Self-taught Clustering

Wenyuan Dai[†] DWYAK@APEX.SJTU.EDU.CN
Qiang Yang[‡] QYANG@CSE.UST.HK
Gui-Rong Xue[†] GRXUE@APEX.SJTU.EDU.CN
Yong Yu[†] YYU@APEX.SJTU.EDU.CN
(†) Shanghai Jiao Tong University, 800 Dongchuan Road, Shanghai 200240, China
(‡) Hong Kong University of Science and Technology, Clearwater Bay, Kowloon, Hong Kong

Abstract

This paper focuses on a new clustering task, called *self-taught clustering*. Self-taught clustering is an instance of *unsupervised transfer learning*, which aims at clustering a small collection of target unlabeled data with the help of a large amount of *auxiliary* unlabeled data. The target and auxiliary data can be different in topic distribution. We show that even when the target data are not sufficient to allow effective learning of a high quality feature representation, it is possible to learn the useful features with the help of the auxiliary data on which the target data can be clustered effectively. We propose a co-clustering based self-taught clustering algorithm to tackle this problem, by clustering the target and auxiliary data simultaneously to allow the feature representation from the auxiliary data to influence the target data through a common set of features. Under the new data representation, clustering on the target data can be improved. Our experiments on image clustering show that our algorithm can greatly outperform several state-of-the-art clustering methods when utilizing irrelevant unlabeled auxiliary data.

1. Introduction

Clustering (Jain & Dubes, 1988) aims at partitioning objects into groups, so that the objects in the same groups are relatively similar, while the objects in different groups are relatively dissimilar. Clustering has a long history in machine learning (MacQueen, 1967), and recent works on clustering research have focused on improving the clustering performance using the prior knowledge in semi-supervised clustering (Wagstaff et al., 2001) and supervised clustering (Finley & Joachims, 2005).

In the past, semi-supervised clustering incorporates pairwise supervision, such as *must-link* or *cannot-link* constraints (Wagstaff et al., 2001), to bias clustering results. Supervised clustering methods learn distance functions from a small sample of auxiliary *labeled* data (Finley & Joachims, 2005). Different from these clustering problems, in this paper, we address a new clustering task where we use a large amount of *auxiliary unlabeled* data to enhance the clustering performance of a small amount of *target* unlabeled data. In our problem, we do not have any labeled data or pairwise supervisory constraint knowledge. All we have are the auxiliary data which are totally unlabeled and may be irrelevant to the target data. Our target data consist of a collection of unlabeled data from which it may be insufficient to learn a good feature representation. Thus, applying clustering directly on these target data may give very poor performance. However, with the help of auxiliary data, we are able to uncover a good feature set to enable high quality clustering on the target data.

Our problem can be considered as an instance of transfer learning, which makes use of knowledge gained from one learning task to improve the performance of another, even when these learning tasks or domains follow different distributions (Caruana, 1997). However, since all the data are unlabeled, we can consider it as an instance of *unsupervised transfer learning* (Teh et al., 2006). This unsupervised transfer learning problem could also be viewed as a clustering version of the self-taught learning (Raina et al., 2007), which uses irrelevant unlabeled data to help supervised learning. Thus, we refer to our problem as *self-taught clustering*

Appearing in *Proceedings of the 25th International Conference on Machine Learning*, Helsinki, Finland, 2008.

(a) diamond (b) platinum

(c) ring (d) titanium

Figure 1. Example for common features among different types of objects, using images as the instance.

(or STC for abbreviation).

To tackle the problem, we observe that the performance of clustering highly relies on data representation when the objective function and the distance measure are fixed. Therefore, to improve the clustering performance, one alternative way is to seek a better data representation. We observe that different objects may share some common or relevant features. For example, in Figure 1, `diamond` and `ring` share quite a lot of features about "diamond"; `ring` and `platinum` share quite a lot of features about "platinum"; moreover, `platinum` and `titanium` share quite a lot of features about "metal". In this situation, the auxiliary data can be used to help uncover a better data representation to benefit the target data set. Our approach to tackling this problem is by using co-clustering (Dhillon et al., 2003), so that the commonality can be found in the feature spaces that corresponds to similar semantic meanings.

In our solution to the self-taught clustering problem, two clustering operations, on the target data and the auxiliary data are respectively performed together. This is done through co-clustering. We extend the information theoretic co-clustering algorithm (Dhillon et al., 2003) which minimizes *loss in mutual information* before and after co-clustering. An iterative algorithm is proposed to monotonically reduce the objective function. The experimental results show that our algorithm can greatly improve the clustering performance by effectively using auxiliary unlabeled data, as compared to several other state-of-the-art clustering algorithms.

