principal direction for visualization.

Estimating $\boldsymbol{w}$

Support vector machine is a method to deal with classification problems. Since we have investigated the responses of all the pairwise combinations of S_m, we are able to use $X^d_{(i,j)}$ and $Y^d_{(i,j)}$, $i, j \in S_m$ as the training data of SVM. Notice that the SVM is applied to distance data X^d but not the original data X. To be more specific, if we design m elements to be investigated, we will have $\binom{m}{2}$ responses of Y^d.

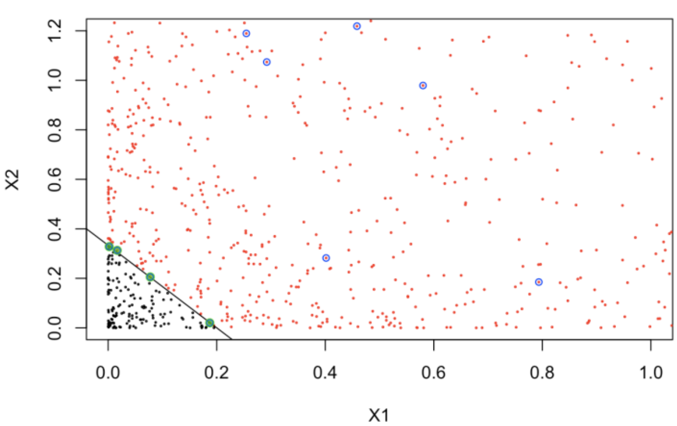

Figure 2: Red dots represent the -1 class in the distance data, which is same as the unmatched pairs in the original data. By contrast, black dots represent the 1 class in the distance data. While blue dots represent the chosen elements, green dots are support vectors which influence the result of estimation for w. The black line is the separating hyperplane.

4. Numerical study

For a given data set $X_i, \ldots, X_{100}$, set the decision rule as $f(X_i, X_j) = \text{sign}\{2 - (X_i - X_j)^2(.3,\ .2,\ .5)^T\}$.

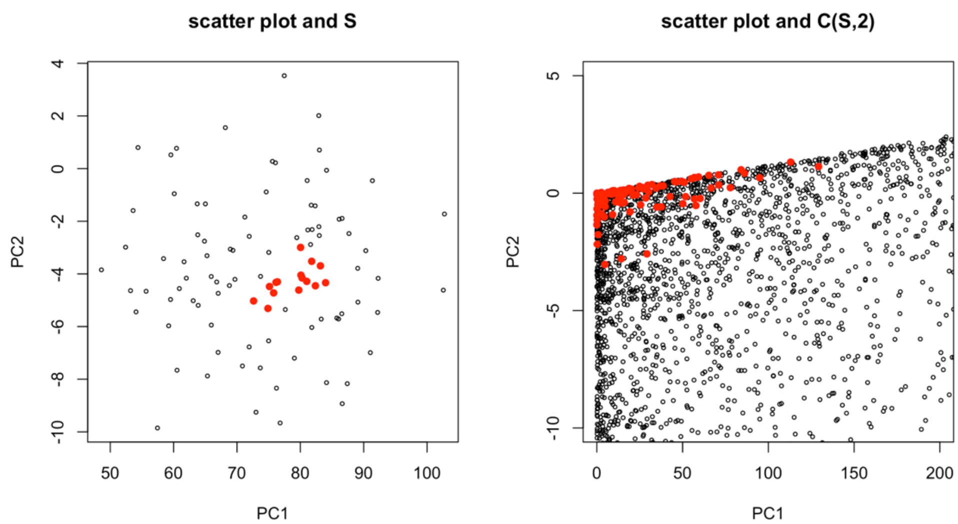

Figure 3: Let m=15, left panel shows the scatter plot of the original data, and red dots represents the selected 15-subset. Right panel shows the scatter plot of the distance data, and red dots represents the pairwise combinations in distance data.

Training data

True / Pred.	-1	1
-1	78	3
1	2	22

Testing data

True / Pred.	-1	1
-1	4863	20
1	5	62

Figure 4: Left panel shows the classification result of the training data. Right panel show the classification result of the testing data. Both panel shows the classification results are good.

4. Conclusion

We develop a greedy algorithm to select a subset S_m with m elements, which has the minimal full-connected distance. Such subset has a property that either all pairwise combination are matched or allows us to estimate the merging criterion by using SVM. The result only relates to support vectors, which allows us only to investigate a few elements in the original data.