Multi-Task Clustering using Constrained Symmetric Non-Negative Matrix Factorization

Samir Al-Stouhi* Chandan K. Reddy†

Abstract

Researchers have attempted to improve the quality of clustering solutions through various mechanisms. A promising new approach to improve clustering quality is to combine data from multiple related datasets (tasks) and apply multi-task clustering. In this paper, we present a novel framework that can simultaneously cluster multiple tasks through balanced Intra-Task (within-task) and Inter-Task (between-task) knowledge sharing. We propose an effective and flexible geometric affine transformation (contraction or expansion) of the distances between Inter-Task and Intra-Task instances. This transformation allows for an improved Intra-Task clustering without overwhelming the individual tasks with the bias accumulated from other tasks. A constrained low-rank decomposition of this multi-task transformation will allow us to maintain the class distribution of the clusters within each individual task. We impose an Intra-Task soft orthogonality constraint to a Symmetric Non-Negative Matrix Factorization (NMF) based formulation to generate basis vectors that are near orthogonal within each task. Inducing orthogonal basis vectors within each task imposes the prior knowledge that a task should have orthogonal (independent) clusters. Using several real-world experiments, we demonstrate that the proposed framework produces improves clustering quality compared to the state-of-the-art methods proposed in literature.

Keywords: Multi-Task Learning, Clustering, Non-Negative Matrix Factorization, Affinity Matrix.

1 Introduction

Several techniques have been proposed to improve the quality of a clustering solution [1]. A prominent new method to improve clustering quality is to simultaneously apply a clustering algorithm to a set of related datasets (tasks) in a *multi-task clustering* setting. Multi-task clustering aims to improve the clustering solution of each individual dataset using the knowledge gained from other related datasets in what is defined as Inter-Task knowledge sharing. While knowledge gained from related datasets can be a helpful source of auxiliary knowledge, this knowledge transfer can overwhelm each individual dataset and alter its distribution. For example, assuming we have 4 tasks with 200 samples within each task, each individual task will leverage 200 of its own samples and 600 samples from the related tasks and thus its distribution would be skewed by the overwhelming number of samples in the related tasks.

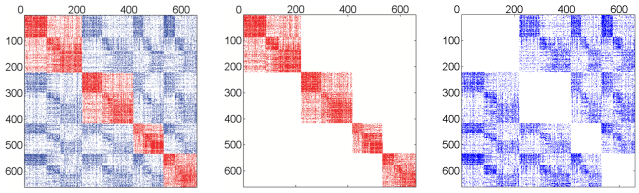
An "Affinity Matrix" is a symmetric non-negative similarity matrix that describes the distance (weight) between a set of instances. In this paper, we develop a novel framework for multi-task clustering where several tasks are clustered simultaneously with a mechanism to control the contribution of Intra-Task vs. Inter-Task knowledge within an affinity matrix. Controlling the bias between tasks can improve the clustering quality without overwhelming the individual tasks. Figure 1 shows a simple example to demonstrate the decomposition of an affinity matrix, with multiple tasks, into Intra-Task and Inter-Task components. We plot the affinity matrix for a four-task (200 samples per task) clustering problem where each task represents a different university and the instances represent different types of websites (personal or project) collected from each university. Figure 1(a) shows the full affinity matrix where all of the distances are treated equally and all tasks are combined into a single affinity matrix. The matrix in Figure 1(b) presents the Intra-Task components while Figure 1(c) present the Inter-Task components.

To incorporate a controlled bias into multi-task clustering, the Affinity Matrix is transformed to a Multi-Task Affinity Matrix where the weight, w, between two instances i and k can be biased (compressed or stretched) for an optimal clustering solution. Controlling this bias will be performed using general multi-task coefficients (λ) as follows:

$$w_{ik} = \begin{cases} \lambda_{\text{intra}} \cdot w_{ik} & \text{if } \langle \text{Intra} - \text{Task} \rangle \\ \lambda_{\text{inter}} \cdot w_{ik} & \text{if } \langle \text{Inter} - \text{Task} \rangle \end{cases} \tag{1.1}$$

*Corresponding Author. Department of Computer Engineering, Wayne State University, Detroit, MI. E-mail: s.alstouhi@.wayne.edu

†Department of Computer Science, Wayne State University, Detroit, MI. E-mail: reddy@cs.wayne.edu



(a) Full Task (b) Intra-Task (c) Inter-Task

Figure 1: Decomposition of a Multi-Task Affinity Matrix (with four tasks).

where $0 \leq \lambda \leq 1$ is a multi-task coefficient (or maybe a matrix of coefficients for exact tailoring) that can be modified for different clustering solutions.
Diminishing the Inter-Task connections can diminish the bias induced by other tasks but as the tasks become loosely connected, standard clustering methods will cut the weakly connected tasks into different clusters. For example, in the four-task example in Figure 1(b), standard clustering might group the first two tasks as one cluster and the last two tasks as another (or some other combination where each task belongs to only one class). To prevent this phenomenon, we obtain a low-rank decomposition that contains near orthogonal basis vectors within each task. Imposing orthogonal basis vectors within each task is analogous to forcing the basis vectors to contain two different clusters within each task and thus the cut is not between the tasks but rather within individual tasks. The main contributions of this paper are:

1. Introduce a novel Muti-Task Affinity Transformation that allows more flexibility in controlling (compress or stretch) the Inter-Task distances.