2. Problem Formulation

For clarity, we first define the self-taught clustering task. Let X and Y be two discrete random variables, taking values from two value sets $\{x_1, \ldots, x_n\}$ and $\{y_1, \ldots, y_m\}$, respectively. X and Y correspond to the target and auxiliary data. Let Z be a discrete random variable, taking values from the value set $\{z_1, \ldots, z_k\}$, that corresponds to the *common* feature space of both target and auxiliary data.

Let $p(X, Z)$ be the joint probability distribution with respect to X and Z, and $q(Y, Z)$ be the joint probability distribution with respect to Y and Z. In general, $p(X, Z)$ and $q(Y, Z)$ can be considered as two $n \times k$ and $m \times k$ matrices respectively, which can be estimated from data observations. For example, consider the case that $x_1 = \{z_1, z_3\}$, $x_2 = \{z_2\}$, and $x_3 = \{z_2, z_3\}$. Then, the joint probability distribution $p(X, Z)$ can be estimated as

$$p(X, Z) = \begin{bmatrix} 0.2 & 0.0 & 0.2 \\ 0.0 & 0.2 & 0.0 \\ 0.0 & 0.2 & 0.2 \end{bmatrix}. \quad (1)$$

We wish to cluster X into N partitions $\tilde{X} = \{\tilde{x}_1, \ldots, \tilde{x}_N\}$ and Y into M clusters $\tilde{Y} = \{\tilde{y}_1, \ldots, \tilde{y}_M\}$. Furthermore, Z can be clustered into K feature clusters $\tilde{Z} = \{\tilde{z}_1, \ldots, \tilde{z}_K\}$. We use $C_X : X \mapsto \tilde{X}$, $C_Y : Y \mapsto \tilde{Y}$ and $C_Z : Z \mapsto \tilde{Z}$ to denote three clustering functions, which map variables in the three value sets to their corresponding clusters. For brevity, in the following, we will use $\tilde{X}$, $\tilde{Y}$ and $\tilde{Z}$ to denote $C_X(X)$, $C_Y(Y)$ and $C_Z(Z)$, respectively.

Our objective is to find a good clustering function C_X for the target data, with the help of the clusters C_Y on the auxiliary data and C_Z on the common feature space.

3. The Self-taught Clustering Algorithm

In this section, we present our co-clustering based *self-taught clustering* (STC) algorithm, and then discuss its theoretical properties based on information theory.

3.1. Objective Function for Self-taught Clustering

We extend the information theoretic co-clustering (Dhillon et al., 2003) to model our self-taught clustering algorithm. In the information theoretic co-clustering, the objective function of co-clustering is defined as minimizing *loss in mutual information* between instances and features, before and after co-clustering. Formally, using the target data X and their feature space Z for illustration, the objective function can be expressed as

$$I(X, Z) - I(\tilde{X}, \tilde{Z}), \quad (2)$$

where $I(\cdot\,;\,\cdot)$ denotes the mutual information between two random variables (Cover & Thomas, 1991) that $I(X; Z) = \sum_{x \in X} \sum_{z \in Z} p(x, z) \log \frac{p(x,z)}{p(x)p(z)}$. Moreover, $I(\tilde{X}, \tilde{Z})$ corresponds to the joint probability distribu-

tion $p(\tilde{X}, \tilde{Z})$ which is defined as

$$p(\tilde{x}, \tilde{z}) = \sum_{x \in \tilde{x}} \sum_{z \in \tilde{z}} p(x, z). \quad (3)$$

For example, for the joint probability $p(X, Z)$ in Equation (1), suppose that the clustering on X is $\tilde{X} = \{\tilde{x}_1 = \{x_1, x_2\}, \tilde{x}_2 = \{x_3\}\}$, and the clustering on Z is $\tilde{Z} = \{\tilde{z}_1 = \{z_1, z_2\}, \tilde{z}_2 = \{z_3\}\}$. Then,

$$p(\tilde{X}, \tilde{Z}) = \begin{bmatrix} 0.4 & 0.2 \\ 0.2 & 0.2 \end{bmatrix}. \quad (4)$$

In this work, we model our self-taught clustering algorithm (STC) as performing co-clustering operations on the target data X and auxiliary data Y, simultaneously, while the two co-clusters share the same features clustering $\tilde{Z}$ on the feature set Z. Thus, the objective function can be formulated as

$$\mathcal{J} = I(X, Z) - I(\tilde{X}, \tilde{Z}) + \lambda \Big[I(Y, Z) - I(\tilde{Y}, \tilde{Z}) \Big]. \quad (5)$$

In Equation (2), $I(X, Z) - I(\tilde{X}, \tilde{Z})$ is computed on the co-clusters on the target data X, while $I(Y, Z) - I(\tilde{Y}, \tilde{Z})$ on the auxiliary data Y. λ is a trade-off parameter to balance the influence between the target data and the auxiliary data which we will test in our experiments. From Equation (5), we can see that, although the two co-clustering objective functions $I(X, Z) - I(\tilde{X}, \tilde{Z})$ and $I(Y, Z) - I(\tilde{Y}, \tilde{Z})$ are performed separately, they share the same feature clustering $\tilde{Z}$. This is the "bridge" to transfer the knowledge between the target and auxiliary data.

