# Relaxed Transfer of Different Classes via Spectral Partition

Xiaoxiao Shi[1,⋆], Wei Fan[2], Qiang Yang[3], and Jiangtao Ren[4]

[1] Department of Computer Science
University of Illinois, Chicago, USA
xshi9@uic.edu
[2] IBM T.J.Watson Research, USA
weifan@us.ibm.com
[3] Department of Computer Science
Hong Kong University of Science and Technology
qyang@cse.ust.hk
[4] Department of Computer Science
Sun Yat-sen University, Guangzhou, China
issrjt@mail.sysu.edu.cn

**Abstract.** Most existing transfer learning techniques are limited to problems of knowledge transfer across tasks sharing the same set of class labels. In this paper, however, we relax this constraint and propose a spectral-based solution that aims at unveiling the intrinsic structure of the data and generating a partition of the target data, by transferring the eigenspace that well separates the source data. Furthermore, a clustering-based KL divergence is proposed to automatically adjust how much to transfer. We evaluate the proposed model on text and image datasets where class categories of the source and target data are explicitly different, e.g., 3-classes transfer to 2-classes, and show that the proposed approach improves other baselines by an average of 10% in accuracy. The source code and datasets are available from the authors.

## 1 Introduction

Traditional supervised and semi-supervised learning work well under the strict assumption that the labeled training data and unlabeled test data are drawn from the same distribution and have shared feature and category spaces. In many real world applications, however, this assumption may be violated. In fact, we often encounter the situations where we do not have sufficient labeled training examples in the target learning task. Examples include spam filtering, biological sequence annotation, web searching, and the like. To acquire more labels could usually be expensive or infeasible. For example, in the field of computational biology, many expensive and time-consuming experiments are needed to provide the labels for even a small number of examples. As an alternative solution, transfer learning was proposed to help extract some supervisory knowledge from

⋆ Part of the work was done when the author was a visiting student at HKUST.

W. Buntine et al. (Eds.): ECML PKDD 2009, Part II, LNAI 5782, pp. 366–381, 2009.

related source data to help learn the target task (e.g., [3,4,16,22]). Existing transfer learning techniques implicitly assume that there is sufficient overlap between the source data and the target data, and categories of class labels are the same, in order to allow the transfer of knowledge. However, this can significantly limit the applicability of transfer learning, as it is not always possible to find labeled data satisfying these constraints. In order to improve its applicability, we study how to transfer knowledge across tasks having different class categories.

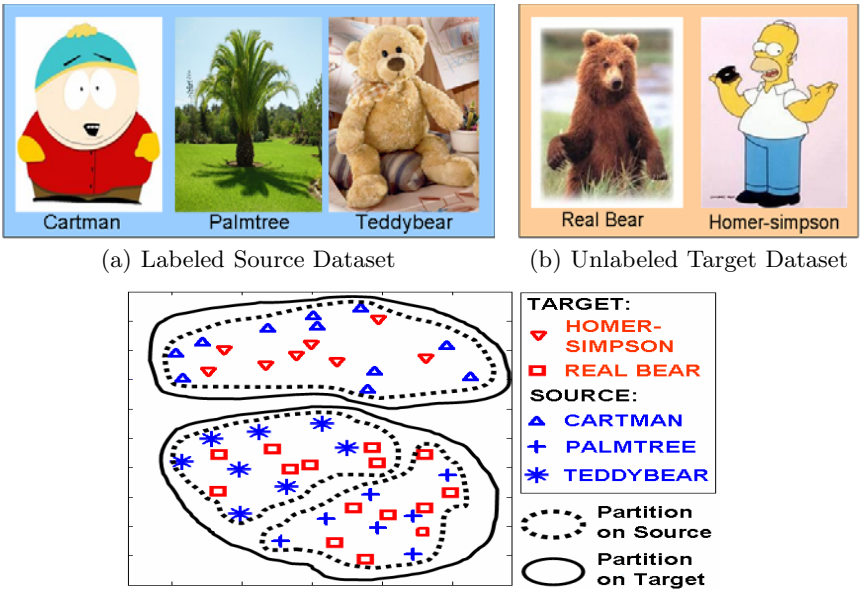
For example, can the text documents labeled in "wikipedia" help classify those documents in "ODP[1]" even though they have different index systems? Can the labeled source image data in Fig 1(a) help classify the target data in Fig 1(b) given that they are images of different objects? The problem formulation is to partition an unlabeled target data, by the supervision from a labeled source data that has different class categories. To solve the problem, two issues need to be addressed:

1. What and how to transfer? Since the source and target data have different class labels, we can not directly take advantage of the class conditional density $p(\mathbf{x}|y)$ or posterior $p(y|\mathbf{x})$ to construct the model, and thus most of the previous transfer learning methods do not work. A new transfer learning strategy independent of class labels is needed.
2. How to avoid "negative transfer"? Given that the source and target data do not share class labels, they may come from significantly different domains. Thus, it is also necessary to avoid negative transfer (or accuracy worse than no transfer) when the source and target data are unrelated.

We propose a spectral-based solution that uses eigenspace to unveil the intrinsic structure similarities between source and target data. The key idea is that, regardless of their class category naming, if the source and target data are similar in distribution, the eigenspace constructed from the source data should be also helpful to reflect the intrinsic structure of the target data. We illustrate the intuition in Fig 1. Although the target data (Fig 1(b)) and source data (Fig 1(a)) have totally different class labels, the eigenspace Fig 1(c) constructed with the supervision from Fig 1(a) still helps group the target data. On the one hand, the images about homer-simpson are similar to the images about cartman because they are all cartoon characters; on the other hand, the shape of real bear is similar to teddybear, and the background of real bear may contain plants similar to palm tree. Thus, the eigenspace that well separates Fig 1(a) also helps separate Fig 1(b) even though their class labels are different.

To be specific, the proposed model finds an eigenspace through a combination of two optimization objectives. The first is to find the eigenspace that well separates the source data: the labeled data with the same class categories will be grouped together. The second objective is to maximize the marginal separation of the unlabeled target data. Moreover, to avoid negative transfer, we also derive a clustering-based Kullback-Leibler divergence to measure the difference in distribution between two finite datasets more effectively (see Lemma 1). We then

[1] "Open Directory Projects" (http://www.dmoz.org/). Both "wikipedia" and "ODP" are systems categorizing large amount of documents.




