B565 DATA MINING

23 April 2018

Homework 5

Chuck Jia

Problem 1

(a) Let (X_1, X_2) and Y denote the data point variable and the target label variable. Note that

$$P(Y=-) = P(Y=+) = \frac{1}{2}$$
$$P(X_1=0|Y=-) = P(X_2=0|Y=-) = P(X_1=1|Y=-) = P(X_2=1|Y=-) = \frac{1}{2}$$
$$P(X_1=0|Y=+) = P(X_2=0|Y=+) = P(X_1=1|Y=+) = P(X_2=1|Y=+) = \frac{1}{2}$$

That is, for any data point $(x_1, x_2) \in \mathcal{D}$ and $y \in \{-,+\}$,

$$P(Y=y) = \frac{1}{2}, \quad P\big((X_1,X_2)=(x_1,x_2)|Y=y\big) = \frac{1}{2}$$

To predict any data point (x_1, x_2),

$$P(-)P(X_1=x_1|Y=-)P(X_2=x_2|Y=-) = \frac{1}{2}\cdot\frac{1}{2}\cdot\frac{1}{2} = \frac{1}{8}$$

and

$$P(+)P(X_1=x_1|Y=+)P(X_2=x_2|Y=+) = \frac{1}{2}\cdot\frac{1}{2}\cdot\frac{1}{2} = \frac{1}{8}$$

That is, for any data point in the data set $\mathcal{D}$, the Naive Bayes Classifier would result in a tie when deciding on the class label. If we use random guessing as tie breaker, then the error rate would be

$$\text{error rate} = 0.5$$

Therefore, the Naive Bayes Classifier cannot correctly predict the class labels for $\mathcal{D}$. Or more precisely, the Naive Bayes Classifier does not perform better than random guessing on $\mathcal{D}$.

(b) After transformation, the data set is

$$\begin{aligned}\mathcal{D}' = \{&(1,0,0,0,0,0),\\ &(1,0,1,0,0,1),\\ &(1,1,0,0,1,0),\\ &(1,1,1,1,1,1)\}\end{aligned}$$

To predict $(1,0,0,0,0,0)$,

$$\begin{aligned}&P(Y=-)P(X_1=1|-)P(X_2=0|-)P(X_3=0|-)P(X_4=0|-)P(X_5=0|-)P(X_6=0|-)\\ =&\frac{1}{2}\cdot 1\cdot\frac{1}{2}\cdot\frac{1}{2}\cdot\frac{1}{2}\cdot\frac{1}{2}\cdot\frac{1}{2} = \frac{1}{64}\end{aligned}$$

$$\begin{aligned}&P(Y=+)P(X_1=1|+)P(X_2=0|+)P(X_3=0|+)P(X_4=0|+)P(X_5=0|+)P(X_6=0|+)\\=&\frac{1}{2}\cdot 1\cdot\frac{1}{2}\cdot\frac{1}{2}\cdot 1\cdot\frac{1}{2}\cdot\frac{1}{2}=\frac{1}{32}\end{aligned}$$

Therefore, the predicted label for $(1,0,0,0,0,0)$, i.e. $(0,0)$, is $Y=+$.

To predict $(1,0,1,0,0,1)$,

$$\begin{aligned}&P(Y=-)P(X_1=1|-)P(X_2=0|-)P(X_3=1|-)P(X_4=0|-)P(X_5=0|-)P(X_6=1|-)\\=&\frac{1}{2}\cdot 1\cdot\frac{1}{2}\cdot\frac{1}{2}\cdot\frac{1}{2}\cdot\frac{1}{2}\cdot\frac{1}{2}=\frac{1}{64}\end{aligned}$$

$$\begin{aligned}&P(Y=+)P(X_1=1|+)P(X_2=0|+)P(X_3=1|+)P(X_4=0|+)P(X_5=0|+)P(X_6=1|+)\\=&\frac{1}{2}\cdot 1\cdot\frac{1}{2}\cdot\frac{1}{2}\cdot 1\cdot\frac{1}{2}\cdot\frac{1}{2}=\frac{1}{32}\end{aligned}$$

Therefore, the predicted label for $(1,0,1,0,0,1)$, i.e. $(0,1)$, is $Y=+$.