Our remaining task is to minimize the value of the objective function in Equation (5)[1]. However, minimizing Equation (5) is not an easy task, since it is non-convex and there are no good solutions currently to directly optimize this objective function. In the following, we will rewrite the objective function in Equation (5) into the form of Kullback-Leibler divergence (Cover & Thomas, 1991) (KL divergence), and minimize the reformulated objective function.

3.2. Optimization for Co-clustering

We first define two new probability distributions $\tilde{p}(X, Z)$ and $\tilde{q}(Y, Z)$ as follows.

Definition 1 *Let $\tilde{p}(X, Z)$ denote the joint probability distribution of X and Z with respect to the co-clusters (C_X, C_Z); formally,*

$$\tilde{p}(x, z) = p(\tilde{x}, \tilde{z}) \frac{p(x)}{p(\tilde{x})} \frac{p(z)}{p(\tilde{z})}, \quad (6)$$

where $x \in \tilde{x}$ and $z \in \tilde{z}$. Therefore, with regard to Equations (1) and (4), $\tilde{p}(X, Z)$ is given by

$$\tilde{p}(X, Z) = \begin{bmatrix} 0.089 & 0.178 & 0.133 \\ 0.044 & 0.089 & 0.067 \\ 0.067 & 0.133 & 0.200 \end{bmatrix}. \quad (7)$$

Likewise, let $\tilde{q}(Y, Z)$ denote the joint probability distribution of Y and Z with respect to the co-clusters (C_Y, C_Z). We have

$$\tilde{q}(y, z) = q(\tilde{y}, \tilde{z}) \frac{q(y)}{q(\tilde{y})} \frac{q(z)}{q(\tilde{z})}, \quad (8)$$

where $y \in \tilde{y}$ and $z \in \tilde{z}$.

Using the probability distributions $\tilde{p}(X, Z)$ and $\tilde{q}(Y, Z)$ defined above, we can reformulate the objective function in Equation (5) into a form based on KL divergence (Cover & Thomas, 1991).

Lemma 1 *When the clusters C_X, C_Y and C_Z are fixed, the objective function in Equation (5) can be reformulated as*

$$\begin{aligned} & I(X; Z) - I(\tilde{X}; \tilde{Z}) + \lambda \Big[I(Y; Z) - I(\tilde{Y}; \tilde{Z}) \Big] \quad (9) \\ & = D(p(X, Z) || \tilde{p}(X, Z)) + \lambda \, D(q(Y, Z) || \tilde{q}(Y, Z)), \end{aligned}$$

where $D(\cdot||\cdot)$ denotes the KL divergence between two probability distributions (Cover & Thomas, 1991), where $D(p||q) = \sum_x p(x) \log \frac{p(x)}{q(x)}$.

Proof Based on the Lemma 2.1 in (Dhillon et al., 2003), $I(X, Z) - I(\tilde{X}, \tilde{Z}) = D(p(X, Z)||\tilde{p}(X, Z))$. Similarly, $I(Y, Z) - I(\tilde{Y}, \tilde{Z}) = D(q(Y, Z)||\tilde{q}(Y, Z))$. Therefore, Lemma 1 can be proved straightforwardly. □

Lemma 1 converts the loss in mutual information to the KL divergence between the distributions p and $\tilde{p}$, and between q and $\tilde{q}$, respectively. However, the probability distributions in Lemma 1 are joint distributions, and are therefore difficult to optimize. Hence, in Lemma 2, we rewrite the objective function in Lemma 1 as a conditional probability form. We then show how to optimize the objective function in the new form.

Lemma 2 *The KL divergence with respect to joint probability distributions can be reformulated as*

$$\begin{aligned} & D(p(X, Z)||\tilde{p}(X, Z)) \\ & = \sum_{\tilde{x} \in \tilde{X}} \sum_{x \in \tilde{x}} p(x) D(p(Z|x)||\tilde{p}(Z|\tilde{x})) \quad (10) \\ & = \sum_{\tilde{z} \in \tilde{Z}} \sum_{z \in \tilde{z}} p(z) D(p(X|z)||\tilde{p}(X|\tilde{z})). \quad (11) \end{aligned}$$

[1] To be mentioned, in this paper, our minimization is for a fixed numbers of clusters N, M and K.

