Knowledge Transfer Using Bayesian Belief Network

HAIYI ZHANG
Jodrey School of Computer Science,
Acadia University,
Wolfville, Nova Scotia,B4P 2R6
CANADA

Abstract: - This paper presents a method of selective knowledge transfer using Bayesian belief network. An implementation of the method was developed and tested on a synthetic domain of tasks. The results of several experiments indicate that the method has some merits

Key-Words: - Knowledge Transfer, Bayesian beliefs, Machine Learning, KNN.

1 Introduction

The $k$-nearest neighbour algorithm ($k$NN) is a popular machine learning method. $k$NN considers every instance to be a point in an $n$-dimensional space, where $n$ is the number of input attributes. $k$NN is trained by simply storing training examples and it classifies a query instance $q$ based on the $k$ training examples that are closest to $q$. Three methods of life-long learning through knowledge transfer using $k$NN have previously been proposed (Caruana, 1993, Thrun, 1995, Silver, 2000). Knowledge is selectively transferred based on structural measures of relatedness at the task level. This paper introduces a method of selective knowledge transfer in the context of $k$NN, which is based on functional measures and at the classification level using virtual training instances.

In many applications the relationship between the attribute set and the class variable is non-deterministic. In other words, the class label of a test record cannot be predicted with certainty even though its attribute set is identical to some of training examples. This situation may arise because of noisy data or the presence of certain confounding factors that classification but are not included in the analysis. For example, consider the task of predicting whether a person is at risk for heard disease based on the person’s diet and workout frequency. In the paper we use an approach for modeling probabilistic relationships between the attribute set and the class variable. Bayes theorem—a statistical principle for combining prior knowledge of the classes with new evidence gathered from data is explained and the Bayesian belief network is used.

Machine learning systems often encounter insufficient training examples per task to develop a sufficiently accurate hypothesis. For example, a hospital may have records on only 100 patients with a particular type of heart disease. One approach to overcoming the deficiency of training examples is to utilize knowledge that has been acquired during the learning of previous tasks that are related. For example, assuming we have learned a model of identifying patients with high blood pressure; we can use its knowledge to help us to identify patients with heat disease. The process is to transfer the previously acquired knowledge (high blood pressure diagnosis) to the new and related learning task (heart disease diagnosis).

A thorough discussion of the fundamental theory of knowledge transfer has been provided along with a method of selective knowledge transfer in the context of $k$NN (Silver, 2000). In other papers, two similar methods are discussed and tested (Caruana, 1993, Thrun, 1995). All of these methods use the similarity between the distance metric (a structural measure) used in each task and do not consider the functional relationship between the output values of the tasks.

Only a few people have looked at knowledge transfer in the context of the $k$NN
algorithm and all methods previously proposed transfer based on structural measures at the task level. The paper [Sun, Yuan 2004] develops a new functional measure of relatedness at the classification level and uses the measure to achieve knowledge transfer between kNN tasks. The shortcoming is a naïve method is adopted to deal with duplicated instance, which are instances sharing the same set of input attributes from two or more tasks. In our paper we use Bayesian network to calculate the conditional probabilities.

2 Selective Knowledge Transfer from kNN Tasks

kNN doesn’t explicitly generalize the training examples to form a hypothesis. The knowledge of a kNN system is represented by a pool of instances. Therefore, the most natural way to transfer knowledge from previously learned kNN tasks is to utilize training instances from those tasks.

The functional similarity between the training instances of two tasks describes a degree of relatedness between the tasks. This relatedness can also be represented by conditional probabilities. For the example, let \( P(T_0 = + \mid T_1 = +) \) equal the probability that an instance of \( T_0 \) is positive given that an instance of \( T_1 \) is positive. Then we can express the relatedness of \( T_1 \) to \( T_0 \) by the conditional probabilities \( P(T_0 = + \mid T_1 = +) \), \( P(T_0 = - \mid T_1 = +) \), \( P(T_0 = + \mid T_1 = -) \) and \( P(T_0 = - \mid T_1 = -) \). We do not know the exact value of these conditional probabilities but they can be estimated by observing the primary task and secondary task training instances.

The method pf knowledge transfer in the context of KNN concept learning tasks is extended to multi-class learning problems. The rationale is the same, i.e. use conditional probabilities to determine the relatedness of virtual instances to the primary task. The only difference between concept learning tasks and multi-class tasks is the number of classes.