To predict $(1,1,0,0,1,0)$,

$$\begin{aligned}&P(Y=-)P(X_1=1|-)P(X_2=1|-)P(X_3=0|-)P(X_4=0|-)P(X_5=1|-)P(X_6=0|-)\\=&\frac{1}{2}\cdot 1\cdot\frac{1}{2}\cdot\frac{1}{2}\cdot\frac{1}{2}\cdot\frac{1}{2}\cdot\frac{1}{2}=\frac{1}{64}\end{aligned}$$

$$\begin{aligned}&P(Y=+)P(X_1=1|+)P(X_2=1|+)P(X_3=0|+)P(X_4=0|+)P(X_5=1|+)P(X_6=0|+)\\=&\frac{1}{2}\cdot 1\cdot\frac{1}{2}\cdot\frac{1}{2}\cdot 1\cdot\frac{1}{2}\cdot\frac{1}{2}=\frac{1}{32}\end{aligned}$$

Therefore, the predicted label for $(1,1,0,0,1,0)$, i.e. $(1,0)$, is $Y=+$.

To predict $(1,1,1,1,1,1)$,

$$\begin{aligned}&P(Y=-)P(X_1=1|-)P(X_2=1|-)P(X_3=1|-)P(X_4=1|-)P(X_5=1|-)P(X_6=1|-)\\=&\frac{1}{2}\cdot 1\cdot\frac{1}{2}\cdot\frac{1}{2}\cdot\frac{1}{2}\cdot\frac{1}{2}\cdot\frac{1}{2}=\frac{1}{64}\end{aligned}$$

$$\begin{aligned}&P(Y=+)P(X_1=1|+)P(X_2=1|+)P(X_3=1|+)P(X_4=1|+)P(X_5=1|+)P(X_6=1|+)\\=&\frac{1}{2}\cdot 1\cdot\frac{1}{2}\cdot\frac{1}{2}\cdot 0\cdot\frac{1}{2}\cdot\frac{1}{2}=0\end{aligned}$$

Therefore, the predicted label for $(1,1,1,1,1,1)$, i.e. $(1,1)$, is $Y=-$.

The classifier correctly predicted class labels for all data points except the first one in the data set $\mathcal{D}$. The error rate for the result is

$$\text{error rate}=\frac{1}{4}$$

Problem 2

Assume we have n items in total. Also, we assume here that $n > k$.

We give an algorithm that calculates the number of possible k-sequences. Algorithm 2.1 below defines a recursive function `calcNumSequences` that calculates the number of potential sequences of length k.

Algorithm 2.1 Function `calcNumSequences`(k)

```
1: if k == 0
2:     ans = 1
3: else
4:     ans = 0
5:     for i = 1 to k
6:         ans += C(n, i) · calcNumSequences(k − i)
7:     end for
8: end if
9: return ans
```

The idea is as follows[1].

- First we determine the first itemset in the sequence. Note that every sequence has a first itemset, which guarantees that we do not miss any sequences, and sequences with different first itemset are different, which ensures that we do not count one sequence twice.

  The first itemset can have size of $i = 1, 2, \cdots, k$. There are in total a number of $\binom{n}{i}$ different itemsets with i items.

- With any itemset of i items as its first itemset, a k-sequence has $k - i$ items in its other itemsets. These $k - i$ itemsets form a $(k - i)$-sequence. The total number of different $(k - i)$-sequences is `calcNumSequences`$(k - i)$. Therefore, with any itemset of size i as its first itemset, there are a number of `calcNumSequences`$(k - i)$ different k-sequences.

- In conclusion, the total number of possible k-sequences is

$$\texttt{calcNumSequences}(k) = \sum_{i=1}^{k} \binom{n}{i} \texttt{calcNumSequences}(k-i)$$

  which correspond to line 5 - 7 in Algorithm 2.1.

- For convenience, when $k = 0$, we define $\texttt{calcNumSequences}(0) = 1$. In this way, when $k = 1$, $\texttt{calcNumSequences}(1) = \sum_{i=1}^{k} \binom{n}{i} \texttt{calcNumSequences}(k-i) = \binom{n}{1} \texttt{calcNumSequences}(0) = \binom{n}{1} = n$, which is the correct answer for the number of 1-sequences.

For $k = 1, 2, 3, 4, 5$, the total number of different k-sequences is given explicitly in the table below.