Similarly,

$$D(q(Y,Z)||\tilde{q}(Y,Z)) = \sum_{\tilde{y}\in\tilde{Y}}\sum_{y\in\tilde{y}} q(y)D(q(Z|y)||\tilde{q}(Z|\tilde{y})) \quad (12)$$

$$= \sum_{\tilde{z}\in\tilde{Z}}\sum_{z\in\tilde{z}} q(z)D(q(Y|z)||\tilde{q}(Y|\tilde{z})). \quad (13)$$

Proof We only give the proof to Equation (10). Using an identical argument, Equations (11), (12) and (13) can be easily derived.

$$D(p(X,Z)||\tilde{p}(X,Z)) = \sum_{\tilde{x}\in\tilde{X}}\sum_{\tilde{z}\in\tilde{Z}}\sum_{x\in\tilde{x}}\sum_{z\in\tilde{z}} p(x,z)\log\frac{p(x,z)}{\tilde{p}(x,z)}.$$

Since $\tilde{p}(x,z) = p(x)\frac{p(\tilde{x},\tilde{z})}{p(\tilde{x})}\frac{p(z)}{p(\tilde{z})} = p(x)\tilde{p}(z|\tilde{x})$, we have

$$\begin{aligned} &D(p(X,Z)||\tilde{p}(X,Z)) \\ &= \sum_{\tilde{x}\in\tilde{X}}\sum_{\tilde{z}\in\tilde{Z}}\sum_{x\in\tilde{x}}\sum_{z\in\tilde{z}} p(x)p(z|x)\log\frac{p(x)p(z|x)}{p(x)\tilde{p}(z|\tilde{x})} \\ &= \sum_{\tilde{x}\in\tilde{X}}\sum_{x\in\tilde{x}} p(x)\sum_{\tilde{z}\in\tilde{Z}}\sum_{z\in\tilde{z}} p(z|x)\log\frac{p(z|x)}{\tilde{p}(z|\tilde{x})} \\ &= \sum_{\tilde{x}\in\tilde{X}}\sum_{x\in\tilde{x}} p(x)D(p(Z|x)||\tilde{p}(Z|\tilde{x})). \end{aligned}$$

□

From Lemma 2 and Equation (10), we can see that minimizing $D(p(Z|x)||\tilde{p}(Z|\tilde{x}))$ for a single x can reduce the value of $D(p(X,Z)||\tilde{p}(X,Z))$ and thus can then decrease global optimization function in Equation (9). Therefore, if we iteratively choose the best cluster $\tilde{x}$ for each x to minimize $D(p(Z|x)||\tilde{p}(Z|\tilde{x}))$, the objective function will be minimized monotonically. Formally,

$$C_X(x) = \arg\min_{\tilde{x}\in\tilde{X}} D(p(Z|x)||\tilde{p}(Z|\tilde{x})). \quad (14)$$

Using a similar argument on Y and Z, we have

$$C_Y(y) = \arg\min_{\tilde{y}\in\tilde{Y}} D(q(Z|y)||\tilde{q}(Z|\tilde{y})), \quad (15)$$

and

$$C_Z(z) = \arg\min_{\tilde{z}\in\tilde{Z}} p(z)D(p(X|z)||\tilde{p}(X|\tilde{z})) + \lambda\, q(z)D(q(Y|z)||\tilde{q}(Y|\tilde{z})). \quad (16)$$

Based on Equation (14), (15) and (16), an alternative way to minimize the objective function in Equation (9) is derived, as shown in Algorithm 1.

In Algorithm 1, in each iteration, our self-taught clustering algorithm (STC) minimizes the objective function by choosing the best $\tilde{x}$, $\tilde{y}$ and $\tilde{z}$ for each x, y and z based on Equations (14), (15) and (16). As we discussed above, this can reduce the value of the global objective function in Equation (9). In the following theorem, we show the monotonically decreasing property of the objective function of the STC algorithm.

Algorithm 1 The Self-taught Clustering Algorithm: STC

Input: A target unlabeled data set X; an auxiliary unlabeled data set Y; the feature space Z shared by both X and Y; the initial clustering functions $C_X^{(0)}$, $C_Y^{(0)}$ and $C_Z^{(0)}$; the number of iterations T.

Output: The final clustering function $C_X^{(T)}$ on the target data X.