2. Develop a constrained symmetric non-negative matrix factorization to constrain the clustering solution of a multi-task affinity matrix to an Intra-Task solution.

3. Demonstrate the performance of the proposed framework using several real-world datasets and compare it with the state-of-the-art methods for standard and multi-task clustering.

The rest of the paper is organized as follows: Section 3 proposes a flexible and efficient construction method for a "Multi-Task Affinity Matrix" and section 4 presents an algorithm for generating relevant multi-task clustering solutions from the affinity transformation. Section 5 demonstrates our experimental results on several real-world datasets while section 6 concludes our work.

2 Related Work

We will primarily describe two groups of works that are related to our paper. The first one is in the area of *multi-task clustering* and the second one is in the area of *Non-negative Matrix Factorization*. Combining all tasks into one single clustering problem generally yields inferior results since multiple tasks, with different distributions, bias and distort each individual task's distribution. Multi-Task clustering methods aim to control the effect of the Inter-Task knowledge via regularization of the clustering objective or by finding a mapping (or view) where the multiple distributions share a common distribution in the mapped space. The multi-task clustering method proposed in [13] used Bregman divergence for task regularization to require the learned local mixture densities for all tasks to be similar. In [9], different tasks are mapped to a shared distribution in a Reproducing Kernel Hilbert Space (RKHS) where standard clustering can be performed in the common RKHS. Information theoretic clustering methods minimize the difference in mutual information between the original data matrix and that of the clustered random variables [5]. Self-Taught clustering [4] used the information theoretic approach for unsupervised transfer learning while the loss in mutual information was added for Inter-Task regularization for multi-task co-clustering in [11]. Multi-Task clustering was also applied via domain adaptation in [14]. Existing multi-task methods do not directly control the effect of the Inter-Task knowledge bias on individual tasks. Combining multiple tasks can overwhelm the Intra-Task distribution.
Non-negative Matrix Factorization (NMF) [8] is a matrix factorization technique with a non-negativity constraint that is beneficial for a parts representation of the data where the basis vectors are distributed and form sparse combinations that can generate expressiveness in the reconstructions. Given a non-negative data matrix X, non-negative matrix factorization is a linear, non-negative approximate data representation that aims to find two non-negative matrices U and V whose product can approximate the original matrix: $X \approx UV^T$. Various objective functions have been proposed [10] and the most widely used are the sum of squared error, Euclidean distance functions:

$$\min_{U,V \geqslant 0} \left\| X - UV^T \right\|^2 \tag{2.2}$$

Symmetric NMF is a special case of NMF decomposition where the basis U is replaced with V and the NMF optimization approximates a symmetric matrix W as: $W \approx VV^T$. Symmetric NMF can improve over standard NMF as it can discover clusters with a non-linear underlying structure [7]. There are several formulations for solving a Symmetric NMF problem [7, 6].

Symmetric NMF is also useful for clustering as it can be constrained to morph into several popular clustering methods. For example, for a square symmetric affinity matrix, W, Symmetric NMF can be equivalent to kernel k-means clustering with the additional constraints of orthogonality on V as follows:

$$\arg\min_{V\geq 0} \left\|W - VV^T\right\|^2 \quad s.t.\left(V^TV = I\right) \tag{2.3}$$

NMF can also be transformed to Normalized-Cut spectral clustering as:

$$\tilde{W} = D^{-\frac{1}{2}}WD^{-\frac{1}{2}}, D = diag\left(d_1, \ldots, d_m\right), d_i = \sum_j w_j \tag{2.4}$$

In this paper, we will be using a constrained version of symmetric NMF to solve our multi-task clustering formulation.

3 Multi-Task Affinity Transformation

3.1 Multi-Task Transformation Example Figure 2 illustrates how a multi-task affinity transformation translates to different clustering solutions where we present the simplest variations of λ_{intra} and λ_{inter} as we set $(\lambda_{\text{intra}} = 1)$ and $(\lambda_{\text{inter}} = \lambda)$. The goal is to cluster documents as either sports or science documents and each individual task has samples that belong to a branch of science (Chemistry, Biology) or sports (Basketball, Football). Intra-Task connections link documents via task-dependent (NBA, Avogadro) features and task-independent (Score, Celsius) features. On the other hand, Inter-Task instances only connect via task independent features. Following the multi-task definition in Equation (1.1), different λ values generate different clustering solutions as:

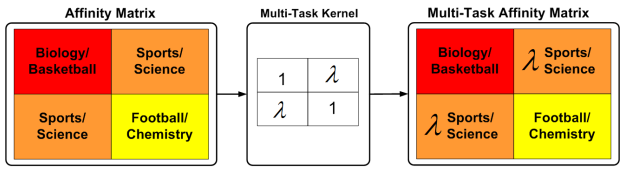


Figure 2: Multi-Task Affinity Transformation.

- *Intra-Task Clustering* $(\lambda = 0)$: This coefficient removes all Inter-Task connections and thus a task cannot share or get knowledge (or bias) from the other task. It should be noted that at $(\lambda = 0)$, standard clustering will generate two clusters where all the documents that belong to task one (Biology/Basketball) will form one cluster while all the documents the belong to task two (Football/Chemistry) will form the second cluster. For standard clustering, the tasks can be split into two affinity matrices and form two independent solutions.
- *Global-Task Clustering* $(\lambda = 1)$: All weights are biased equally as the multi-task coefficients are equal and the two tasks will combine into one global clustering solution.
- *Multi-Task Clustering* $(0 < \lambda < 1)$: This is the general definition of multi-task clustering where the clustering is an Intra-Task solution with an Inter-Task bias (or knowledge sharing).