(a) Labeled Source Dataset (b) Unlabeled Target Dataset

(c) Solution in eigenspace

**Fig. 1.** An example on two image datasets with totally different class labels. Fig 1(c) is the eigenspace constructed with the supervision from the source dataset, and we plot a sub set of examples to illustrate the intuition. In this example, the eigenspace learnt from source dataset (Fig 1(a)) can also reflect the structure of the target dataset (Fig 1(b)) though their class labels are different.

make use of the measure to define "transfer risk" to regulate the effects of the two objectives. When the two datasets are very different, the effect of the first objective (or the supervision from source task) automatically decreases to minimize the risk of negative transfer. We provide a PAC bound for the proposed method in Theorem 1. In addition, the proposed algorithm is tested in several datasets where target and source data have very different class categories. For instance, in one of the experiments, we apply the 4-classes document sets "Graphics vs. Hardware vs. Politics.mid vs. Religion.misc" to supervise the partition of the binary document datasets "Comp vs. Rec". The proposed model achieves an accuracy 98%, while the accuracy of the baseline that does not apply transfer learning is only 74%.

## 2 Problem Formulation

We consider the problem to find a good partition of the unlabeled target data, possibly with the supervision from the labeled source data having different class labels, only when the supervision is helpful. We denote the source data as $\mathcal{L} = \{\mathbf{l}_1, \mathbf{l}_2, ..., \mathbf{l}_s\}$, and the target data as $\mathcal{U} = \{\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_t\}$, where $s$ and $t$ are

the sample size for the source and target data, respectively. Both $\mathcal{L}$ and $\mathcal{U}$ are drawn from the same feature space $R^d$ under different distributions $\mathcal{L} \sim P_{\mathcal{L}}$ and $\mathcal{U} \sim P_{\mathcal{U}}$. We also denote that the class labels of $\mathcal{L}$ are $\mathcal{Y} = \{y_1, y_2, ..., y_r\}$ where $r$ is the number of categories of $\mathcal{L}$, while the number of class labels of $\mathcal{U}$ is $k$. And we do not assume the class labels of source and target data are the same.

*No Transfer:* To generate a partition $C = \{c_1, c_2, ..., c_k\}$ of the target data $\mathcal{U}$, we find a clustering decision function $\mathcal{N} : \mathcal{U} \mapsto C$ to minimize $\varepsilon[\mathcal{N}(\mathcal{U})]$, where $\varepsilon[*]$ is a cost function to measure the quality of the partition. Example of such a function is Normalized Cut [19].

*Transfer Learning:* We learn the partition $\mathcal{U} \mapsto C$ by making optional use of the knowledge from the source data. There are many interpretations of transfer learning strategies, e.g., [3,4] reviewed in Section 5. One straightforward strategy is formulated as follows. We first learn a decision function $\mathcal{F} : \mathcal{L} \mapsto \mathcal{Y}$ on the feature space $R^d$ that correctly reflects the true label of the source data $\mathcal{L}$. A simple transfer learning strategy can be $\mathcal{T}(\mathcal{U}) = \mathcal{M}(\mathcal{F}(\mathcal{U}))$, where $\mathcal{M} : \mathcal{Y} \mapsto C$.

*Negative Transfer:* Since the source and target data have different class labels, they may be from very different domains. Thus, one of the main challenges is how to avoid negative transfer when the source and target data are too different. Formally, $\mathcal{L}$ and $\mathcal{U}$ may be very different so that the performance after transfer learning is worse than no transfer. Formally,

$$\varepsilon[\mathcal{N}(\mathcal{U})] < \varepsilon[\mathcal{T}(\mathcal{U})] \tag{1}$$

where $\varepsilon[*]$ is the cost function to measure the partition quality. There can be other criteria such as error rates in classification problem. When Eq (1) holds, it is said to have negative transfer [17].

## 3 Risk-Sensitive Spectral Partition

We propose an improved spectral partition model to transfer the eigenspace that well separates the source data, in order to generate a good partition of the target data. Importantly, source and target data have different class labels. Since negative transfer may happen, we use the "risk of negative transfer" to automatically decide when and how much to transfer.

### 3.1 Divergence Measure and Transfer Risk

We first derive a new formula of the KL-divergence and propose a clustering-based approach to calculate it, in order to quantify the difference between source and target data more effectively. Normally, given two probability distributions $P$ and $Q$, the KL divergence $\mathbf{KL}(P||Q)$ is defined as follows[2]:

$$\mathbf{KL}(P||Q) = \sum_x P(x)\big(\mathbf{log}P(x) - \mathbf{log}Q(x)\big) \tag{2}$$

[2] KL can be also written in terms of probability density in a continuous form, which is difficult to estimate without prior knowledge [1].

On finite datasets, one usually calculates $P(x)$ and $Q(x)$ for every value of $x$ via Eq (3).

$$\begin{aligned} P(x=a) &= \frac{|\{x|x=a \wedge x \in \mathbb{D}_P\}|}{|\{x|x \in \mathbb{D}_P\}|} \\ Q(x=a) &= \frac{|\{x|x=a \wedge x \in \mathbb{D}_Q\}|}{|\{x|x \in \mathbb{D}_Q\}|} \end{aligned} \tag{3}$$

where $\mathbb{D}_P \sim P$ and $\mathbb{D}_Q \sim Q$ are the datasets generated from the distributions $P$ and $Q$, respectively. However, small changes to the datasets, such as those sparse ones typically found in text mining, can result in significant changes by the above approximations, making it difficult to distinguish different distributions [1]. To resolve this problem, we derive another format of the KL divergence and propose a clustering-based approach to calculate it. We first perform a clustering on the combined dataset $\mathbb{D}_P \cup \mathbb{D}_Q$. We then directly employ some basic statistics of the clustering results as shown in Lemma 1.