Procedure STC

1: Initialize $p(X,Z)$ and $q(Y,Z)$ based on the data observations on X, Y, and Z.
2: Initialize $\tilde{p}^{(0)}(X,Z)$ based on $p(X,Z)$, $C_X^{(0)}$, $C_Z^{(0)}$, and Equation (6).
3: Initialize $\tilde{q}^{(0)}(Y,Z)$ based on $q(Y,Z)$, $C_Y^{(0)}$, $C_Z^{(0)}$, and Equation (8).
4: **for** $t \leftarrow 1,\ldots,T$ **do**
5: Update $C_X^{(t)}(X)$ based on p, $\tilde{p}^{(t-1)}$, and Equation (14).
6: Update $C_Y^{(t)}(Y)$ based on q, $\tilde{q}^{(t-1)}$, and Equation (15).
7: Update $C_Z^{(t)}(Z)$ based on p, q, $\tilde{p}^{(t-1)}$, $\tilde{q}^{(t-1)}$, and Equation (16).
8: Update $\tilde{p}^{(t)}$ based on based on $p(X,Z)$, $C_X^{(t)}$, $C_Z^{(t)}$, and Equations (6).
9: Update $\tilde{q}^{(t)}$ based on based on $q(Y,Z)$, $C_Y^{(t)}$, $C_Z^{(t)}$, and Equations (8).
10: **end for**
11: Return $C_X^{(T)}$ as the final clustering function on the target data X.

Theorem 1 *In Algorithm 1, let the value of objective function $\mathcal{J}$ in the t-th iteration be*

$$\mathcal{J}(C_X^{(t)},C_Y^{(t)},C_Z^{(t)}) = D(p(X,Z)||\tilde{p}^{(t)}(X,Z)) + \lambda\, D(q(Y,Z)||\tilde{q}^{(t)}(Y,Z)). \quad (17)$$

Then,

$$\mathcal{J}(C_X^{(t)},C_Y^{(t)},C_Z^{(t)}) \geq \mathcal{J}(C_X^{(t+1)},C_Y^{(t+1)},C_Z^{(t+1)}). \quad (18)$$

Proof (Sketch) Since in each iteration, the clustering functions are updated based on Equations (14), (15) and (16), which locally minimize the values of $D(p(X,Z)||\tilde{p}(X,Z))$ and $D(q(Y,Z)||\tilde{q}(Y,Z))$, the objective function is monotonically non-increasing as a result. Theorem 1 follows as a consequence. □

Note that, although STC is able to minimize the objective function value in Equation (9), it is only able to find a locally optimal one. Finding the global optimal solution is NP-hard. The next corollary emphasizes the convergence property of our algorithm STC.

Corollary 1 *Algorithm 1 converges in a finite number of iterations.*

Proof (Sketch) The convergence of our algorithm STC can be proved straightforwardly based on the monotonical decreasing property in Theorem 1, and the finiteness of the solution space. □

3.3. Complexity Analysis

We now analyze the computational cost of our algorithm STC. Suppose that the total number of (x, z) co-occurrences in the target data set X is L_1, and the total number of (y, z) co-occurrences in the auxiliary data set Y is L_2. In each iteration, updating the target instance clustering C_X takes $O(N \cdot L_1)$. Updating the auxiliary instance clustering C_Y takes $O(M \cdot L_2)$. Moreover, updating the feature clustering C_Z takes $O(K \cdot (L_1 + L_2))$. Since the number of iterations is T, the time complexity of our algorithm is $O(T \cdot ((K + N) \cdot L_1 + (K + M) \cdot L_2)))$. In the following experiments, it is shown that $T = 10$ is enough for convergence. Usually, the number of clusters N, M and K can be considered as constants, so that the time complexity of STC is $O(L_1 + L_2)$.

Considering space complexity, our algorithm needs to store all the (x, z) and (y, z) co-occurrences and their corresponding probabilities. Thus, the space complexity is $O(L_1 + L_2)$. This indicates that the time complexity and the space complexity of our algorithm are all linear on the input. We conclude that the algorithm scales well.

4. Experiments

In this section, we evaluate our self-taught clustering algorithm STC on the image clustering tasks, and show effectiveness of STC.

4.1. Data Sets

We conduct our experiments on eight clustering tasks generated based on the Caltech-256 image corpus (Griffin et al., 2007). There are a total of 256 categories in the Caltech-256 data set, where we randomly chose 20 categories from this corpus. For each category, 70 images are randomly selected to form our clustering tasks. Six binary clustering tasks, one 3-way clustering task, and one 5-way clustering task were generated using these 20 categories, as shown in Table 1. The first column in Table 1 presents the categories with respect to the target unlabeled data. For each clustering task, we used the data from the corresponding categories as target unlabeled data, while the data from the remaining categories as the auxiliary unlabeled data.

For data preprocessing, we used the "bag-of-words" method (Li & Perona, 2005) to represent images in our experiments. Interesting points in images are found and described by SIFT descriptor (Lowe, 2004). Then, we clustered all the interesting points to get the codebook, and set the number of clusters to 800. Using this codebook, each image can be represented as a vector in the subsequent learning processes.

