Knowledge Source Selection By Estimating Distance Between Datasets

Yi-ting Chiang*, Wen-Chieh Fang† and Jane Yung-jen Hsu‡

Department of Computer Science and Information Engineering
National Taiwan University
Taipei, Taiwan
Email: wcf123@ntu.edu.tw†, {yitingchiang*, prof.hsu‡}@gmail.com

Abstract—Most traditional machine learning methods make an assumption that the distribution of the training dataset is the same as the applied domain. Transfer learning omits this assumption and is able to transfer knowledge between different domains. It is a promising method to make machine learning technology become more practical. However, negative transfer can hurt the performance of the model, therefore, it should be avoided. In this paper, we focus on how to select a good knowledge source when there are multiple labelled datasets available. A method to estimate the divergence between two labelled datasets is given. In addition, we also provide a method to decide the mappings between features in different datasets. The experimental results show that the divergence estimated by our method is highly related to the performance of the model.

Keywords-transfer learning; similarity measure; machine learning

I. INTRODUCTION

Data mining and machine learning technologies have been widely used in a broad spectrum of practical domains. For example, document classification methods are brought into practice in spam detection software. Siri, a famous intelligent personal assistant application developed by Apple, utilizes speech recognition, natural language processing, and information retrieval technologies. It is to be expected that more and more applications we can use in our daily lives will be involved more with machine learning methods in the near future.

Most traditional machine learning methods make an assumption that the distribution of the training dataset is the same with that of the applied model. In addition, many machine learning methods require sufficient amount of samples to ensure their models perform accurately, especially for those who use statistical data modelling technologies. Data collection, however, can be very costly for many applications, especially when it is necessary to not only collect data samples, but also annotate these samples to apply supervised learning methods. These restrictions limit the regions where the model can be put into practical use.

Various promising schemes and methods have been proposed promising to remove these restrictions. For example, active learning reduces the cost of annotating data samples by actively requiring users to annotate only some important samples. Semi-supervised learning utilizes the labelled and unlabelled data samples. Labelled data samples can provide information about how to make use of the unlabelled data samples in the learning process. Multi-task learning learns several tasks at the same time. By exploiting the relatedness between different learning tasks, it is possible to learn these related tasks simultaneously.

Transfer learning, on the other hand, releases the assumption of an identical distribution between the training and the testing datasets, and utilizes various methods to transfer knowledge between different domains. It focuses on learning information from one environment to facilitate the learning process on the other. In other words, reusable knowledge is extracted from a source domain and applied to the learning process in a target domain. Negative transfer, which means transferring knowledge between two domains which are not sufficiently related, should be avoided. Therefore, transfer learning deals with the problem of not only knowledge extraction but also knowledge evaluation to facilitate the learning process.

A. Related Work

Transfer learning has been successfully applied in some machine learning problems, such as the document classification problem [1][2][3][4], sentiment analysis [5], collaborative filtering [6], location system [2], image classification [7], and activity recognition[8][9][10][11]. A good survey and categorization of transfer learning can be found in [12], and [13] is a survey for activity recognition problems using transfer learning.

Various theoretical analysis of transfer learning have also been carried out. In [14], an analysis on the error bound of transferring knowledge between different tasks is given. The authors assumed that a disparity matrix $D$, which $D(i,j)$ bounds the difference between source $i$ and source $j$, is given. Specifically, for any two models $m_i$ and $m_j$, the expected loss of $m_j$ on the target function $m_i$, denoted by $e(m_i, m_j)$, can be bounded by $D(i,j)$ such that $e(m_i, m_j) \leq D(i,j)$.

There are some measures to estimate the difference of two distributions. A well-known method, the Kullback-Leibler divergence, for two distributions $P$ and $Q$ is defined as

$$KL(P,Q) = \sum_x P(x) \log \frac{P(x)}{Q(x)}$$
KL divergence is always non-negative. However, it is not symmetric. In addition, it is not bounded. Some new divergence measures have been proposed based on KL divergence. One of these measures is the Jenson-Shannon divergence [15], JSD$(P, Q)$, which is defined as

$$JSD(P, Q) = \frac{1}{2} \left( KL(P, \frac{P + Q}{2}) + KL(Q, \frac{P + Q}{2}) \right)$$  \hspace{1cm} (1)$$

It is easy to see that Jenson-Shannon divergence is non-negative, symmetric and bounded.

In this paper, we focus on how to select a knowledge source when there are multiple labelled datasets available. We estimate the divergence between different domains given their datasets. Based on the estimated divergence, we can choose a nearby source domain for the target domain. In addition, we also propose a method to decide the mapping between features in different datasets, thus one can reformulate the datasets according to the mapping and apply traditional machine learning methods to transfer knowledge.

In the following sections, we first introduce our methodology of estimating the divergence between two labelled datasets in section II. After that, we show our experimental results, including the method we used to measure the distance of datasets, what datasets are used in our experiments, the dataset preprocessing procedures, and the machine learning methods we adopted in section III. Finally, we give the conclusion and future work in section IV.

II. METHODOLOGY

If there are multiple source domain datasets, and we know a method for estimating the divergence between the source and the target domain datasets, we may choose a better source domain according to their divergence. In this section, a method for estimating the divergence between two datasets is proposed. Roughly speaking, we provide a method to decide a mapping between features of two datasets, and compute the divergence between the two datasets accordingly. The divergence of datasets varies with different feature mappings. Two possible criteria for computing the feature mapping are proposed, and two graph matching algorithms are introduced for each of these criteria.

We first define what is a mapping between two feature sets:

**Definition 1.** A mapping $\mathcal{M} : \mathcal{F}_1 \rightarrow \mathcal{F}_2$ is a binary relation between two feature sets $\mathcal{F}_1$ and $\mathcal{F}_2$. If $(f, g) \in \mathcal{M}(\mathcal{F}_1, \mathcal{F}_2)$, we say feature $f \in \mathcal{F}_1$ and $g \in \mathcal{F}_2$ are mapped together.

If a mapping is computed, we can use it to transfer knowledge between two domains. The divergence between two datasets should be a significant indicator to analyse the mapping between two domains. Let $\mathcal{F}$ and $\mathcal{G}$ be two feature sets, and $H_{\mathcal{F}, \mathcal{G}}(\mathcal{F}; T)$ be the divergence between $\mathcal{F}$ to $\mathcal{G}$ under a task $T$. We estimate $H_{\mathcal{F}, \mathcal{G}}(\mathcal{F}; T)$ by using the following equation:

$$H_{\mathcal{F}, \mathcal{G}}(\mathcal{F}; T) \approx \sum_i H_{\{f_i\}, \{g_i\}}(\{f_i\}; T),$$  \hspace{1cm} (2)$$

where $\forall i, (f_i, g_i) \in \mathcal{M}(\mathcal{F}, \mathcal{G})$. This estimated divergence is used as the distance of two datasets under the task in the target domain. Equation 2 is a feature-based divergence estimation. Assuming we know how to estimate the divergence of features between two datasets under a task, the divergence of these two datasets can be estimated by summing the divergence of each pair of features which are mapped together.

Assume that the divergence of any two features in two datasets is available, that is, for all $i$ and $j$, $H_{\{f_i\}, \{g_j\}}(\{f_i\}; T)$ is given, we reduce the problem of computing the feature mapping to graph matching problems on complete bipartite graphs. Assume there are $m$ features in feature set $\mathcal{F}$ and $n$ features in feature set $\mathcal{G}$, we define a complete bipartite graph $K_{m,n} = (U, V, E)$, where $|U| = m$ and $|V| = n$. Vertices $u_i \in U$ and $v_j \in V$ represent features $f_i \in \mathcal{F}$ and $g_j \in \mathcal{G}$, respectively. We also assign weight values to all edges in $E$. Let $w(u_i, v_j)$ be the weight of edge $(u_i, v_j)$, we have

$$w(u_i, v_j) = H_{\{f_i\}, \{g_j\}}(\{f_i\}; T)$$  \hspace{1cm} (3)$$

If $u_i$ and $v_j$ are matched in the graph $K_{m,n}$ according to a graph matching algorithms, then feature $f_i$ and feature $g_j$ are mapped together, as defined in definition 1. By this reduction, we can solve our feature mapping problem by solving graph matching problems.

Two criteria are possible to decide which features should be mapped together. In one criterion, we want to minimize the estimated distance of two datasets. In the other criterion, we want to map two least divergent features together first. Two graph matching algorithms are available for each of the two criteria. We introduce both of them in details in the following subsections.

A. Graph Matching Algorithms

Before introducing graph matching algorithms, we explain some terminology in graph theory first. In graph theory, a matching is a set of edges that any two edges can not share one vertex. Given a graph $G(V, E)$, a matching of $G$ is a subset of $E$ that if two edges $(v_i, v_j)$ and $(v_r, v_s)$ are in the subset, $i \neq r, s$ and $j \neq r, s$. A perfect matching $E_p$ is a matching that no vertex is left behind in the matching. That is, for each $v_i \in V$, we have one and only one edge $e \in E_p$ that $e = (v_i, v_j)$ or $e = (v_j, v_i)$.

1) Minimum Cost Perfect Matching: A minimum cost perfect matching [16] problem is problem of finding a perfect matching with minimum cost. That is, given a weighted graph $G$, algorithms for this problem find a matching in the graph $G$ such that the summation of the weights of these edges in the matching is minimized. Formally, let $w_i$ be the
weight of edge $e_i$, the perfect matching $E_p$ is a set of edges
in a perfect matching $E_k$ with the property:

$$E_p = \arg\min_{E_k} \sum_{e_i \in E_k} w_i$$

(4)

2) Stable Marriage: The stable marriage problem \[17\] in graph theory is a problem of finding a stable matching
between two sets of vertices, $V_a$ and $V_b$. Let $a_i$, $b_j$ and $b_k$ be
three vertices with edges $(a_i, b_j)$ and $(a_i, b_k)$ whose weight
values are $w_{i,j}$ and $w_{i,k}$ respectively. In stable marriage
problem, $a_i$ prefers $b_j$ to $b_k$ if $w_{i,j} < w_{i,k}$. A matching
is stable if when an edge $(a_i, b_j)$ is in the matching, there is
no edge $(a_j, b_j)$ in the matching such that $a_i$ prefers $b_j$ to $b_i$ and $b_j$ also prefers $a_i$ to $a_j$. Note that a stable matching
may not have minimum total cost, as the example we show
in figure 1.

**Figure 1.** A stable marriage matching $(A, D), (B, C)$ is with cost 105,
which is not a minimum cost matching.

Choosing different graph matching algorithms has different meanings for knowledge transfer, thus is appropriate for
different criteria. We can see the difference by observing figure 1. If the algorithm for minimum cost perfect match
problem is applied, we find a mapping in a global view that the total divergence between two feature sets is minimized,
which is the mapping for the first criteria. On the other hand, the stable marriage algorithm aligns the most preferred
features in two datasets first. Therefore, the total divergence may not be minimized, but the most preferred pairs of features
will not be sacrificed. The mapping computed by using this algorithm is for meeting the second criteria.

### III. Experimental Results

In this section, we describe the experiments conducted. We introduce the datasets we used, the data preprocessing
procedures, the machine learning models, and the results of our experiments in this paper. We also present the method
of how we estimate the distance between two datasets in our experiments.

#### A. The Datasets

In order to verify the method we propose to estimate the similarity between datasets which can be used to choose a
better source to transfer knowledge, we run our experiments on three public handwriting digits datasets, the semeion
handwritten digit dataset (SEM), the optical recognition of

handwritten digits dataset (OPT), and the pen-based recognition of handwritten digits dataset (PEN). These datasets are
all available on the UCI Machine Learning Repository \[18\].

In the SEM dataset, there are 1593 handwritten digits
from around 80 persons. In the raw file, each row has 256 binary features which represents a 16 by 16 bitmap, and is
annotated by using 10 binary columns which indicate the represented digits.

The OPT dataset is collected from 43 people, 30 of which contributed to the training set and the other 13 to the test set.
We used the preprocessed version of this dataset in the UCI data repository. In this dataset, there are 64 features columns
followed by one label column. Each feature represents a 4 by 4 block of a 32 by 32 bitmap, so the values of these features
are between 0 to 16, which indicates how many pixels are painted in this bitmap.

Unlike the SEM and OPT datasets, each digit in the PEN dataset is recorded by 8 pairs of points $(x, y)$ which is the
sequence of the stroke of this digits. The creators of this dataset resampled the stroke of the digits to make the arc
length between each point the same. The values of both $x$ and $y$ are between 0 to 100.

#### B. Preprocessing Datasets

For convenience, we converted all these datasets to have the same number of features.

Since the SEM and OPT datasets record digits by bitmap format, and the PEN dataset uses a stroke format. We
need first convert the PEN dataset into bitmap format by interpolating points between any two points in the dataset.
Each converted record contains a 101 by 101 bitmap, and each pixel is with a binary value. After that, we combine the
24th, the 49th, the 74th, the 87th, and the 100th pixel in both $x$-axis and $y$-axis with the 23rd, the 48th, the 73rd the 86th,
and the 99th pixel, respectively. The resultant value will be 0 if and only if both the two values in the two merged pixels
are 0. This procedure generates a 96 by 96 bitmap for each digit.

Finally, 64 non-overlapping blocks are extracted from each bitmap. Each block is a 12 by 12 sub-bitmap of the
original bitmap. The new dataset contains 64 features. The range of these feature values are from 0 to 144.

In the SEM dataset, records are already bitmaps. Similar to what we did on the PEN dataset, we divide each record in
the SEM dataset into 64 non-overlapping blocks. Each block is a 2 by 2 sub-bitmap in the original 16 by 16 bitmap. The
result is also a dataset with 64 discrete feature values. The values are 0, 1, 2, 3, and 4.

Finally, we want to make all the three datasets to have the same range of feature values. Feature values in the PEN and
the OPT dataset are adjusted to 0 to 4 as in the SEM dataset. Consequently, we have three datasets. All of which has
64 features with the same range of values.
### C. Measuring Dataset Distance

As mentioned in section II, we used a feature-based method to estimate the divergence of two datasets. Given the divergence of any two features in the source and the target domain datasets, we can choose one of the graph matching algorithm we introduced in section II to compute the mapping according to the criterion we adopted in the knowledge transferring process.

The distance estimation method we used to estimate the divergence between two features is based on the Jenson-Shannon divergence we introduced in equation 1. Specifically, we compute the expected Jenson-Shannon divergence of \( P_f \) and \( Q_f \) given the label \( C \), which is defined as

\[
E[\text{JSD}(P_f, Q_f|C)] = \sum_i f_i(c_i) \text{JSD}(P_f, Q_f|C = c_i),
\]

where \( f_i \) is the probability distribution function of the label \( C \) in the target domain dataset. We estimate \( f_i \) using only the training data of the target domain dataset in our experiments.

We use the \( E[\text{JSD}(P_f, Q_{bg}|C)] \) as the weight of edge \((u_i, v_j)\) of the bipartite graph. Recall equation 3 in section II, we have

\[
w(u_i, v_j) = H_{\{f_i, \{g_i\}\}}(\{f_i\}; T) = E[\text{JSD}(P_f, Q_{bg}|C)],
\]

and we can run a graph matching algorithm to compute the mapping of features now. The distance of two datasets is computed according to the mapping.

### D. The Experiments

The purpose of our experiments is to verify whether our distance measurement for datasets is consistent with the accuracy of the model. If the two datasets are near to each other, we should be able to transfer more useful knowledge from one dataset to another when training the model, and have better accuracy.

In our experiments, we train models from the training data of the source dataset, and apply the model to the target domain datasets. In both the OPT and the PEN dataset, there are different files for the training and the testing data. But in the SEM dataset, there is only one file with 1593 samples. Therefore, we only use the SEM dataset as the source domain dataset. The training-testing combinations are shown in table I. The distance between the source domain dataset and the target domain training data after mapping by the minimum cost perfect matching algorithm and the stable marriage algorithm is given in table II. This table is sorted by the estimated distance. Since the mapping of features between two datasets is used to estimate the distance between these two dataset, the estimated distance will vary with different mappings. We have a total of 8 rows in table II, since the SEM dataset is used only as the source domain dataset.

### Table I

<table>
<thead>
<tr>
<th>Source</th>
<th>Target</th>
<th>Matching Algorithm</th>
<th>Estimated Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEM</td>
<td>PEN</td>
<td>Mincost</td>
<td>8.687377</td>
</tr>
<tr>
<td>SEM</td>
<td>PEN</td>
<td>Stable</td>
<td>9.355471</td>
</tr>
<tr>
<td>SEM</td>
<td>OPT</td>
<td>Mincost</td>
<td>11.363237</td>
</tr>
<tr>
<td>SEM</td>
<td>OPT</td>
<td>Stable</td>
<td>12.559258</td>
</tr>
<tr>
<td>PEN</td>
<td>OPT</td>
<td>Mincost</td>
<td>16.692963</td>
</tr>
<tr>
<td>PEN</td>
<td>OPT</td>
<td>Stable</td>
<td>16.707949</td>
</tr>
<tr>
<td>PEN</td>
<td>PEN</td>
<td>Stable</td>
<td>17.755014</td>
</tr>
</tbody>
</table>

SVM and naïve Bayes models are chosen as our machine learning models. The toolkits we used are libsvm [19] and Weka [20]. We first compute the best mapping between two datasets using either the minimum cost perfect algorithm or the stable marriage algorithm, and reorder the sequence of features in the target domain dataset (both the training and the testing data file) accordingly. Next, we train a model using the source domain dataset (only the training data if the PEN or OPT datasets are the source), and test the accuracy of this model. The result is shown in figure 2(a) and figure 2(b).

Note that there are 8 groups of points on each of the two figures, each corresponds to one row in table II. In each group, there are 3 points in figure 2(a) and figure 2(b). A point is a combination of the source domain dataset, the target domain dataset, and the algorithm used to compute the mapping. For example, the distance of the left most points of the three lines in figure 2(a) is 8.687377. Referring to table II, these points belong to the group of the experiments that used the SEM dataset as the source domain dataset, and the PEN dataset as the target domain dataset, with the minimum cost perfect matching algorithm as the graph matching algorithm. The two points in this group on the blue and the red lines are the results of training the model using the SEM dataset, and testing this model on the training data and the testing data, respectively. Another point on the yellow line is the result without transfer learning, that is, the model is trained using the training data in dataset PEN, and is tested on the testing data of the same dataset. Note
that the dataset distance of the training and the testing data in dataset PEN is not 8.687377. For comparison, we put this point at the corresponding locations in figure 2(a) and figure 2(b).

![Distance v.s. Accuracy SVM](image1.png)  
(a) The result of SVM

![Distance v.s. Accuracy Naive Bayes](image2.png)  
(b) The result of naïve Bayes model

Figure 2. The results of distance and the accuracy values on two models.

From figure 2(a) and figure 2(b), we can find that both SVM and naïve Bayes model have similar patterns on the accuracy and the distance. Obviously, the results without transfer learning, which are the yellow lines in the two figures, are not related to the distance between the source and the target domain datasets. In addition, the results of using training data to test the model are slightly better than that of using the testing data. This is reasonable because we compute the distance of datasets using only the training data.

In the case of using the same datasets as the training and the testing dataset, but different graph matching algorithms causes different results. It can be found that the experiments using minimum cost perfect matching algorithm outperforms that of using the stable marriage algorithm in most cases. Minimum cost perfect matching algorithm computes the mapping of features which minimized the distance of two datasets, therefore this result is not beyond our expectation.

Moreover, The accuracy values of the blue and the red lines decrease along with the distance of datasets. When the distance values are larger than 16, the accuracy values become unstable. Since this phenomenon occurs not only in the result of transfer learning experiments (the blue and the red lines), but also can be found in the cases without transfer learning (the yellow line). Therefore the cause should not be from the feature mapping procedure that we applied to these datasets.

To eliminate possible factors that may come from using different datasets, we run another experiment on only the SEM and the PEN datasets. In this experiment, we transfer knowledge from the SEM dataset, and test the model using the PEN testing file. Instead of just using the mapping computed by the graph matching algorithms, we also generate many other feature mappings. For each mapping, we compute the distance between the SEM and the PEN datasets. Afterwards, a model is trained using the SEM dataset under the mapping, and is tested on the PEN testing data. The relevance of the distance of datasets and the accuracy of the model can be found in table III and in figure 3. We can see that as the distance increases, the accuracy value decreases.

IV. CONCLUSION AND FUTURE WORK

In this paper, we showed that it is possible to provide a measurement of dataset distance to choose a knowledge source in transfer learning. We proposed a distance measure for datasets by estimating the expected Jenson-Shannon divergence of their features, and computed the best mapping

<table>
<thead>
<tr>
<th>Distance value</th>
<th>Center</th>
<th>Mean</th>
<th>Standard deviation</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>9.199201</td>
<td>0.017933</td>
<td>0.593739</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>10.109127</td>
<td>0.065426</td>
<td>0.481749</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>11.130477</td>
<td>0.027611</td>
<td>0.352065</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>11.565707</td>
<td>0.013208</td>
<td>0.350348</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>13.359726</td>
<td>0.030129</td>
<td>0.138656</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>14.231069</td>
<td>0.020077</td>
<td>0.104897</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>14.586383</td>
<td>0.019062</td>
<td>0.086255</td>
<td></td>
</tr>
</tbody>
</table>

Figure 3. The distance values and the accuracy value
by a graph matching algorithm. The effectiveness of the distance measure is verified by applying transfer learning by using SVM and naive Bayes models. All datasets we used in this paper are available in the UCI Machine Learning Repository. The experimental results in this paper show that the distance we estimated between two datasets is related to how well the model performed.

However, there are obviously some problems with our distance measure. We compute the distance between two datasets as the sum of the divergence of their mapped features. As a result, the number of features can affect their distance. In addition, if there are two features $f_i$ and $f_j$ in the same dataset which are totally identical, the knowledge they transfer to the target domain is redundant. In these cases, the distance of datasets is overestimated.

A. Future Work

In order to accurately estimate the distance of two datasets, further refinement is necessary. The dependency of features in the same datasets is a problem we need to deal with. Reducing the amount of redundant information provided by two features should increase the precision of the distance Estimation. Another possible solution is to define a new distance measure which is independent from the number of features. Finally, a more general solution is to identify the necessary properties of the distance measures which can be used to estimate the distance of datasets.

References