[1] We use the following terminologies. We call each element of a sequence an itemset and call each element of an itemset an item. For example, if we have a sequence $\langle \{i_1, i_2\}, \{i_3\} \rangle$, we call $\{i_1, i_2\}$

k	**Number of Possible Sequential Patterns**
1	n
2	$\frac{3}{2}n^2 - \frac{1}{2}n$
3	$\frac{1}{6}\left(13n^3 - 9n^2 + 2n\right)$
4	$\frac{1}{8}\left(25n^4 - 26n^3 + 11n^2 - 2n\right)$
5	$\frac{1}{120}\left(541n^5 - 750n^4 + 455n^3 - 150n^2 + 24n\right)$

Problem 3

(a) Mining Motifs in Massive Time Series Databases

In this article, the authors introduced a new method to calculate a discrete representation of the time series data, based on which they proposed a new algorithm to efficiently mine motifs in time series data.

In section 2, the authors reviewed the definitions of some basic concepts to be discussed: time series, subsequence, match of subsequences, trivial match, and K-motifs. They then gave a brute-force algorithm for motif mining, and provided an analysis on its disadvantages in time and space complexity, which will be used for comparison with the proposed new algorithm.

In section 3, the authors provided a method to achieve dimensionality reduction and discretization. Their method involves 2 steps to transform the data into a discrete representation of reduced size.

- In the first step, the Piecewise Aggregate Approximation method (PAA) is used to reduce the data size. The idea of PAA is to divide the length of the time series into w sections, and use one data point to represent each one of the sections. PAA provides an approximated and smoothed representation of the original time series. Since we can set w to be less than the length of the time series, the size of the data set is reduced after applying PAA. The authors claim although PAA is simple and intuitive, its performance rivals many of the more sophisticated methods.
- In the second step, the method further transforms the data set by discretizing the result of PAA. The authors proposed a method based on the observation that the normalized time series data is approximately Gaussian. Then we can break the distribution into a equally likely sections, each represented by a symbol. Therefore, replacing the data points in the PAA result by these symbols, a subsequence can be transformed into a "word" of symbols.

In order to measure the similarity of the subsequences or words, the authors then defined the distance measures used in the proposed new algorithm. They modified the Euclidean distance measure and proposed a distance function $DR(\cdot,\cdot)$ for the PAA representation and a distance $MINDIST(\cdot,\cdot)$ for the symbolic representation. The authors claimed that the distance for the PAA representation lower bounds the Euclidean distance.

In Section 4, the authors proposed a new algorithm *EMMA* to find the 1-Motifs. They claimed that their algorithm can easily extend to the general K-Motifs. *EMMA* uses several techniques to reduce the algorithm complexity.

- First of all, on the dissimilarity calculation, it utilizes the triangle inequality to avoid unnecessary distance calculations.
- In addition, one main contribution of the new method is instead of using a distance matrix containing all distances, the algorithm creates smaller matrices containing distances between likely candidates. If the candidates turn out not to have the 1-Motif we want, then a new small matrix need to be computed, and we repeat previous steps. There is no guarantee that we can find the 1-Motif we need using any small matrix. Therefore, we need to repeat until we find the 1-Motif. Worst case scenario,

we need to construct a matrix of size $O(m^2)$, where m is the length of the time series. This would be as bad as the brute force algorithm. The authors, though, claimed that this would be very rare in real world and most likely the performance would be far better than $O(m^2)$. One trick the authors used for constructing the small matrices is to use a hash table, which serves the purpose of converting the time series to symbolic representation and calculating an address. Moreover, this method helps in clustering neighboring words.

- Another technique the authors used for speeding up is using a modified ADM method to prune away a large portion of unnecessary candidates. This would bring noticeable speedup in searching.

In the last section of the article, the authors provided experimental results. They proposed using an efficiency score that is the ratio of the numbers of calls of the Euclidean distance function between the new method and the brute-force method. They performed experiments on two datasets and showed that on those two data sets, the new algorithm produces speedup of one to two order of magnitude.

Critical Assessment

The article provided a way to reduce dimensionality on time series data and found a distance measure in the symbolic space that correlates with the Euclidean measure in the original data space. By using triangle inequality property, constructing small candidate distance matrices, and performing ADM pruning, we can see that the algorithm produced speedups of one to two orders in magnitudes on the 2 test sets provided by the authors. Therefore, we can see that the algorithm has better performance than the brute-force algorithm.

One disadvantage of the new algorithm proposed by the authors is that the distance measure they used in the symbolic space is based on the Euclidean distance in the original time series value space, which does not generalize easily to general distances. For example, if the cosine distance is more suitable to our mining task, we cannot easily find a corresponding distance for our symbolic space, and therefore we cannot use the new algorithm to achieve speedups.

Another disadvantage of the new algorithm is lack of interpretability. Using the new discretization method would make it hard for theoretical analysis and evaluation. Therefore, there is no theoretic guarantee on the performance of the algorithm.

One aspect the article lacks is experiments on other data sets. The algorithm works well on the two data sets discussed in the paper, but we do not yet know if the performance can be generalized to other data sets.

(922 words)

(b) A Symbolic Representation of Time Series, with Implications for Streaming Algorithms

In this article, the authors proposed a symbolic representation method (SAX) for time series that reduces data dimensionality, enables the use of distance measures correlated with the Euclidean distance, and are also suitable for streaming data time series.

In sections 1 and 2, the authors reviewed the usual tasks and the generic framework in the time series data mining. They discussed previous work in time series mining on their advantages and disadvantages. They claimed that no methods in the previous work on time series mining allow a distance measure that lower bounds a distance function on the original time series data. They also pointed out that many of the previous methods on time series data mining have other fatal flaws like high dimensionality or requirement on access to the whole data set, which limits their use on streaming data.

In Section 3, the authors proposed a new time series data representation method SAX, which has the desirable properties of dimensionality reduction and compatibility with distance measures that lower bound a distance on the original data space. The SAX method uses 2 steps to transform the original time series data into a symbolic representation.

- The first step of the SAX method uses a technique called Piecewise Aggregate Approximation (PAA). In this step, the length of the time series is divided into w sections of equal size. Each section is then set to be represented by the mean of the data points within the section. The idea is essentially the same with using piecewise constant function to approximate the original time series. Since w is usually set to be smaller than the time series, the use of PAA would result in a data set of smaller size. Although PAA is a very simple method, the authors claim that it has performance similar in quality with many more sophisticated algorithms.
- The second step in the SAX method is to discretize the result from PAA. In this step, the method relies on the fact that time series in general follow a Gaussian distribution. Therefore it is possible to divide range of the values in the PAA result into equally likely sections using breakpoints. Each equal-probability section is then assign with a symbol as its representation. Using these symbols, we can represent any subsequence as a "word", i.e. a concatenation of symbols.

Using the 2 steps above, we can find a new symbolic representation of the original data. Then the author defined the distance measures on the symbolic space. They defined a distance function $DR(\cdot,\cdot)$ on the PAA space, which lower bounds the usual Euclidean distance. Then base on that, a distance $MINDIST(\cdot,\cdot)$ is defined on the symbolic space. The *MINDIST* resembles the *DR* distance, except that it uses a small distance table to compute its values. The authors also raised the issue of choosing the correct parameter w and a, i.e. the number of the PAA approximation sections and the number of breakpoint sections in the Gaussian distribution. They did not provide any argument on how to efficiently determine the two paramters, but they provided a heuristic to find a good pair of w and a, based the the tightness of lower bound.

In section 3, in addition to the dimensionality reduction method given above, the proposed method also utilizes a method to reduce numerosity. The idea is since we have symbolic representation of the sequences, when performing sliding window subsequence extraction, not all subsequences need to be be stored. If one subsequence is the same with the subsequence in the next position, we do not need the store the latter

one. When we need to retrieve the location of the subsequence, we simply go to the location of the first occurrence, and then slide to the right. We stop whenever the next word, or subsequence, changes. This is the same idea used in text compression.

In the final section of the article, the authors provided experimental results. Experiments on different tasks including hierarchical cluster, partitional clustering, nearest neighbor classification, decision tree classification, query by content, anomalous behavior detection and motif discovery were conducted. The authors claimed in all the experiments they conducted, SAX gives competitive performance, comparing to classical methods applied to time series data.

Critical Assessment

In this article, the authors proposed a new type of symbolic representation of time series data, which has the advantages of reduced dimensionality/numeriosity, allowing distance measures on the symbolic space that lower bounds the Euclidean distance on the original time series data, and allowing data mining on streaming data.

One short coming of the proposed method is that there does not exist easy way to generate symbolic distance measure from distances other than the Euclidean distance. Therefore, we lack flexibility of choosing the proper distance measure for our task.

Another disadvantage of this type of discretization is that the resulting data representation lacks interpretability. Therefore, theoretical analysis and evaluation would be challenging, which results in no guarantee on the performance of the algorithm.

In addition, the article only provided an heuristics without rigorious argument on how to use the w and a, i.e. the number of approximation sections or "frames" in PAA and the number of breakpoint intervals for the symbol transformation. The authors proposed the use of tightness of lower bound on the parameter selection and tested the formula on 50 data sets. But it does not guarantee performance of SAX on other data sets.

(899 words)