4.2. Evaluation Criteria

In these experiments, we used *entropy* to measure the quality of clustering results, which reveals the purity of clusters. Specifically, the entropy for a cluster $\tilde{x}$ is defined as $H(\tilde{x}) = -\sum_{c \in C} p(c|\tilde{x}) \log_2 p(c|\tilde{x})$, where c represents a category label in the evaluation corpus, and $p(c|\tilde{x})$ is defined as $p(c|\tilde{x}) = \frac{|\{x|\ell(x)=c \wedge x \in \tilde{x}\}|}{|\tilde{x}|}$, where $\ell(x)$ denotes the *true* label of x in the evaluation corpus. The *total entropy* for the whole clustering is defined as the weighted sum of the entropy with respect to all the clusters; formally, $H(\tilde{X}) = \sum_{\tilde{x} \in \tilde{X}} \frac{|\tilde{x}|}{n} H(\tilde{x})$. The quality of clustering $\tilde{X}$ is evaluated using the entropy $H(\tilde{X})$.

4.3. Empirical Analysis

We compared our algorithm STC to several state-of-the-art clustering methods as baseline methods. For each baseline method considered below, we have two different options: one is to apply the baseline method on the target data only, which we refer to as `separate`, and the other is to apply on the combined data consisting of target data and the auxiliary data, which we refer as `combined`. The first baseline method is a traditional 1D-clustering solution `CLTUO` (Zhao & Karypis, 2002) using its default parameter. The second baseline method is clustering on the target data under a new feature representation that is first constructed through feature clustering (on the target or the combined data set); this baseline is designed to evaluate the effectiveness of co-clustering based method as opposed to naively constructing new data representation for clustering. We refer to this class of baseline methods as `Feature Clustering`. The third baseline method is an information theoretic co-clustering method applied to the target (or the combined) data set (Dhillon et al., 2003), which we refer to as `Co-clustering`. This base-

Table 1. Performance in terms of entropy for each data set and evaluation method.

Data Set	CLUTO		Feature Clustering		Co-clustering		STC
	separate	combined	separate	combined	separate	combined	
eyeglass vs sheet-music	0.527	0.966	0.669	0.669	0.630	0.986	0.187
airplane vs ostrich	0.352	0.696	0.512	0.479	0.426	0.753	0.252
fern vs starfish	0.865	0.988	0.588	0.953	0.741	0.968	0.575
guitar vs laptop	0.923	0.965	0.999	0.970	0.925	1.000	0.569
hibiscus vs ketch	0.371	0.446	0.659	0.649	0.399	0.793	0.252
cake vs harp	0.882	0.879	0.998	0.911	0.860	0.996	0.772
car-side, tire, frog	1.337	1.385	1.362	1.413	1.316	1.275	1.000
cd, comet, vcr, diamond-ring, skyscaper	1.663	1.827	1.755	1.751	1.715	1.772	1.274
Average	0.865	1.019	0.943	0.974	0.877	1.068	0.610

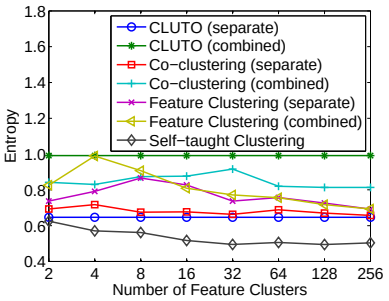

Figure 2. The entropy curves as a function of different number of feature clusters.

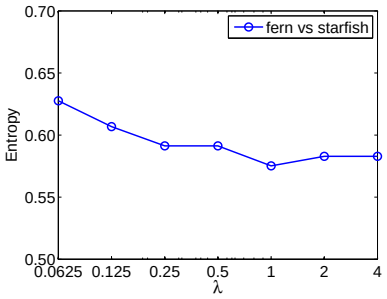

Figure 3. The entropy curves as a function of different trade-off parameter λ.

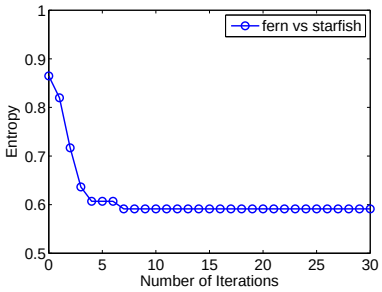

Figure 4. The entropy curves as a function of different number of iterations.

line is designed to test the effectiveness of our special co-clustering model for self-taught clustering.

Table 1 presents the clustering performance in entropy according to each data set and each evaluation method. From this table, we can see that `Feature Clustering` and `Co-clustering` perform somewhat worse than `CLUTO`. This is a little different from the results shown in the previous literatures such as (Dhillon et al., 2003). In our opinion, it is because our self-taught clustering problem focuses on a different situation from the previous ones; that is, the target data are insufficient for traditional clustering algorithms. In our experiments, there are only 70 instances in each category, which is too few to build a good feature clustering partition. Therefore, the performance of `Feature Clustering` and `Co-clustering` declines. Moreover, the performance with respect to `combined` is worse than that with respect to `separate` in general. We believe that it is because the target data and the auxiliary data are more or less independent of each other, and thus the topics in the combined data set may be biased towards the auxiliary data and thus harm the clustering performance on the target data. In general, our algorithm STC greatly outperforms the three baseline methods. We observe that the reason for the outstanding performance of STC is that the co-clustering part of STC makes feature clustering result consistent with the clustering result on both the target data and the auxiliary data. Therefore, using this feature clustering as the new data representation, the clustering performance of the target data is improved.