A Multi-Task Affinity Matrix transformation can stretch the distance between tasks reducing the weighted connections between them. This is beneficial since Inter-Task knowledge should only contribute complementary (auxiliary) knowledge and not overwhelm the Intra-Task knowledge. The drawback of diminishing the connection between tasks is that standard clustering does not distinguish if the "cut" is between tasks or within tasks and given that a Multi-Task Affinity Matrix stretches the distance between tasks, clustering will "cut" between tasks and thus designate a task as a cluster (versus the correct solution where there is the prior knowledge that each task has more than one cluster). To prevent the Inter-Task "cut", a Non-negative Matrix Factorization method is proposed in section 4 where orthogonality in each task's basis vectors is promoted. Enforcing orthogonal basis vectors, within each task, is equivalent to forcing a solution with two different clusters within each task and thus the clustering "cut" is within each task and not between different tasks.

3.2 Objectives of Multi-Task Affinity Transformation To improve clustering we propose a general framework for multi-task clustering and aim to solve three problems:

1. *Efficiency*: Multi-task problems can grow rapidly in size as a result of the concatenation of features that form the combined set of all tasks. We aim to produce an efficient transformation from a standard Affinity Matrix to a Multi-Task Affinity Matrix.
2. *Flexibility*: In Section 3.1, it was demonstrated that modifying the multi-task coefficient (λ) generates different feature sets and clustering solutions. With variable feature sets, a general framework should be flexible to meet different clustering solutions and dataset feature variants.
3. *Relevance*: Modifying the task connections requires a special formulation to produce the desired clustering outcome. More details will be presented in Section 4.

3.3 Multi-Task Graph For a single formulation to meet the requirements outlined in Section 3.2, a star structured network will be constructed accommodating a variable number of tasks, exploiting the symmetry of the affinity matrix and allowing for heterogeneous (and different) set of features. Table 1 presents a summary of the notations used to create a "Multi-Task Affinity Matrix" that starts from a star structured network that is constructed with tasks $t = \{1, \ldots, T\}$.
Starting with an Input Data Matrix ($D^{n \times z}$) with n samples and z features, a network is constructed where:

1. Instances (samples) form nodes in an "Instance Task": $I^{n \times 1} = \{X_t\}_{t=1}^{T}$ and each task (t) has n^t samples for a total of $n = \sum_{t=1}^{T} n^t$ nodes.
2. Features form nodes in the "Feature Task" or the Zero$^{\text{th}}$ task: $Z^{1 \times z} = \{X_t\}_{t=0}$ for a total of z feature nodes.

With all instances and features mapped as nodes, an information graph $G = \langle M, E, W \rangle$ is constructed where the union of the "Instance Tasks" and the "Feature Task" forms the Multi-Task Graph, M:

$$M = \{I \cup Z\} = \{X_t\}_{t=0}^{T} \tag{3.5}$$

The binary relation between any two nodes i and j within this network is:

$$e_{ij} \in E \in \{0, 1\} \tag{3.6}$$

This network is weighted with the non-negative weights mapping feature nodes to instance nodes with $w \in \Re^{+}$ such that:

$$\forall e_{ij} = \left\langle x_t^j, x_t^i \right\rangle, \left\{ x_t^j \in X_{t=0} \wedge x_t^i \in X_{t \neq 0} \right\} \tag{3.7}$$

Equation (3.7) states that instance nodes only connect to feature nodes to form a bipartite graph. This graph is considered to be a bipartite graph since instance nodes and feature nodes can be divided into two disjoint sets ($t = 0$ and $t \neq 0$) such that every edge connects one vertex in ($t = 0$) to another vertex in ($t \neq 0$). This bipartite graph will be transformed into a Weighted Multi-Task Affinity Matrix (W).

3.4 Sub-Graph Matrices For a flexible mapping of the bipartite graph for the multi-task transformation, two types of sub-graphs will be constructed:

1. *Intra-Task sub-graphs*: Each sub-graph is a weighted graph connecting the Intra-Task instance nodes. For T tasks, a total of T sub-graphs are constructed.

Table 1: Summary of the notations used.

	Description
D	Input Data Matrix: $D^{n \times z}$ with n samples and z features
I	Instance Tasks: $I = \{X_t\}_{t=1}^{T}$
Z	Feature Task (Zero$^{\text{th}}$ Task): $Z = \{X_t\}_{t=0}$
M	Multi-Task Graph: $M = \{I \cup Z\}$
T	Total Number of Tasks
t	Task Index: $t = \{1, \ldots, T\}$
n^t	Number of Instances in the t^{th} Instance Task
n^f	Number of Features in the Feature Task
f^i	i^{th} feature
e_{ij}	Binary Relation between any two nodes i and j: $e_{ij} \in \{0, 1\}$
w_{ij}	Weighted Relation between any two nodes i and j: $w_{ij} \in \{^{+}\}$
E	Binary Relation Set
W	Weighted Multi-Task Affinity Matrix
x_t^i	Node i in Task t (Instance or Feature)
N	Total Number of Instances
l	Number of Labels
$V^{N \times 2}$	NMF Basis Vectors
$H^{T \times N}$	Task-Indicator Function
$K^{2 \times T}$	Class-Indicator Function

2. *Inter-Task sub-graphs*: Each sub-graph is a weighted graph connecting instance nodes from two different tasks using the feature nodes that are common to both tasks. A total of $T_{C_2} = \frac{T(T-1)}{2}$ sub-graphs are constructed.