**Lemma 1.** *Given two distributions $P$ and $Q$, the KL divergence can be rewritten with a new formula as*[3]*:*

$$\begin{aligned} &\mathbf{KL_c}\big(P||Q\big) \\ =&\frac{1}{\mathbb{E}(P)}\Big(\sum_C \big(P'(C)S(P',C)\mathbf{log}\frac{S(P',C)}{S(Q',C)}\big) \\ &+\sum_C \big(P'(C)S(P',C)\mathbf{log}\frac{P'(C)}{Q'(C)}\big)\Big) + \mathbf{log}\frac{\mathbb{E}(Q)}{\mathbb{E}(P)} \end{aligned} \tag{4}$$

*where $C$ is the cluster generated from the combined dataset, and*

$$\begin{aligned} &S(P',C) = \frac{|\mathbb{D}_P \cap C|}{|C|}, P'(c) = \frac{|\mathbb{D}_P \cap C|}{|\mathbb{D}_P \cup \mathbb{D}_Q|}, \\ &\mathbb{E}(P) = \frac{|\mathbb{D}_P|}{|\mathbb{D}_P \cup \mathbb{D}_Q|} \end{aligned} \tag{5}$$

*Likewise are the definitions of $S(Q',c)$, $Q'(c)$ and $\mathbb{E}(Q)$ (by replacing $\mathbb{D}_P$ with $\mathbb{D}_Q$ in the nominator).*

*Proof.* Define $P'(x=a) = \frac{|\{x|x=a \wedge x \in \mathbb{D}_P\}|}{|\{x|x \in \mathbb{D}_P \vee x \in \mathbb{D}_Q\}|}$, $\mathbb{E}(P) = \frac{|\mathbb{D}_P|}{|\mathbb{D}_P \vee \mathbb{D}_Q|}$. We then have

$$P(x) = P'(x)/\mathbb{E}(P)$$

$Q'(x=a)$ and $\mathbb{E}(Q)$ are defined in a similar way. Note that the first step of the proposed calculation is to perform clustering on the combined dataset. We expect a reasonable clustering approach can guarantee that the instances with the same value are assigned to the same cluster. In other words, $\{x|x=a \wedge x \in$

[3] To distinguish the new formula with the original formula of KL, we denote the new version as $\mathbf{KL_c}$ (Clustering-based KL). Their difference is explained after the proof.

$\mathbb{D}_P \wedge x \in c\} = \{x | x = a \wedge x \in \mathbb{D}_P\}$ where $c$ is a cluster and $x \in c$. This property can be valid for many clustering approaches, such as K-means. We then have

$$P'(x=a,c) = \frac{|\{x|x=a \wedge x \in \mathbb{D}_P \wedge x \in c\}|}{|\{x|x \in \mathbb{D}_P \vee x \in \mathbb{D}_Q\}|} = \frac{|\{x|x=a \wedge x \in \mathbb{D}_P\}|}{|\{x|x \in \mathbb{D}_P \vee x \in \mathbb{D}_Q\}|} = P'(x=a)$$

With these equations, the KL divergence in Eq (2) becomes:

$$\begin{aligned}
&\mathbf{KL_c}\big(P(x)||Q(x)\big) \\
=&\sum_x P(x)\mathbf{log}\frac{P(x)}{Q(x)} = \sum_x \frac{P'(x)}{\mathbb{E}(P)}\mathbf{log}\frac{P'(x)\mathbb{E}(Q)}{Q'(x)\mathbb{E}(P)} \\
=&\sum_x \frac{P'(x)}{\mathbb{E}(P)}\mathbf{log}\frac{P'(x)}{Q'(x)} \\
&+\mathbf{log}\frac{\mathbb{E}(Q)}{\mathbb{E}(P)}\sum_x \frac{P'(x)}{\mathbb{E}(P)}\sum_c\sum_{x\in c} P(x,c)\mathbf{log}\frac{P(x,c)}{Q(x,c)} \\
=&\frac{1}{\mathbb{E}(P)}\Big(\sum_c\sum_{x\in c}\big(P'(x|c)P'(c)\mathbf{log}\frac{P'(x|c)P'(c)}{Q'(x|c)Q'(c)}\big)\Big) + \mathbf{log}\frac{\mathbb{E}(Q)}{\mathbb{E}(P)} \\
=&\frac{1}{\mathbb{E}(P)}\Big(\sum_c\big(P'(c)\sum_{x\in c}P'(x|c)\mathbf{log}\frac{P'(x|c)}{Q'(x|c)}\big) \\
&+\sum_c\big(P'(c)\mathbf{log}\frac{P'(c)}{Q'(c)}\sum_{x\in c}P'(x|c)\big)\Big) + \mathbf{log}\frac{\mathbb{E}(Q)}{\mathbb{E}(P)}
\end{aligned}$$

Recall that the instances assigned to the same cluster are very similar to each other. We can then assume that in the same cluster, the expectation of instances from distribution P is the same as expectation of instances from Q:

$$\begin{aligned}
&\mathbb{E}_{x\in\mathbb{D}_P, x\in c}[x] = \mathbb{E}_{x\in\mathbb{D}_Q, x\in c}[x] \\
&\Rightarrow \sum_{x\in c} x\frac{P'(x|c)}{\sum\limits_{x\in c}P'(x|c)} = \sum_{x\in c} x\frac{Q'(x|c)}{\sum\limits_{x\in c}Q'(x|c)} \\
&\Rightarrow \sum_{x\in c} x\big(\frac{P'(x|c)}{\sum\limits_{x\in c}P'(x|c)} - \frac{Q'(x|c)}{\sum\limits_{x\in c}Q'(x|c)}\big) = 0
\end{aligned}$$

Note that this property can be guaranteed to be satisfied by applying clustering techniques such as bisecting k-means [18], with $|\mathbb{E}_{x\in P, x\in c}[x] - \mathbb{E}_{x\in Q, x\in c}[x]| < \theta$ as the termination condition, where $\theta$ set close to 0. In other words, if the condition does not satisfy in one of the clusters, a binary clustering procedure can be performed to divide the cluster smaller, until each cluster satisfies the condition. This process also adaptively decides the number of clusters. Since $x$ can take any value, to validate the above equation, we let

$$\frac{P'(x|c)}{\sum\limits_{x\in c}P'(x|c)} = \frac{Q'(x|c)}{\sum\limits_{x\in c}Q'(x|c)}$$