In our STC algorithm, it is assumed that we have already known the number of feature clusters K. However, in reality, this number should be carefully tuned. In these experiments, we tuned this parameter empirically. Figure 2 presents the entropy curves with respect to different number of feature clusters given by `CLUTO`, `Feature Clustering`, `Co-clustering` and STC respectively. The entropy in Figure 2 is the average over 6 binary image clustering tasks. Note that the curve given by `CLUTO` never changes, since `CLUTO` does not incorporate feature clustering. From this figure, we can see `Feature Clustering` and `Co-clustering` perform somewhat unstably as a function of the increasing number of feature clustering. We believe the reason is that there are only too few instances in each clustering task, which makes the traditional clustering results unreliable. Our algorithm STC incorporates a large amount of auxiliary unlabeled data, so that its variance is much smaller than that of traditional clustering algorithms. STC performs increasingly better in general, along with the increasing number of feature clustering, until the number of feature clusters reaches 32. When the number of feature clusters is greater than 32, the performance of STC becomes insensitive to the number of feature clusters. We believe a number of feature clustering which is no less than 32 will be sufficient to make STC perform well. In these experiments, we set the number of feature clustering

to 32.

We next tested the choice for the trade-off parameter λ in our algorithm STC (refer to Equation (5)). Generally, it is difficult to theoretically determine the value of the trade-off parameter λ. Instead, in this work, we tuned this parameter empirically on the data set `fern vs starfish`. Figure 3 presents the entropy curve given by STC along with changing trade-off parameter λ. From this figure, it can be seen that, when λ decreases, which implies that the weights of the auxiliary unlabeled data lower, the performance of STC declines rapidly. On the other hand, when λ is sufficiently large, i.e. $\lambda > 1$, the performance of STC is relatively insensitive to the parameter λ. This indicates the auxiliary data can help the clustering on the target data in our clustering tasks. In these experiments, we set the trade-off parameter λ to one, which is the best point in Figure 3.

Since our algorithm STC is iterative, the convergence property is also important to evaluate. Theorem 1 and Corollary 1 have already proven the convergence of STC theoretically. Here, we analyze the convergence of STC empirically. Figure 4 shows the entropy curve given by STC corresponding to different number of iterations on the data set `fern vs starfish`. From this figure, we can see that STC converges very well after 7 iterations, while the performance of STC reaches the lowest point when STC converges. This indicates that our algorithm STC converges very fast and very well. In these experiments, we set the number of iterations T to 10. We believe 10 iterations are enough for STC to converge.

5. Related Work

In this section, we review several past research works that are related to our work, including semi-supervised clustering, supervised clustering and transfer learning.

Semi-supervised clustering improves clustering performance by incorporating additional constraints provided by a few labeled data, in the form of *must-links* (two examples must in the same cluster) and *cannot-links* (two examples cannot in the same cluster) (Wagstaff et al., 2001). It finds a balance between satisfying the pairwise constraints and optimizing the original clustering criteria function. In addition to (Wagstaff et al., 2001), Basu et al. (2002) used a small amount of labeled data to generate initial seed clusters in K-means and constrained K-means algorithm by labeled data. Basu et al. (2004) generalized the previous semi-supervised clustering algorithms and proposed a probabilistic framework based on *hidden Markov random fields* that combines the constraints and clustering distortion measures in a general framework. Recent semi-supervised clustering works include (Nelson & Cohen, 2007; Davidson & Ravi, 2007).

Supervised clustering is another branch of work designed to improve clustering performance with the help of a collection of auxiliary labeled data. To address the supervised clustering problem, Finley and Joachims (2005) proposed an SVM-based supervised clustering algorithm by optimizing a variety of different clustering functions. Daumé III and Marcu (2005) developed a Bayesian framework for supervised clustering based on Dirichlet process prior.

Transfer learning emphasizes the transferring of knowledge across different domains or tasks. For example, multi-task learning (Caruana, 1997) or clustering (Teh et al., 2006) learns the common knowledge among different related tasks. Wu and Dietterich (2004) investigated methods for improving SVM classifiers with *auxiliary* training data sources. Raina et al. (2006) proposed to learn logistic regression classifiers by incorporating labeled data from irrelevant categories through constructing informative prior from the irrelevant labeled data. Raina et al. (2007) proposed a new learning strategy known as *self-taught learning*, which utilizes irrelevant unlabeled data to enhance the classification performance.