A sub-graph is defined as G_{tt^*}, where t is the index of the first task and t^* is the index of the second task. $t \neq t^*$ for an Inter-Task graph while $t = t^*$ for an Intra-Task sub-graph.
Let us define $e_{x_t^i z^j}$ as the binary relation between the j^{th} feature node (z^j) and the i^{th} instance node (x^i) of task t. Let $e_{x_t^i z^j}$ indicate if the j^{th} feature node (z^j) is connected to the i^{th} instance node (x^i) of task t as:

$$e_{x_t^i z^j} \equiv \left(z^j \in x_t^i\right) = \begin{cases} 1 & w_{x_t^i z^j} \neq 0 \\ 0 & otherwise \end{cases} \tag{3.8}$$

To check if a feature node (z^j) belongs to any instance node (x_t) in task t, it has to connect to at least one of the task's instance nodes and thus we define the binary indicator $f_{z^j}^t$ to designate if the j^{th} feature node z^j belongs to task t as:

$$f_{z^j}^t \equiv \left(z^j \in t\right) = \begin{cases} 1 & \sum_{i=1}^{n^t} e_{x_t^i z^j} \neq 0 \\ 0 & otherwise \end{cases} \tag{3.9}$$

Now that the preliminaries have been defined, a sub-graph $G_{tt^*}\left[x_t^i x_{t^*}^k\right]$ is constructed as follows:

(3.10)
$$\sum_{j=1}^{z}\left(s_{x_t^i x_{t^*}^k}\right)\left(f_{z^j}^t f_{z^j}^{t^*}\right)\left(e_{x_t^i z^j} e_{x_{t^*}^k z^j}\right)\left(w_{x_t^i z^j}+w_{x_{t^*}^k z^j}\right)$$

Equation (3.10) is divided into four components where the first 3 components are Kronecker's Delta "checks" to generate weighted connections between instance nodes:

1. $\left(s_{x_t^i x_{t^*}^k}\right)$ is optional to prevent self-loops and is defined as:

$$\left[s_{x_t^i x_{t^*}^k}\right]=\begin{cases}1 & x_t^i \neq x_{t^*}^k\\ 0 & otherwise\end{cases} \tag{3.11}$$

2. $\left(f_{z^j}^t f_{z^j}^{t^*}\right)$: This section is an Inter-Task check and by definition of $f_{z^j}^t$, we will get a value of one if a feature node belongs to both tasks. It is included for efficiency as it eliminates all feature nodes that are not shared by the two tasks. For example, if two tasks only share 10% of the feature nodes, this check can eliminate 90% of the feature nodes for a reduced sub-graph. It is defined as:

$$\left[f_{z^j}^t f_{z^j}^{t^*}\right]=\begin{cases}1 & \left(z^j \in t\right)\wedge\left(z^j \in t^*\right)\\ 0 & \text{otherwise}\end{cases} \tag{3.12}$$

3. $(e_{x_t z_i} e_{x_{t^*} z_i})$: This section checks if a feature node (z_i) belongs to both instance nodes (x_{t^*}) and (x_t). The outcome is a value of one if that feature node belongs to both instance nodes.

$$\left[e_{x_t z_i} e_{x_{t^*} z_i}\right]=\begin{cases}1 & \left(z^j \in x_t^i\right)\wedge\left(z^j \in x_{t^*}^i\right)\\ 0 & \text{otherwise}\end{cases} \tag{3.13}$$

4. $\left(w_{x_t^i z^j}+w_{x_{t^*}^k z^j}\right)$: If a feature node belongs to both tasks and is connected to instance nodes (x_t and x_{t^*}), the weight of the path connecting these two instance nodes through the feature node (z_i) would be combined in the sub-graph.

3.5 Multi-Task Weighted Affinity Matrix For T tasks, a total of $T_{C_2}+T$ weighted sub-graphs are constructed and are combined to form a single Multi-Task Weighted Affinity Matrix with it's weight calculated as:

(3.14)
$$w_{x_t^i x_{t^*}^k}=\left[\left(\delta_{tt^*}\right)\lambda_{\text{intra}}+\left(\delta'_{tt^*}\right)\lambda_{\text{inter}}\right]G_{tt^*}\left[x_t^i x_{t^*}^k\right]$$

where (δ'_{tt^*}) is the inverse of the Kronecker's delta (δ_{tt^*}) which is defined as:

$$\delta_{tt^*}=\delta\left[tt^*\right]=\begin{cases}1 & t=t^*\\ 0 & t\neq t^*\end{cases} \tag{3.15}$$

Equation (3.14) can be broken down to:

$$w_{x_t^i x_{t^*}^k}=\begin{cases}\lambda_{\text{intra}}G_{tt^*}\left[x_t^i x_{t^*}^k\right] & t=t^*\\ \lambda_{\text{inter}}G_{tt^*}\left[x_t^i x_{t^*}^k\right] & t\neq t^*\end{cases} \tag{3.16}$$

This is the original definition of multi-task clustering given in Equation (1.1).

4 Symmetric Multi-Task NMF

4.1 Symmetric Multi-Task Non-Negative Matrix Factorization In this section, we modify a symmetric NMF objective function so that the clustering on a symmetric non-negative multi-task affinity matrix can generate local "Intra-Task" solutions while simultaneously incorporating the knowledge from all the tasks. Mathematically, given a Symmetric Multi-Task Affinity Matrix W, we want to find the basis vectors V such that:

(4.17)
$$\arg\min_{V\geq 0}\left[J\left(V\right)\right]=\arg\min_{V\geq 0}\left[\frac{1}{2}\left\|W-VV^T\right\|^2+\alpha Tr\left(\phi\phi^T\right)\right]$$

We add a multi-task sparsity/orthogonality constraint, ϕ, to the standard Symmetric-NMF formulation and define it as:

$$\phi=HVK \tag{4.18}$$

where $H^{T\times N}\in\{0,1\}$ is a task-indicator function. Within the trace penalty constraint, this matrix limits the orthogonality constraint to Intra-Task basis while excluding Inter-Task basis. It is defined as:

$$H\left(t,i\right)=\begin{cases}1 & i\in t\\ 0 & i\notin t\end{cases} \tag{4.19}$$

$K^{2\times T}\in\{-1,+1\}$ is the class-indicator function and sums the basis (if normalized) to zero when an Intra-Task solution is orthogonal. For a binary, two class, problem it is defined as:

$$K\left(i,t\right)=\begin{cases}+1 & i=1\\ -1 & i=2\end{cases} \tag{4.20}$$

This penalty encourages Intra-Task orthogonality which is analogous to enforcing that each task should have two independent basis. The penalty $Tr\left(\phi\phi^T\right)$ equals zero for a fully orthogonal within-task solution and is strictly increasing otherwise.

4.2 Multiplicative Update Rule To derive the updating rule for Equation (4.17) with non-negative constraints on V_{ij}, we introduce the Lagrangian multipliers λ to minimize the Lagrangian function: $L=$

$J + \sum_{ij} \lambda_{ij} V_{ij}$.

The first order KKT condition for local minima is:

$$\frac{\partial L}{\partial V_{ij}} = 0 \text{ and } \lambda_{ij} V_{ij} = 0, \forall i, j \tag{4.21}$$

Expanding the Lagrangian function L:

$$\begin{aligned} L &= \tfrac{1}{2}\left\|W - VV^T\right\|^2 + \alpha Tr\left(\phi\phi^T\right) + Tr\left(\lambda V^T\right) \\ &= Tr\left(\tfrac{1}{2}\left(W^TW - 2WVV^T + VV^TVV^T\right)\right) \\ &\quad + Tr\left(\alpha HVKK^TV^TH^T + \lambda V^T\right) \end{aligned} \tag{4.22}$$

The gradient of Equation (4.22) is:

$$\frac{\partial L}{\partial V} = -2WV + 2VV^TV + 2\alpha H^THVKK^T + \lambda \tag{4.23}$$

The KKT complementarity condition for the non-negativity of V_{ik} gives:

$$\left(-2WV + 2VV^TV + 2\alpha H^THVKK^T\right)_{ik} V_{ik} = 0 \tag{4.24}$$

This is the fixed point relation that the local minima for V must satisfy. To minimize Equation (4.17), we use the gradient descent method:

$$V_{ij} \leftarrow V_{ij} - \varepsilon_{ij} \frac{\partial J}{\partial V_{ij}} \tag{4.25}$$

Setting $\varepsilon_{ij} = \frac{V_{ij}}{4VV^TV}$, we derive the proposed updating rules of Equation (4.26).

$$V_{ij} = \frac{1}{2}\left[V_{ij}\left(1 + \frac{\left(WV - \alpha H^THVKK^T\right)_{ij}}{\left(VV^TV\right)_{ij}}\right)\right] \tag{4.26}$$

A value of α has to be set such that non-negativity is enforced $\forall_{ij}$:

$$\left[WV - \alpha\left(H^THVKK^T\right)\right]_{ij} \geq 0 \tag{4.27}$$

Since $KK^T \in \{\mathbb{R}_{<0}, \mathbb{R}_{>0}\}$, Equation (4.27) can be decomposed into its negative and positive components as:

$$\left[WV - \alpha\left(H^THVKK^T\right)^+ - \alpha\left(H^THVKK^T\right)^-\right]_{ij} \geq 0 \tag{4.28}$$

The term $\alpha\left(H^THVKK^T\right)^-$ can be dropped from Equation (4.28) since $\forall_{ij}$:

$$\left[-\alpha\left(H^THVKK^T\right)^-\right]_{ij} \geq 0 \tag{4.29}$$

Thus α has to be set to any value such that $\forall_{ij}$:

$$\alpha \leq \left(\frac{WV}{\left(H^THVKK^T\right)^+}\right)_{ij} \tag{4.30}$$

Simply stated, non-negativity is preserved if α is set to any positive value less than the minimum of the matrix calculated in Equation (4.30).

$$0 \leq \alpha \leq \min\left(\frac{WV}{\left(H^THVKK^T\right)^+}\right) \tag{4.31}$$

In our implementation, we preserved non-negativity and minimized $Tr\left(\phi\phi^T\right)$ by setting α to:

$$\alpha = \min\left(\frac{WV}{\left(H^THVKK^T\right)^+}\right) \tag{4.32}$$

4.3 Symmetric Multi-Task NMF Clustering Algorithm In this section, we present our Multi-Task clustering algorithm "Symmetric Multi-Task NMF". The first two steps generate a Multi-Task Affinity Matrix where the Inter-Task connection have their weights reduced by the Multi-Task coefficients λ[1]. We iterate to get the basis vectors and set the class membership to the basis vector with highest value.

Algorithm 1 Symmetric Multi-Task NMF (SMT-NMF)

Input: Input Data Matrix ($D^{n \times z}$). Multi-Task Coefficients λ

1: Construct Sub-Graph Matrices using equation (3.10).
2: Construct Weighted Multi-Task Affinity Matrix W using equation (3.14).
3: Set H using equation (4.19) and K using equation (4.20).
4: Initialize V with random non-negative values
5: **repeat**
6: $\quad \alpha = \min\left(\frac{WV}{\left(H^THVKK^T\right)^+}\right)$
7: $\quad V_{ij} = \frac{1}{2}\left[V_{ij}\left(1 + \frac{\left(WV - \alpha H^THVKK^T\right)_{ij}}{\left(VV^TV\right)_{ij}}\right)\right]$
8: **until** Convergence

Output: Assign i to cluster $j = \arg\max_j \left(V_{ij}\right), \forall i$.

5 Real-World Experimental Results

5.1 Experiment Setup In this section, we compare the proposed Symmetric Multi-Task NMF (SMT-NMF) clustering algorithm with single-task and all-task clustering methods including K-means, Normalized Cut (N-Cut) and standard Symmetric NMF ($\alpha = 0$). Additionally, we also compare with the recently proposed multi-task clustering algorithm "LNKMTC" [9].

[1] For simplicity we set: $\lambda_{\text{intra}} = 1$.

For N-Cut, we search for the best distance kernel and for LNKMTC we follow the setup described in [9] where the neighborhood size for the LNKMTC's lambda was uniform for all labels, the k-NN graph is set to $k = 10$, the regularization parameter C is set by searching the grid $\{0.1, 1, 10, 100, 500, 1000\}$ and b is set to 30. For SMT-NMF, we set $\lambda_{\text{inter}} = 1$ and search the multi-task coefficient $\lambda_{\text{intra}} = \{0, \ldots, 1\}$.
Since single task algorithms performed poorly when the number of samples is small, we varied the number of samples and compiled the results at four different sample sizes. Instances were picked randomly where the number of samples and the number of feature are outlined in Table 2. As the number of samples increased, so did the number of features (processed into TF-IDF representations [2]). At each sample size, we calculated the average clustering accuracy [12] of 30 runs and tabulated the total average from 120 runs.

5.2 Dataset Description The details of the datasets used in our experiments are summarized in Table 2.

- **20 Newsgroups**: The 20 Newsgroups[2] dataset is a collection of newsgroup documents. We generated six multi-task learning problems where each task is drawn from different sub-categories as outlined in [3]. For example, if the classes are from the top two categories: "Rec vs. Talk", the first task is from sub-categories \rec.sport.hockey and \talk.religions.misc whereas the second task is from \rec.sport.baseball and \talk.politics.mideast.
- **Reuters-21758:** The Reuters-21758[3] corpus contains Reuters news articles from 1987. Three multi-task problems with 2 tasks per problem were generated where the subcategory splits are analogous to the split in [3].
- **WebKB4**: The WebKB4[4] dataset contains web pages from four different universities (Cornell, Texas, Washington, Wisconsin) and thus 4 tasks were generated. Websites belong to either Personal (student/faculty) or Project (course/project).

5.3 Experimental Results on Real-World Multi-Task Datasets Different sets of experiments, (4-tasks, 2-classes) and (2-tasks, 2-classes), tested the ability of SMT-NMF to improve learning in a multi-task learning setting. We generated six 20newsgroups (4-tasks, 2-classes)[4], one WebKB4 (4-tasks, 2-classes) and three Reuters (2-tasks, 2-classes) experiments. The comparisons of clustering accuracy are presented in Tables (3-5) and they demonstrate that SMT-NMF consistently outperforms other algorithms. For the (4-tasks, 2-classes) datasets, the second best algorithms were (Single-Task NMF for 80% of the experiments) and (All-Task NMF for 20% of the experiments). For the (2-tasks, 2-classes) Reuters dataset, the second best algorithms were evenly split between All-Task NMF, Single-Task NMF, and Single-Task N-cut. "LNKMTC" did not perform well as the k-NN graph construction method creates sparse affinity matrices where a disjoint (or very weakly connected) set is formed when a set of instances in any one task only connects to the instances within their own task and does not connect with the remainder of the graph.

Table 2: Description of the datasets.

Dataset	Task	Tasks	Samples	Features
20News	rec-tlk	4	40-160	636-1963
	rec-sci	4	40-160	448-1876
	rec-comp	4	40-160	405-1468
	tlk-sci	4	40-160	631-2388
	tlk-comp	4	40-160	504-2066
	sci-comp	4	40-160	634-1939
WebKB	proj-pers	4	80-320	141-760
Reuters	orgs-ppl	2	40-160	1514-2552
	orgs-plcs	2	40-160	1501-2583
	ppl-plcs	2	40-160	1281-2610

5.4 Clustering with Different Number of Samples Figure 3 depicts the clustering performance across a variable number of instances. The WebKB4 and Reuters datasets are not plotted because SMT-NMF performed **significantly** better across all the sample sizes. For clarity, we plot the 20newsgroups dataset and **only** compare against the two most competitive algorithms (Single-Task NMF and Single-Task N-Cut). The sub-figures demonstrate that SMT-NMF improved the clustering accuracy where each individual task benefited from the additional sources of information. The performance of Single-Task algorithms eventually improved with increased availability of data with each individual task where each task eventually acquired a "sufficient" number of samples for Single-Task clustering to perform well without the need for Multi-Task learning.

6 Conclusion

We presented a novel multi-task clustering framework where the distances within and between tasks can be

[2] http://people.csail.mit.edu/jrennie/20Newsgroups/
[3] http://kdd.ics.uci.edu/databases/reuters21578/
[4] http://archive.ics.uci.edu/ml/

Table 3: Performance comparison of Clustering Accuracy (20Newsgroups(1-3)).

Four-Tasks, Two-Classes. (20 Newsgroups)												
DataSet	Rec vs Talk				Rec vs Sci				Rec vs Comp			
Method	T1	T2	T3	T4	T1	T2	T3	T4	T1	T2	T3	T4
S-Ncut	82.5	91.9	81.0	85.8	81.4	89.1	86.0	82.2	86.0	92.4	79.4	83.8
S-NMF	84.9	93.8	86.1	89.0	86.6	91.6	91.4	82.9	90.9	94.5	86.1	89.3
S-Kmeans	76.4	88.5	79.6	78.9	80.1	86.8	86.1	74.3	83.2	90.1	79.7	83.4
A-Ncut	82.4	86.0	61.3	64.0	81.0	81.3	60.5	60.7	84.8	86.5	72.4	75.1
A-NMF	83.7	91.9	74.9	76.1	84.3	85.2	75.6	70.2	90.7	94.9	80.7	83.3
A-Kmeans	81.8	89.1	64.3	64.7	81.5	87.7	55.4	54.3	87.2	94.0	71.3	71.0
LNKMTC	53.1	80.8	52.0	55.3	60.0	63.7	56.9	56.1	58.8	77.3	51.4	52.7
SMT-NMF	**93.2**	**96.7**	**89.6**	**91.2**	**91.2**	**94.9**	**91.8**	**88.7**	**95.7**	**97.5**	**92.0**	**92.8**

stretched (or compressed) to increase (or diminish) the knowledge-sharing and bias between tasks. The effectiveness of the framework was demonstrated and it was illustrated that it can address several multi-task clustering problems. A Symmetric Multi-Task Non-Negative Matrix Factorization method is developed where the NMF basis vectors are orthogonal within each task thus producing a clustering solution where the knowledge-sharing does not overwhelm or bias the individual tasks. The superiority of this new multi-task formulation was demonstrated with an extensive set of real-world multi-task clustering datasets.

Acknowledgements

This work was supported in part by the National Science Foundation grants IIS-1242304 and IIS-1231742.

References

[1] Aggarwal, C.C., Reddy, C.K. (eds.): Data Clustering: Algorithms and Applications. Chapman and Hall/CRC Press (2013)

[2] Aizawa, A.: An information-theoretic perspective of tf–idf measures. Information Processing & Management 39(1), 45–65 (2003)

[3] Dai, W., Xue, G.R., Yang, Q., Yu, Y.: Co-clustering based classification for out-of-domain documents. In: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 210–219 (2007)

[4] Dai, W., Yang, Q., Xue, G.R., Yu, Y.: Self-taught clustering. In: Proceedings of the 25th International Conference on Machine Learning. pp. 200–207. ACM (2008)

[5] Dhillon, I.S., Mallela, S., Modha, D.S.: Information-theoretic co-clustering. In: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 89–98. KDD (2003)

[6] He, Z., Xie, S., Zdunek, R., Zhou, G., Cichocki, A.: Symmetric nonnegative matrix factorization: Algorithms and applications to probabilistic clustering. Neural Networks, IEEE Transactions on 22(12), 2117–2131 (2011)

[7] Kuang, D., Ding, C., Park, H.: Symmetric nonnegative matrix factorization for graph clustering. In: Proceedings of 2012 SIAM International Conference on Data Mining. pp. 106–117 (2012)

[8] Lee, D.D., Seung, H.S.: Learning the parts of objects by non-negative matrix factorization. Nature 401(6755), 788–791 (Oct 1999)

[9] Quanquan, G., Zhenhui, L., Han, J.: Learning a kernel for multi-task clustering. In: Proceedings of the 25th Conference on Artificial Intelligence (AAAI). pp. 000–000. AAAI '11 (2011)

[10] Seung, D., Lee, L.: Algorithms for non-negative matrix factorization. Advances in neural information processing systems 13, 556–562 (2001)

[11] Xie, S., Lu, H., He, Y.: Multi-task co-clustering via nonnegative matrix factorization. In: Pattern Recognition (ICPR), 2012 21st International Conference on. pp. 2954–2958. IEEE (2012)

[12] Xu, W., Liu, X., Gong, Y.: Document clustering based on non-negative matrix factorization. In: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval. pp. 267–273. ACM (2003)

[13] Zhang, J., Zhang, C.: Multitask bregman clustering. In: Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence. pp. 655–660 (2010)

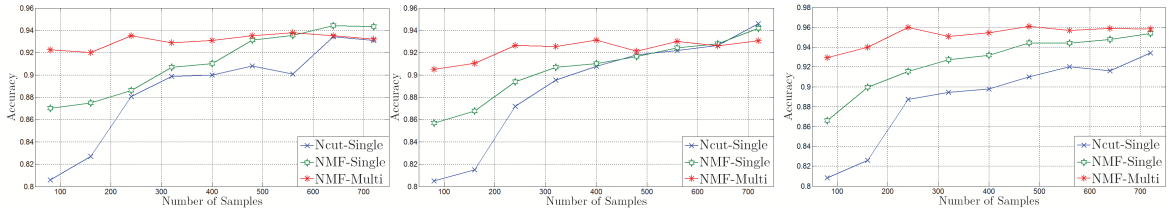
[14] Zhang, Z., Zhou, J.: Multi-task clustering via domain adaptation. Pattern Recognition 45(1), 465 – 473 (2012)

Table 4: Performance comparison of Clustering Accuracy (20Newsgroups(4-6)).

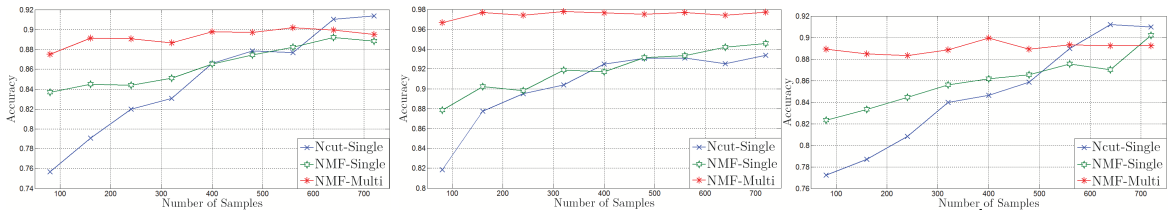
Four-Tasks, Two-Classes. (20 Newsgroups)												
DataSet	Talk vs Sci				Talk vs Comp				Sci vs Comp			
Method	T1	T2	T23	T4	T1	T2	T3	T4	T1	T2	T3	T4
S-Ncut	75.7	84.5	83.9	77.2	84.3	81.8	94.0	89.3	70.1	84.4	84.9	82.8
S-NMF	80.6	88.4	87.5	81.0	85.4	90.1	92.2	92.0	75.8	88.4	89.6	82.0
S-Kmeans	71.6	80.4	78.2	70.6	79.3	81.7	88.6	85.9	68.7	80.8	82.8	73.4
A-Ncut	71.9	68.6	72.3	66.1	86.3	87.2	91.4	89.2	66.1	80.2	79.0	72.4
A-NMF	76.3	76.7	80.7	71.7	91.6	92.5	95.9	94.1	65.9	82.7	84.7	78.7
A-Kmeans	70.3	73.1	78.2	66.9	91.5	91.2	96.0	93.6	65.0	79.5	81.1	74.7
LNKMTC	54.5	54.1	68.1	56.8	51.8	54.5	80.1	55.0	52.4	62.2	60.5	61.8
SMT-NMF	**89.3**	**88.5**	**91.9**	**84.7**	**97.0**	**97.2**	**97.9**	**97.4**	**78.2**	**91.8**	**93.6**	**91.0**

Table 5: Performance comparison of Clustering Accuracy (WebKB4, Reuters).

Four-Tasks,Two-Classes(WebKB4). Two-Tasks,Two-Classes(Reuters)										
	WebKB4					Reuters				
DataSet	Project vs Personal					ppl-orgs		plcs-orgs		ppl-plcs	
Method	T1	T2	T3	T24		T1	T2	T1	T12	T1	T2
S-Ncut	77.3	72.5	78.1	65.8		70.9	74.0	74.3	69.1	66.7	69.7
S-NMF	84.6	83.0	84.9	86.6		72.3	65.3	62.1	71.4	69.8	61.1
S-Kmeans	76.2	72.1	80.0	60.5		55.6	57.0	59.2	58.8	57.0	57.5
A-Ncut	69.8	67.9	67.2	70.4		71.6	73.9	72.9	66.5	61.7	65.6
A-NMF	85.9	82.9	81.3	86.5		73.1	75.4	74.0	68.8	63.4	69.3
A-Kmeans	67.8	67.1	67.1	67.0		57.6	57.3	54.6	58.1	59.5	56.5
LNKMTC	61.7	52.9	52.0	63.3		59.4	62.8	62.8	59.1	56.7	60.0
SMT-NMF	**92.1**	**88.3**	**88.0**	**92.6**		**81.0**	**84.6**	**81.3**	**77.6**	**73.7**	**75.1**



(a) Rec vs. Talk (b) Rec vs. Sci (c) Rec vs. Comp



(d) Talk vs. Sci (e) Talk vs. Comp (f) Sci vs. Comp

Figure 3: Performance comparison against best competing algorithms across varying number of instances.