Eq (2) can then be rewritten as

$$
\begin{aligned}
&\mathbf{KL_c}\big(P(x)||Q(x)\big)\\
=&\frac{1}{\mathbb{E}(P)}\Big(\sum_c\big(P'(c)\sum_{x\in c}P'(x|c)\mathbf{log}\frac{\sum\limits_{x\in c}P'(x|c)}{\sum\limits_{x\in c}Q'(x|c)}\big)\\
&+\sum_c\big(P'(c)\mathbf{log}\frac{P'(c)}{Q'(c)}\sum_{x\in c}P'(x|c)\big)\Big)+\mathbf{log}\frac{\mathbb{E}(Q)}{\mathbb{E}(P)}\\
=&\frac{1}{\mathbb{E}(P)}\Big(\sum_c\big(P'(c)S(P',c)\mathbf{log}\frac{S(P',c)}{S(Q',c)}\big)\\
&+\sum_c\big(P'(c)S(P',c)\mathbf{log}\frac{P'(c)}{Q'(c)}\big)\Big)+\mathbf{log}\frac{\mathbb{E}(Q)}{\mathbb{E}(P)}
\end{aligned}
$$

□

The main difference between the original version of the KL divergence in Eq (2) and its new formula $\mathrm{KL_c}$ in Lemma 1 is that they are calculated in different ways in practice. The original version of KL in Eq (2) is usually calculated by Eq (3) because its formula requires to know every specific values of each variable $x$. However, with the new formula in Lemma 1, we can calculate the KL divergence by the clustering result on the whole dataset with several advantages. First, the clustering-based KL divergence in Lemma 1 can be computed efficiently and easily, because it only uses some basic statistics of the clustering. For example, $S(P', C)$ in Eq (5) represents the proportion of examples in the cluster $C$ originally generated from the distribution $P$. Second, we do not explicitly calculate the marginal distribution $P(x)$, which is normally difficult to approximate with a limited number of instances. Third, "high-level structures" (clusters) of the datasets are applied as a bridge to learn their differences, which are normally a more effective way to reflect the divergence. Other than the proof, we also empirically study the proposed version of KL in the experiment.

It is important to note that the KL divergence is asymmetric. In other words, $\mathbf{KL_c}(P_\mathcal{L}||P_\mathcal{U})$ is not necessarily equal to $\mathbf{KL_c}(P_\mathcal{U}||P_\mathcal{L})$, where $\mathcal{L}\sim P_\mathcal{L}$, $\mathcal{U}\sim P_\mathcal{U}$. However, we keep this property because we are only interested in the "risk" of learning the unlabeled data $\mathcal{U}$ based on the concept learnt from the labeled data $\mathcal{L}$. In other words, we use the risk of coding $\mathcal{U}$ based on the encoding from $\mathcal{L}$, as reflected by $\mathbf{KL_c}(P_\mathcal{U}||P_\mathcal{L})$. With Lemma 1, we define the "transfer risk" $\Re(\mathcal{L};\mathcal{U})$ in the logistic form, to regularize it into $[0, 1]$ and it is consistent with the known form of probability distribution:

$$\Re(\mathcal{L};\mathcal{U})=\big(1+\mathbf{exp}(\lambda-\mathbf{KL_c}(P_\mathcal{U}||\mathcal{P}_\mathcal{L}))\big)^{-1} \tag{6}$$

where $\mathcal{L}\sim P_\mathcal{L}$, $\mathcal{U}\sim P_\mathcal{U}$, and $\lambda=e^2$ is a deviation to make the minimum value of $\Re(\mathcal{L};\mathcal{U})$ close to 0. We then incorporate the transfer risk $\Re(\mathcal{L};\mathcal{U})$ into the proposed optimization function to automatically regulate the objectives to avoid negative transfer.

### 3.2 Objective Function

To generate a partition of the target data, the proposed algorithm finds an eigenspace where the target data can be clearly separated through a combination of two objectives. The first is to ensure the labeled data with the same class labels will be grouped together, and the second is to adapt the feature space to cater to the target data. Importantly, the transfer risk $\Re(\mathcal{L};\mathcal{U})$ (Section 3.1) automatically regulates the two goals.

Formally, the proposed optimization function, based on graph partition, can be written as

$$\min_{Y} \mathcal{J}(\mathcal{L},\mathcal{U}) = Cut(G_{\mathcal{L}\cup\mathcal{U}}, Y) + \beta\Big(\big(1-\Re(\mathcal{L};\mathcal{U})\big)T_{\mathcal{L}} + \Re(\mathcal{L};\mathcal{U})T_{\mathcal{U}}\Big) \tag{7}$$

where $Cut(G_{\mathcal{L}\cup\mathcal{U}}, Y)$ is a cost function of the partition $Y$ on a graph $G_{\mathcal{L}\cup\mathcal{U}}$ generated from the combined data $\mathcal{L}\cup\mathcal{U}$. Examples of such cost functions are Normalized Cut [19], MinMax Cut [6], and so on. Note that $T_{\mathcal{L}}$ and $T_{\mathcal{U}}$ are the two objectives formulated as partition constraints; $\beta$ is a parameter to control the overall effect of the constraints. On one hand, $T_{\mathcal{L}}$ is directly derived from the "must-link" constraint [21] to find a subspace where the instances are close to each other if they have the same class labels. On the other hand, $T_{\mathcal{U}}$ is a partition constraint defined on the pre-clustering result of $\mathcal{U}$ to reflect its natural separation. To construct $T_{\mathcal{U}}$, we first perform unsupervised spectral clustering on the target data $\mathcal{U}$ individually by Ncut [19]. The proposed algorithm then prefers to find a subspace to "gather" the instances closer if they are in the same pre-cluster. This constraint is defined to "reinforce" the natural manifold structure of $\mathcal{U}$ by maximizing its marginal separation.

We describe a partition constraint as $T_{\mathcal{L}}$ or $T_{\mathcal{U}}$ in Eq (7). To do this, we construct a constraint matrix $\mathbb{M}$ as follows:

$$\mathbb{M} = [\mathbf{m}_1, \mathbf{m}_2, ..., \mathbf{m}_r]^T \tag{8}$$

where each $\mathbf{m}_d$ is a $(s+t)\times 1$ matrix ($s+t$ is the total size of the combined dataset $\mathcal{L}\cup\mathcal{U}$). Each $\mathbf{m}_d$ represents a constraint on the dataset. For example, if $\mathbf{m}_1$ has an entry of $+1$ in the $i$th row, $-1$ in the $j$th row and the rest are all zero, it represents data $i$ and data $j$ are constrained to be close to each other. There are a total of $r$ constraints on the dataset. Then, let $\mathbb{M}_{\mathcal{L}}$ and $\mathbb{M}_{\mathcal{U}}$ denote the constraint matrix of $T_{\mathcal{L}}$ and $T_{\mathcal{U}}$, respectively. We have