In this paper, we propose a new clustering framework called *self-taught clustering* which is an instance of *unsupervised transfer learning*. The basic idea is to use irrelevant unlabeled data to help the clustering of a small amount of target data. To our best knowledge, our self-taught clustering problem is novel in capturing a large class of machine learning problems.

6. Conclusions and Future Work

In this paper, we investigated an unsupervised transfer learning problem called *self-taught clustering*, and developed a solution by using an unlabeled auxiliary data to help improve the target clustering results. We proposed a co-clustering based self-taught clustering algorithm (STC) to solve this problem. In our algorithm, two co-clusterings are performed simultaneously on the target data and the auxiliary data to uncover the shared feature clusters. Our empirical results show that the auxiliary data can help the target data to construct a better feature clustering as data representation. Under the new data representation, the clustering performance on the target data is indeed enhanced, and our algorithm can greatly outperform several state-of-the-art clustering methods in the ex-

periments.

In this work, we tackled the self-taught clustering by finding a better feature representation using co-clustering. In the future, we will explore several other ways in finding common feature representations.

Acknowledgements

Qiang Yang thanks Hong Kong CERG grants 621307 and CAG grant HKBU1/05C. We thank the anonymous reviewers for their greatly helpful comments.

References

Bach, F. R., Lanckriet, G. R. G., & Jordan, M. I. (2004). Multiple kernel learning, conic duality, and the smo algorithm. *Proceedings of the Twenty-first International Conference on Machine Learning* (pp. 6–13).

Basu, S., Banerjee, A., & Mooney, R. J. (2002). Semi-supervised clustering by seeding. *Proceedings of the Nineteenth International Conference on Machine Learning* (pp. 27–34).

Basu, S., Bilenko, M., & Mooney, R. J. (2004). A probabilistic framework for semi-supervised clustering. *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 59–68).

Caruana, R. (1997). Multitask learning. *Machine Learning*, *28*, 41–75.

Cover, T. M., & Thomas, J. A. (1991). *Elements of information theory*. Wiley-Interscience.

Daumé III, H., & Marcu, D. (2005). A bayesian model for supervised clustering with the dirichlet process prior. *Journal of Machine Learning Research*, *6*, 1551–1577.

Davidson, I., & Ravi, S. S. (2007). Intractability and clustering with constraints. *Proceedings of the Twenty-fourth International Conference on Machine Learning* (pp. 201–208).

Dhillon, I. S., Mallela, S., & Modha, D. S. (2003). Information-theoretic co-clustering. *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 89–98).

Finley, T., & Joachims, T. (2005). Supervised clustering with support vector machines. *Proceedings of the Twenty-second International Conference on Machine Learning* (pp. 217–224).

Griffin, G., Holub, A., & Perona, P. (2007). *Caltech-256 object category dataset* (Technical Report 7694). California Institute of Technology.

Jain, A. J., & Dubes, R. C. (1988). *Algorithms for clustering data*. Englewood, NJ: Prentice-Hall.

Li, F.-F., & Perona, P. (2005). A bayesian hierarchical model for learning natural scene categories. *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2* (pp. 524–531).

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, *60*, 91–110.

MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. *Proceedings of Fifth Berkeley Symposium on Mathematical Statistics and Probability* (pp. 1:281–297).

Nelson, B., & Cohen, I. (2007). Revisiting probabilistic models for clustering with pair-wise constraints. *Proceedings of the Twenty-fourth International Conference on Machine Learning* (pp. 673–680).

Raina, R., Battle, A., Lee, H., Packer, B., & Ng, A. Y. (2007). Self-taught learning: transfer learning from unlabeled data. *Proceedings of the Twenty-fourth International Conference on Machine Learning* (pp. 759–766).

Raina, R., Ng, A. Y., & Koller, D. (2006). Constructing informative priors using transfer learning. *Proceedings of the Twenty-third International Conference on Machine Learning* (pp. 713–720).

Teh, Y. W., Jordan, M. I., Beal, M. J., & Blei, D. M. (2006). Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, *101*, 1566–1581.

Wagstaff, K., Cardie, C., Rogers, S., & Schrödl, S. (2001). Constrained k-means clustering with background knowledge. *Proceedings of the Eighteenth International Conference on Machine Learning* (pp. 577–584).

Wu, P., & Dietterich, T. G. (2004). Improving svm accuracy by training on auxiliary data sources. *Proceedings of the Twenty-first International Conference on Machine Learning* (pp. 110–117).

Zhao, Y., & Karypis, G. (2002). Evaluation of hierarchical clustering algorithms for document datasets. *Proceedings of the Eleventh International Conference on Information and Knowledge Management* (pp. 515–524).