\*

Conditional Probability Distribution for Multi-class Tasks

Though we can still use the classic definition of conditional probability distribution, a new form is required for purpose of the multi-class problem. Formally:

**Definition: Conditional Probability Distribution for Multi-Class Tasks**

First, define the conditional probability distribution per classification, \( CPDC(T_i \mid T_j = O_j^m) \) as a function that takes the \( m \)th element of set \( M \), which contains all possible class values of \( T_i \), as the input and outputs the set:

\[
\{ S \mid S = O_n^0 \times P(T_0 = O_n^0 \mid T_i = O_m^j), \quad 1 < n \leq \|N\| \}
\]

where \( O_n^0 \) is the \( n \)th element of the set \( N \), which contains all possible class values possible output values of \( T_0 \)

Then, define \( CPDC(T_i \mid T_j) \) as a function that takes \( T_i \) as the input and outputs the matrix:

\[
\{ S \mid S = O_m^j \times CPDC(T_0 \mid T_i = O_m^j), \quad 1 < m \leq \|M\| \}
\]

where \( M \) is the number of possible output values of \( T_i \).

Essentially, CPDC is the conditional probability distribution and CPDT is the joint probability distribution. They are different from the classical definition of the conditional probability distribution in that both CPDC and CPDT output the actual class value with which the probability is associated.

As an example, the conditional probability distributions for the example in Section 3.2 can be expressed as:

\[
\begin{align*}
CPDC(T_0 \mid T_1 = +) &= \{\{+, 0\}, \{-, 1\}\}, \\
CPDC(T_0 \mid T_1 = -) &= \{\{+, 0.8\}, \{-, 0.2\}\} \\
CPDT(T_0 \mid T_1 = \{\{+, \{+, 0\}, \{-, 1\}\}, \{-, \{+, 0.8\}, \{-, 0.2\}\}\}
\end{align*}
\]

In general, the output value of virtual instances of \( T_0 \), which are generated from instances with class value \( v \) in \( T_i \), is \( CPDC(T_0 \mid T_1 = v) \).

3 Duplicated instances

In Paper (Sun Yuan, 2004), a naïve method is adopted to deal with duplicated instances, which are instances sharing the same set of
input attributes from two or more tasks. Bayes Networks are used to calculate the conditional probabilities here. For example, the more complex conditional probabilities have to be calculated such as \( P(T_0 \mid T_1 = + \land T_2 = -) \). The major research question is to find out a fast enough way to estimate the conditional probabilities with reasonable accuracy.

4 Empirical Studies

A. Experiment 1: Variation in Transfer from More and Less Related Tasks

This experiment examines the transfer of knowledge from two related tasks to the primary task, where one secondary task is more related to the primary task than the other. We expect the more related secondary task to benefit the primary task the most by generating the greater positive inductive bias. This should result in better generalization accuracy for primary task.

1) Tasks

T3 of Bitmap domain is the primary task. T0 and T1 are used as the previously learned secondary tasks. T3 has 200 training examples and a test set of 800 instances. T1 and T0 were previously trained using 1000 instances for each task. Based on the discussion of the Bitmap domain, T3 is considered more related to T0 than to T1.

2) Method

The experiment consisted of 10 repeated trials where each trial had the following steps:

1. Generate random training and test sets for T3
2. Train the \( k \)NN system by loading the training instances.
3. Test the generalization accuracy of \( k \)NN for T3 using the test set with \( k = 3 \)
4. Transfer knowledge from T0 by generating virtual instances for T0. The acceptance threshold of relatedness was set to 0.001 based on preliminary testing.
5. Test the generalization accuracy of \( k \)NN for T3 using the test set with \( k = 3 \)
6. Transfer knowledge from T1 by generating virtual instances for T0. The acceptance threshold of relatedness was set to 0.001 based on preliminary testing.
7. Test the generalization accuracy of \( k \)NN for T3 using the test set with \( k = 3 \)

3) Results

Table 1 and Figure 1 show that both secondary tasks improve the generalization accuracy of T3. T0 provides the most positive inductive bias to the T3’s hypotheses with an accuracy of 0.788 \((p = 0.000)\) as compared to the hypotheses developed with the aid of T1 with an accuracy of 0.771 \((p = 0.066)\). We conclude that knowledge transferred from the more related task, T0 is of greater value than that of T1.

<table>
<thead>
<tr>
<th>(k)</th>
<th>T0</th>
<th>T1</th>
</tr>
</thead>
<tbody>
<tr>
<td>T3</td>
<td>23</td>
<td>26</td>
</tr>
<tr>
<td>T3</td>
<td>34</td>
<td>36</td>
</tr>
<tr>
<td>T3</td>
<td>84</td>
<td>81</td>
</tr>
<tr>
<td>T3</td>
<td>31</td>
<td>31</td>
</tr>
<tr>
<td>T3</td>
<td>12</td>
<td>81</td>
</tr>
<tr>
<td>T3</td>
<td>39</td>
<td>59</td>
</tr>
<tr>
<td>T3</td>
<td>12</td>
<td>36</td>
</tr>
<tr>
<td>T3</td>
<td>31</td>
<td>55</td>
</tr>
<tr>
<td>T3</td>
<td>31</td>
<td>32</td>
</tr>
<tr>
<td>T3</td>
<td>12</td>
<td>9</td>
</tr>
</tbody>
</table>

Table 1. Results of Experiment 1. The generalization accuracy of T0 before and after the knowledge transfer with \( k = 3 \)

Figure 1. Results of Experiment 1. Mean generalization accuracy of T3 before and after the knowledge transfer from either T0 or T1.