$$\begin{aligned} T_{\mathcal{L}} &= \|\mathbb{M}_{\mathcal{L}}Y\|^2 \\ T_{\mathcal{U}} &= \|\mathbb{M}_{\mathcal{U}}Y\|^2 \end{aligned} \tag{9}$$

where $Y$ is the partition indicator. Now consider normalized cut [19] as the graph partition cost function, the proposed optimization function in Eq (7) becomes:

$$\min_{Y} \mathcal{J}(\mathcal{L},\mathcal{U}) = \frac{Y^T(D-W)Y}{Y^TDY} + \beta\Big(\big(1-\Re(\mathcal{L};\mathcal{U})\big)\|\mathbb{M}_{\mathcal{L}}Y\|^2 + \Re(\mathcal{L};\mathcal{U})\|\mathbb{M}_{\mathcal{U}}Y\|^2\Big) \tag{10}$$

where $Y$ is the partition indicator, $W$ is the similarity matrix of the combined dataset $\mathcal{L} \cup \mathcal{U}$, and $D = \mathbf{diag}(W \cdot e)$ ($e$ is a vector with all coordinates as 1). In Eq (10), the first term reflects the partition quality derived from normalized cut, and the second term consists of two constraints ($\|\mathbb{M}_{\mathcal{L}} Y\|^2$ and $\|\mathbb{M}_{\mathcal{U}} Y\|^2$), representing the two objectives; $\beta$ is the parameter to control the overall effect of the constraints. The transfer risk $\Re(\mathcal{L};\mathcal{U})$ serves as the pivotal component to balance the two constraints.

Then, if the transfer is too risky, or $\Re(\mathcal{L};\mathcal{U})$ is large, the effect of the first constraint decreases, and the optimization step prefers to satisfy the second constraint more in order to maintain the natural manifold structure of the target data and to avoid negative transfer. It is also important to emphasize that the parameter $\beta$ is not the essential component to avoid negative transfer. When negative transfer is likely to happen, the constraint will mainly come from the target data regulated by $\Re(\mathcal{L};\mathcal{U})$. In this case, the partition constraint does not include supervision from source data, and thus negative transfer is avoided regardless of the value of $\beta$. The effect of $\beta$ is also studied in the experiment.

### 3.3 Optimization

We introduce a key step to solve the proposed optimization function Eq (10). First, we denote

$$A = D - W + \beta\Big(\big(1 - \Re(\mathcal{L};\mathcal{U})\big)\mathbb{M}_{\mathcal{L}}^T\mathbb{M}_{\mathcal{L}} + \Re(\mathcal{L};\mathcal{U})\mathbb{M}_{\mathcal{U}}^T\mathbb{M}_{\mathcal{U}}\Big) \tag{11}$$

and $Z = D^{\frac{1}{2}}Y / \|D^{\frac{1}{2}}Y\|$. Then we have:

$$\begin{aligned}\mathcal{J}(\mathcal{L},\mathcal{U}) =& Z^T D^{-\frac{1}{2}}(D - W)D^{-\frac{1}{2}}Z \\ &+ \beta\Big(\big(1 - \Re(\mathcal{L};\mathcal{U})\big)\|\mathbb{M}_{\mathcal{L}} D^{-\frac{1}{2}}Z\|^2 \\ &+ \Re(\mathcal{L};\mathcal{U})\|\mathbb{M}_{\mathcal{U}} D^{-\frac{1}{2}}Z\|^2\Big) \\ =& Z^T D^{-\frac{1}{2}} A D^{-\frac{1}{2}} Z = \frac{Y^T D^{-\frac{1}{2}} A D^{-\frac{1}{2}} Y}{Y^T Y}\end{aligned} \tag{12}$$

It is easy to prove that $A$ is symmetric, because $D$, $W$, $\mathbb{M}_{\mathcal{L}}^T\mathbb{M}_{\mathcal{L}}$ and $\mathbb{M}_{\mathcal{U}}^T\mathbb{M}_{\mathcal{U}}$ are all symmetric while $A$ is a liner combination of these symmetric matrices. When we relax $Y$ to take the real values similar to other spectral clustering methods [10], we can use the $k$ smallest orthogonal eigenvectors of $D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$ to generate the partition.

The proposed algorithm is described in Fig 2. The algorithm first prepares the partition constraint matrices $\mathbb{M}_{\mathcal{L}}$ and $\mathbb{M}_{\mathcal{U}}$ as Eq (8), and the transfer risk $\Re(\mathcal{L};\mathcal{U})$ according to Eq (6). Moreover, we also construct the similarity matrix $W$ by a distance function like cosine distance, and then construct the corresponding diagonal matrix $D$. With these terms, we can get a matrix $A$ according to Eq (11). Then, we use the $k$ smallest eigenvectors of $D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$ to generate the eigenspace. Finally, we can perform clustering on the projected target data

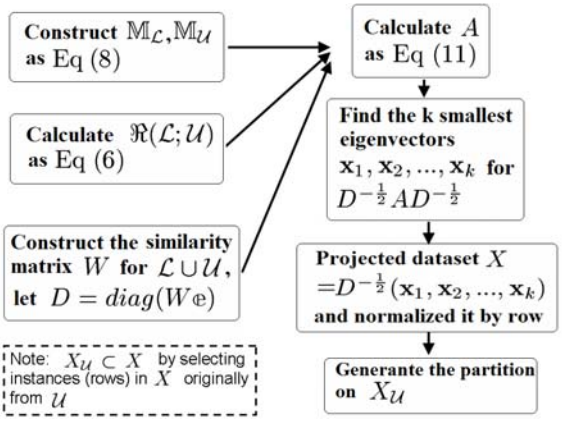



**Fig. 2.** Risk-sensitive Spectral Partition. Input: labeled dataset $\mathcal{L}$, target dataset $\mathcal{U}$ (number of clusters k); constraint parameter $\beta$.

$X_{\mathcal{U}}$. Note that although clustering is a straightforward approach to generate the partition, we can also use classifiers, such as KNN, to generate the partition if the class labels of source and target data are the same. The target data marked with the same class labels are assigned to the same cluster.

### 3.4 PAC Bound

The PAC-Bayes error bound in [14] is adopted with new terms to explain the behavior of the proposed model.

**Theorem 1.** *Let $P_{\mathcal{U}}$ and $P_{\mathcal{L}}$ be the distributions of target and source data respectively, and $s$ be the sample size of source data. Let $\delta \in (0, 1)$ be given. Then, with probability at least $1 - \delta$, the following bound holds,*

$$\varepsilon_{\mathcal{U}}(\mathcal{N}) \leq \varepsilon_{\mathcal{L}}(\mathcal{N}) + \sqrt{\frac{\mathbf{KL}(P_{\mathcal{U}}||P_{\mathcal{L}}) - ln\pi(\mathcal{N}) - \eta - ln\delta}{2s}}$$

*where $\mathcal{N}$ is the partition function, and $\pi(\mathcal{N})$ is a prior distribution of $\mathcal{N}$ that is usually based on domain knowledge, and $\eta$ is a normalization constant[14].*

In the proposed algorithm, we apply semi-supervised spectral clustering to generate the partition. The goal is to minimize the expected partition cost $\varepsilon_{\mathcal{U}}(\mathcal{N})$, similar to the expected error in classification. Like other PAC methods, we can minimize the empirical partition cost $\varepsilon_{\mathcal{L}}(\mathcal{N})$ on the given source data $\mathcal{L}$. In our case, we apply the "must-link" constraint to achieve this goal, by encouraging the labeled instances with the same class categories grouped together. However, observed from the second term of the right hand side, the bound also depends on the divergence of the two distributions $P_{\mathcal{U}}$ and $P_{\mathcal{L}}$. Thus, we apply the supervisory knowledge of source data $\mathcal{L}$ only when their divergence $\mathbf{KL}(P_{\mathcal{U}}||P_{\mathcal{L}})$

is small. Moreover, in order to distinguish the distribution divergence more effectively, we apply the clustering-based formula $\mathbf{KL_c}(P_{\mathcal{U}}||P_{\mathcal{L}})$ to calculate the KL divergence. With small partition cost $\varepsilon_{\mathcal{L}}$ on the labeled data, and small KL divergence, the strategy minimizes the upper bound of the expected cost $\varepsilon_{\mathcal{U}}(\mathcal{N})$.

## 4 Experiments

We empirically study the proposed method RSP (**R**isk-sensitive **S**pectral **P**artition) with two goals: (1) testing whether RSP can transfer across tasks having different class labels; (2) testing whether RSP can judiciously avoid negative transfer.

### 4.1 Experiment Setup

*Datasets*: We first conduct experiments on the text datasets as shown in Table 1. They are generated from 20-newsgroup and Reuters-21578 as in [5]. Each set of experiments contains one labeled source dataset and a corresponding target dataset. In addition, both the target dataset and the source dataset may come from different categories of documents, or even different document corpus. For example, the target dataset is from Reuters-21578, and the source dataset may be from 20-newsgroup. Thus, the class categories and data distributions of the two datasets may be significantly different. Each category contains around 1,500 documents. To speed up the optimization process, we first perform clustering on the target and source dataset respectively by Cluto [24] to generate 100 clusters each. We then choose the center of each cluster as the new data point. Finally, we label the whole cluster by its center.

**Table 1.** Text Datasets

| Target | $\text{Comp}_1$ VS. $\text{Rec}_1$ | Target | $\text{Org}_1$ VS. $\text{People}_1$ |
|---|---|---|---|
| Source | 2 classes: $\text{Comp}_2$ VS. $\text{Rec}_2$<br>4 classes: Graphics, Hardware, Politics.mid, Religion.misc<br>3 classes: Sci.crypt, Sci.med, Politics.guns | Source | 2 classes: $\text{Org}_2$, $\text{People}_2$<br>3 classes: $\text{Place}_2$, $\text{People}_2$, $\text{Org}_2$<br>3 classes: Sci.crypt, Sci.med, Politics.guns |

Note: $\text{Comp}_1$ and $\text{Comp}_2$ are different datasets with different distributions [5], likewise the other dataset with different superscripts.

**Table 2.** Image Datasets

| Target | Homer-simpson VS. Real-bear | Target | Cartman VS. Palmtree |
|---|---|---|---|
| Source | 2 classes: Superman, Teddybear<br>3 classes: Cartman, Palmtree, Teddybear<br>4 classes: Laptop, Pram, Keyboard, Soccer | Source | 2 classes: Superman, Bonsai<br>3 classes: Homer, Bonsai, Rear Bear<br>4 classes: Laptop, Pram, Keyboard, Umbrella |

In addition, we conduct experiments on image datasets in Table 2. Similar to the setting of text data, we also generate 6 sets of experiments, where each set contains one labeled dataset and one unlabeled target dataset. All the data are generated from the Caltech-256 image corpus [9] as shown in Table 1. Each category contains around 100 image instances.

*Baseline Methods:* To verify the effectiveness of the proposed model, an unsupervised spectral partition approach using normalized cut [19] is set as the first baseline method abbreviated as "No-T". This baseline directly generates the partition of the target data without transfer learning. Furthermore, we design another baseline method, abbreviated as "Full-T", by setting the transfer risk $\Re(\mathcal{L};\mathcal{U}) = 0$ in the optimization function in Eq (10) to fully apply the knowledge transferred from labeled data to learn the target data. This model does not include any strategy to avoid negative transfer. In the baseline methods and the proposed model RSP, the parameter $\beta$ (Eq 10) is set to be 0.6. The effect of this parameter is studied in another set of experiment.

*Evaluation Criteria:* Note that the outputs of the proposed model and the baseline methods are actually clusters. Thus, to compare the models, we define their accuracy by the purity of each cluster similar to [11]. The purity of a cluster $c$ can be defined as $\max P(y_i|c)$, and $P(y_i|c) = \frac{|\{x|x\in c, y(x)=y_i\}|}{|\{x|x\in c\}|}$, where $y_i$ is a class label, and $y(x)$ denotes the true label of $x$. The purity can be regarded as the accuracy when we label the whole cluster by its majority label. As a result, the accuracy is defined to be the weighted sum of the purity in all clusters; that is $\sum_c \frac{|\{x|x\in c\}|}{|x|} \max P(y_i|c)$.

### 4.2 Empirical Analysis

Tables 3 and 4 show the performance given by the baseline methods and the proposed model RSP in average accuracy on ten runs. We answer the following questions using three results:

(1) **Can RSP transfer knowledge across tasks having different class labels?** From the experimental result, RSP can achieve a higher accuracy than the strategy of "No-Transfer" especially when the source and target data are detected to be similar in distribution. For example, when the target dataset is "$\text{Org}_1$ VS. $\text{People}_1$" and the source dataset is a 3-classes document sets "$\text{Place}_2$, etc", the clustering-based formula of KL divergence is 0.51, implying that the target and source data are similar in distribution. In this case, RSP achieves an accuracy of 78% while the accuracy of "No-Transfer" is only 65%. It is clear that transfer learning helps improve the accuracy. More specifically, we plot Fig 3 to illustrate that the final eigenspace transferred from the 3-classes datasets also helps separate the binary target data.

(2) **Can the proposed model avoid negative transfer?** When the source dataset is the 3-classes document sets "Sci.crypt, etc" and the target dataset is "$\text{Comp}_1$ VS. $\text{Rec}_1$", the accuracy of full transfer (Full-T) is only 51%, close to random guessing. With the same setting, the accuracy of no transfer (No-T) is 74%. It is clear that negative transfer happens because the accuracy of the

**Table 3.** Experiment Result on Text Datasets

| Target | Source | $KL_t$ | $KL_c$ | Full-T | No-T | RSP |
|---|---|---|---|---|---|---|
| $Comp_1$ | $Comp_2$ VS. $Rec_2$ | 0.21 | 0.37 | 0.99 | 0.74 | **0.99**±0.00 |
| VS. | 4 classes: Graphics, etc | 0.01 | 1.17 | 0.94 | 0.74 | **0.98**±0.01 |
| $Rec_1$ | 3 classes: Sci.crypt, etc | 0.05 | **21.4** | 0.51 | 0.74 | **0.74**±0.03 |
| $Org_1$ | $Org_2$ VS. $People_2$ | 0.11 | 0.24 | 0.80 | 0.65 | **0.80**±0.00 |
| VS. | 3 classes: Places, etc | 0.05 | 0.51 | 0.73 | 0.65 | **0.78**±0.02 |
| $People_1$ | 3 classes: Sci.crypt, etc | 0.21 | **26.5** | 0.56 | 0.65 | **0.65**±0.06 |

Note: "$KL_t$" is the traditional calculation of KL by Eq (3); "$KL_c$" is the KL calculated by clustering according to Lemma 1. "Full-T" denotes the method applied transfer learning without considering the divergence between domains, while "No-T" denotes the traditional normalized cut without the strategy of transfer learning.

**Table 4.** Experiment Result on Image Datasets

| Target | Source | $KL_t$ | $KL_c$ | Full-T | No-T | RSP |
|---|---|---|---|---|---|---|
| Homer | Superman VS. Teddy | 0.62 | 0.17 | 0.85 | 0.72 | 0.85±0.02 |
| VS. | 3 classes: Cartman, etc | 0.88 | 0.29 | 0.81 | 0.72 | **0.81**±0.01 |
| Real-bear | 4 classes: Laptop, etc | 0.11 | **10.3** | 0.53 | 0.72 | 0.72±0.01 |
| Cartman | Superman VS. Bonsai | 0.12 | 0.07 | 0.87 | 0.55 | **0.87**±0.00 |
| VS. | 3 classes: Homer, etc | 0.43 | 0.55 | 0.92 | 0.55 | **0.92**±0.01 |
| Fern | 4 classes: Laptop, etc | 0.54 | 1.58 | 0.61 | 0.55 | **0.68**±0.01 |

transfer learning models are worse than no transfer. In the same situation, the proposed model RSP can judiciously avoid negative transfer and still obtains an accuracy of 74%.

(3) **How the proposed model avoids negative transfer?** In the above example, we observe that the KL divergence calculated by "Lemma 1" is 21.4, implying that the transfer is very risky according to Eq (6). In this case, the proposed model automatically decreases the effect of the first objective to avoid negative transfer. From the experimental result, it is also important to note that the clustering-based version $KL_c$ is a more effective KL to reflect distribution divergence. It is also one of the reasons the proposed model RSP outperforms the baseline models.

*Parameter Sensitivity:* We plot Fig 4 to study the effect of the parameter $\beta$ on the performance of the proposed model RSP (Fig 2). It is important to emphasize again that $\beta$ is not the essential component to avoid negative transfer. Instead, it is the transfer risk $\Re(\mathcal{L};\mathcal{U})$ that decides where the partition constraint comes from (from source data or target data). Thus, if negative transfer may happen, the partition constraint will mainly come from the target data regulated by the transfer risk $\Re(\mathcal{L};\mathcal{U})$, and negative transfer is avoided regardless of the value of $\beta$. In Fig 4, for each unlabeled target dataset in Table 1 and Table 2, the first source dataset is selected to report the result. The best performance appears at around $\beta = 0.6$. In real world practice, there are various ways to select the best value for $\beta$. For instance, partition cost functions, such as normalized cut [19], can be directly applied to evaluate the partition quality, by which one can choose the value of $\beta$ with the best performance.

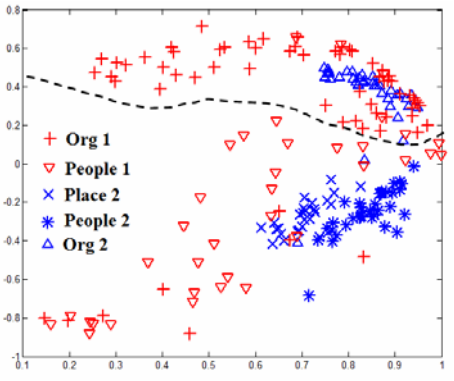



**Fig. 3.** Projection on the eigenspace

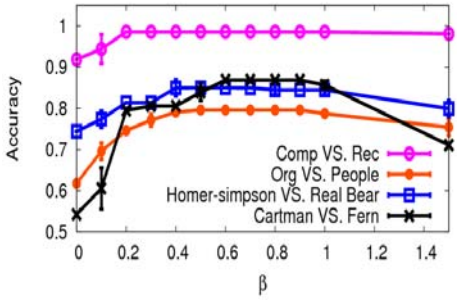



**Fig. 4.** Parameter Sensitivity of RSP

## 5 Related Work

*Spectral Method.* Various unsupervised spectral clustering algorithms have been proved effective in applications such as image segmentation (e.g., [6,12,19]), and the like. Moreover, several works about supervised spectral methods have been proposed to apply the labeled examples to help find the eigenspace of the target data drawn from the same or very similar distributions, such as [11,15,23]. Unlike most of these works, in this paper, we generate a partition of the unlabeled data by transferring knowledge from the given labeled data that may have very different distributions and class categories with the target data.

*Transfer Learning.* Transfer learning is proposed to extract knowledge from source data to help learn the target data. One of the main issues in transfer learning is how to transfer knowledge across different data distributions. A general approach is based on re-sampling (e.g., [3]), where the motivation of it is to "emphasize" the knowledge among "similar" and discriminating instances. Another line of work is to transfer knowledge based on the common features

found in a subspace (e.g., [5]) or a projected feature space where the different tasks are similar to each other (e.g., [2]). There are also some other solutions like model-combination based (e.g., [8]), transfer across similar learning parameters (e.g., [13]), and so on.

Different from these works, we mainly study the problem to transfer knowledge across tasks having different class labels. One important sub-issue of the problem is how to avoid negative transfer [17], which happens when the source data and the target data are significantly different. Previous works like [7,20] are proposed to solve negative transfer in the *supervised setting* where there are a few labeled examples in the target data. The general idea is to build a classifier with the labeled data from the target task, which is applied to identify the harmful knowledge by classification confidence or decrease of accuracy. However, in our problem to transfer knowledge over different class labels, we can not directly apply statics dependant on class labels (e.g., posterior) to select those harmful knowledge. Thus, different from these works, we solve the negative transfer problem in the *unsupervised setting* where the target data does not have any labeled examples at all.

## 6 Conclusions

We proposed a spectral partition based model to transfer knowledge across tasks having different class labels. The main framework is to find the optimal eigenspace to partition the target data by regulating two objectives. The first is to find the eigenspace where the source data of the same class labels will be close to each other, and the second is to maximize the marginal separation of the unlabeled target data. Importantly, a transfer risk term, as defined on the basis of an effective clustering-based KL divergence, is applied to regulate these two objectives to avoid negative transfer. These two objectives are formulated as partition constraints to construct a symmetric matrix, similar to graph Laplacian, to find the optimal solution given the objective function. The most important advantage of the proposed model is that it can automatically avoid negative transfer when the source data is very different from the target data, while still benefiting from transfer learning even when the source and target data have totally different class labels.

We evaluated the proposed model on text datasets and image datasets. For example, in one of the experiments, a 3-classes image dataset was used to supervise the partition of a binary-class dataset. Even though the two datasets have totally different class labels, the proposed method still achieved an accuracy of 81%, while the baseline model that does not apply transfer learning has accuracy of only 72%.

***Acknowledgement.*** We thank the anonymous reviewers for their greatly helpful comments. Qiang Yang thanks the support of Hong Kong CERG Project 621307. Jiangtao Ren is supported by the National Natural Science Foundation of China under Grant No. 60703110.

## References


1. Batu, T., Fortnow, L., Rubinfeld, R., Smith, W.D., White, P.: Testing that distributions are close. In: Proc. 41st FOCS (2000)
2. Ben-David, S., Blitzer, J., Crammer, K., Pereira, F.: Analysis of representations for domain adaptation. In: NIPS (2007)
3. Bickel, S., Bogojeska, J., Lengauer, T., Scheffer, T.: Multi-task learning for hiv therapy screening. In: ICML (2008)
4. Caruana, R.: Multitask learning. Machine Learning 28(1), 41–75 (1997)
5. Dai, W., Xue, G.R., Yang, Q., Yu, Y.: Co-clustering based classification for out-of-domain documents. In: KDD (2007)
6. Ding, C., He, X., Zha, H., Gu, M., Simon, H.: Spectral min-max cut for graph partitioning and data clustering. In: ICDM (2001)
7. Eaton, E., desJardins, M., Lane, T.: Modeling transfer relationships between learning tasks for improved inductive transfer. In: Daelemans, W., Goethals, B., Morik, K. (eds.) ECML PKDD 2008, Part I. LNCS (LNAI), vol. 5211, pp. 317–332. Springer, Heidelberg (2008)
8. Gao, J., Fan, W., Jiang, J., Han, J.: Knowledge transfer via multiple model local structure mapping. In: KDD (2008)
9. Griffin, G., Holub, A., Perona, P.: Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology (2007)
10. Golub, G., Loan, C.V.: Matrix Computation. The Johns Hopkins University Press, Baltimore (1996)
11. Ji, X., Xu, W., Zhu, S.: Document clustering with prior knowledge. In: SIGIR (2006)
12. Joachims, T.: Transductive learning via spectral graph partitioning. In: ICML (2003)
13. Lawrence, N.D., Platt, J.C.: Learning to learn with the informative vector machine. In: ICML (2004)
14. Li, X., Bilmes, J.: A divergence prior for adaptive learning. In: NIPS Workshop on Learning When Test and Training Inputs Have Different Distributions (2006)
15. Ling, X., Dai, W., Xue, G.R., Yang, Q., Yu, Y.: Spectral domain-transfer learning. In: KDD (2008)
16. Raina, R., Ng, A.Y., Koller, D.: Constructing informative priors using transfer learning. In: ICML (2006)
17. Rosenstein, M.T., Marx, Z., Kaelbling, L.P., Dietterich, T.G.: To transfer or not to transfer. In: NIPS (2005)
18. Savaresi, S.M., Boley, D.L.: On the performance of bisecting K-means and PDDP. In: SDM (2001)
19. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 22(8), 888–905 (2000)
20. Shi, X., Fan, W., Ren, J.: Actively transfer domain knowledge. In: Daelemans, W., Goethals, B., Morik, K. (eds.) ECML PKDD 2008, Part II. LNCS (LNAI), vol. 5212, pp. 342–357. Springer, Heidelberg (2008)
21. Wagstaff, K., Rogers, S., Schroedl, S.: Constrained k-means clustering with background knowledge. In: ICML (2001)
22. Wu, P., Dietterich, T.: Improving svm accuracy by training on auxiliary data sources. In: ICML (2004)
23. Yu, Stella X., Shi, Jianbo.: Grouping with Bias. newblock In: NIPS (2001)
24. Zhao, Y., Karypis, G.: Evaluation of hierarchical clustering algorithms for document datasets. In: ICKM (2002)