B. Experiment 2: Knowledge Transfer from Multiple Tasks

Previous experiments focused on transferring knowledge from one secondary task to a primary task. In this experiment, we examine the effect of transferring knowledge from several secondary tasks to a primary task, where the secondary tasks vary in their degree of relatedness.
1) **Tasks**

T₃ of the Bitmap domain is the primary task. T₀, T₁ and T₂ in the same task domain are the previously learned secondary tasks. T₃ had 200 training instances and a test set of 800 instances. T₀, T₁ and T₂ were previously trained by 1000 instances for each task to an accuracy of 0.85. We expect that T₃ will receive a net benefit from the transfer, because the method will promote positive inductive bias from related tasks and mitigate the negative inductive bias from unrelated tasks.

2) **Method**

The experiment consisted of 10 repeated trials where each trial had the following steps:

1. Generate random training and test sets for T₃
2. Train the kNN system by loading the training instances.
3. Test the generalization accuracy of kNN for T₀, T₁ and T₂ using the test set with k = 3
4. Transfer knowledge from T₂ by generating virtual instances for T₀. The acceptance threshold of relatedness was set to 0.001 based on preliminary testing.
5. Test the generalization accuracy of kNN for T₀ using the test set with k = 3

3) **Results**

Table 2 and figure 2 show that the generalization accuracy of T₀ is improved after transferring knowledge from all secondary tasks (p = 0.000). In addition, there is also some evidence to suggest that knowledge transfer from all three tasks improved the generalization accuracy of T₃ more than the knowledge transfer from just T₀ (p = 0.162).

<table>
<thead>
<tr>
<th>Trials</th>
<th>T3 alone</th>
<th>With ALL</th>
<th>With T0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.796504</td>
<td>0.815231</td>
<td>0.815231</td>
</tr>
<tr>
<td>2</td>
<td>0.762797</td>
<td>0.826467</td>
<td>0.803995</td>
</tr>
<tr>
<td>3</td>
<td>0.771536</td>
<td>0.787765</td>
<td>0.799001</td>
</tr>
<tr>
<td>4</td>
<td>0.765293</td>
<td>0.795256</td>
<td>0.791511</td>
</tr>
<tr>
<td>5</td>
<td>0.741573</td>
<td>0.780275</td>
<td>0.774032</td>
</tr>
<tr>
<td>6</td>
<td>0.751561</td>
<td>0.801498</td>
<td>0.789014</td>
</tr>
<tr>
<td>7</td>
<td>0.742821</td>
<td>0.813983</td>
<td>0.812734</td>
</tr>
<tr>
<td>8</td>
<td>0.765293</td>
<td>0.792759</td>
<td>0.790262</td>
</tr>
<tr>
<td>9</td>
<td>0.746567</td>
<td>0.766542</td>
<td>0.781523</td>
</tr>
<tr>
<td>10</td>
<td>0.776529</td>
<td>0.820225</td>
<td>0.805243</td>
</tr>
</tbody>
</table>

Accuracy Experiment 2

4 Conclusion

**Weighted distance.** ND-kNN algorithm can be easily extended to a weighted distance version. Using the weighted distance version, the virtual instance can be further weighted by its distance to the query instance. Therefore, the nearer a virtual instance is to a query instance, the more strongly it can affect the accuracy of
classification. One would have to consider the over-amplification of virtual instances by placing a limit on the maximum distance weight.

**Density of virtual instances.** \(k\)NN derives decision boundaries from training examples. One important factor that greatly affects the shape of the decision boundary is the density of instances. If the existing decision boundary happens to be the optimal one, one more example may decrease the generation accuracy; this is similar to overtraining in ANN. To accommodate this situation, the virtual instances could be generated in such a way that the density of instances is constant throughout the input space. Other techniques such as Model-based \(k\)NN (Guo, Wang, Bell, Bi, & Greer, 2003) would also help.

**Combining structural and functional measures of relatedness.** The measure of relatedness suggested by previous research captures the structure similarity between two \(k\)NN tasks while the CPDT captures the functional similarity. It would seem important to consider both methods when transferring knowledge between tasks. Future research could investigate a combination of these two methods.

References: