# Rethinking Unsupervised Feature Selection: From Pseudo Labels to Pseudo Must-links

Xiaokai Wei[1] ⋆, Sihong Xie[2], Bokai Cao[3] and Philip S. Yu[3]
[1] Facebook Inc., Menlo Park, CA, USA
[2] Department of Computer Science, U of Illinois at Chicago, Chicago, IL, USA
[3] CSE Department, Lehigh University, Bethlehem, PA, USA
weixiaokai@gmail.com, sxie@cse.lehigh.edu, {caobokai, psyu}@uic.edu

**Abstract.** High-dimensional data are prevalent in various machine learning applications. Feature selection is a useful technique for alleviating the curse of dimensionality. Unsupervised feature selection problem tends to be more challenging than its supervised counterpart due to the lack of class labels. State-of-the-art approaches usually use the concept of pseudo labels to select discriminative features by their regression coefficients and the pseudo-labels derived from clustering is usually inaccurate. In this paper, we propose a new perspective for unsupervised feature selection by Discriminatively Exploiting Similarity (DES). Through forming similar and dissimilar data pairs, implicit discriminative information can be exploited. The similar/dissimilar relationship of data pairs can be used as guidance for feature selection. Based on this idea, we propose hypothesis testing based and classification based methods as instantiations of the DES framework. We evaluate the proposed approaches extensively using six real-world datasets. Experimental results demonstrate that our approaches achieve significantly outperforms the state-of-the-art unsupervised methods. More surprisingly, our unsupervised method even achieves performance comparable to a supervised feature selection method.

**Keywords:** Feature selection

## 1 Introduction

Feature selection [5] [7] [10] [20] [22], as a dimension reduction technique, can help improve the performance of machine learning tasks [14], by selecting a subset of features. Besides, it can also enhance the efficiency of subsequent learning process, and provide easier interpretation of the problem.

Depending on the availability of supervision information, feature selection methods can be categorized into two classes: Supervised feature selection and unsupervised feature selection. For supervised feature selection, the criterion on good features is more straightforward: good features should be highly correlated with class labels, such as Fisher Score [3] and HSIC [13]. Without guidance from class labels, it is difficult to evaluate the discriminativeness of features. Different heuristics (e.g., frequency

⋆ The work was performed when the first author was a Ph.D. student at University of Illinois at Chicago.

based, variance based) have been proposed to perform unsupervised feature selection. Similarity-preserving approaches [5] [24] have gained much popularity among others. In such similarity preserving framework, a feature is considered to be of good quality if it can preserve the local manifold structure well.

However, such simple heuristics do not necessarily lead to discriminative features. Recently, pseudo label based algorithms [8] [22] have been developed. Since class labels are not available, such methods attempt to generate pseudo labels via certain clustering methods. They select features based on their utility to predicting pseudo labels. One major drawback of such approach is that the pseudo labels are usually far from accurate (accuracy is usually about $30\%$ - $70\%$ as reported in previous work [8] [11]). So such inaccurate pseudo labels can mislead feature selection. Also, in unsupervised explorative analysis, the number of classes is often not known as apriori.

The central issue in unsupervised feature selection is how to effectively uncover the discriminative information embedded in the data. Is the concept of pseudo-label the only and best way to achieving discriminativeness? In this paper, we present a novel perspective, **pseudo must-link**, and show how it can be a better alternative than pseudo-labels. We refer to the proposed framework as DES (Discriminatively Exploiting Similarity), which performs unsupervised feature selection based on similarity in a discriminative manner. Specifically, DES aims to exploit the key intuition that similar data points are more likely to be of the same class than two random data points. We present a pair-wise formulation to effectively utilize the such difference between similar and dissimilar data points.

By regarding similar pair/dissimilar pair as two classes, the rich arsenal of supervised feature selection approaches can be employed. We propose two instantiations of this framework for unsupervised feature selection: *Hypothesis Testing* based HT-DES and *Classification* based CL-DES. The proposed approaches are conceptually simple, easy to implement but highly effective. Experimental results shows that DES can achieve significantly better performance than state-of-the-art unsupervised methods. Besides, the performance of CL-DES is even comparable to that of supervised mutual information-based method on most datasets. To our best knowledge, this is the first time an unsupervised method reportedly achieves comparable performance as a classic supervised feature selection method, which illustrates the strength of the proposed framework.

## 2 Related Work

In this section, we review related work on feature selection.

Feature selection has attracted considerable amount of attention in the research community. The goal is to alleviate the curse of dimensionality, enabling machine learning model to achieve comparable, if not better, performance. In supervised feature selection, the criterion for good features is more straightforward: good features should be highly correlated with class labels. Many methods have been proposed to capture the correlation between label and feature, such as Mutual Information, Fisher Score [3] and Hilbert-Schmidt Independence Criterion (HSIC) [13] and LASSO [16].

In the unsupervised setting, heuristic-based feature selection algorithms tend to evaluate the importance of features individually [5] [23], which has a limitation of neglecting correlation among features. Recent methods [11] [12] [22] overcome this issue by evaluating feature utility with $L_{2,1}$ norm-based sparse regression, which are typically in the following form:

$$\begin{aligned} \min_{\mathbf{W},\mathbf{U}} \ & l(\mathbf{XW}-\mathbf{U}) + \sum_{i=1}^{R} \alpha_i \cdot reg_i(\mathbf{U}) + \lambda ||\mathbf{W}||_{2,1} \\ s.t. \ & Constraint(\mathbf{U}) \end{aligned}$$

where $\mathbf{X}$ is the feature matrix and $\mathbf{U}$ is cluster indicators/latent factors. The $L_{2,1}$ norm $||\mathbf{W}||_{2,1}$ promotes row sparsity in the coefficient matrix $\mathbf{W}$ and hence achieves the effect the feature selection. Different sparse regression-based methods usually differ in $l(\cdot)$, $reg(\cdot)$ and $Constraint(\mathbf{U})$ they use. A typical choice for $l(\cdot)$ is Frobenius norm, such as in UDFS [22], NDFS [8] and FSASL [2], while RUFS [11] and RSFS [12] employ robust $L_{2,1}$ loss and Huber Loss as $l(\cdot)$, respectively. Concerning $reg(\mathbf{U})$, different choices can also be made. For example, UDFS, NDFS and RSFS use local structure-based regularizations in the objective function. RUFS further adds NMF (Non-negative Matrix Factorization) based regularization and FSASL adds sparse regression [21] based regularization. Most methods also put certain constraints on $\mathbf{U}$, such as non-negative constraint and orthogonal constraint [8] [11] [12].

However, all these pseudo-label approaches have similar drawbacks: the cluster labels are usually not accurate enough [8][12]. The wrongful information contained in the pseudo labels can further mislead feature selection. Besides, they have $3 \sim 4$ parameters (e.g., number of pseudo labels and several regularization terms) to be specified in the objective function. In unsupervised setting, it is difficult to know the optimal parameters, which limits their practical utility.

Recently, feature selection for non-traditional data types has drawn increasing attention, such as feature selection for linked data [6] [15] [17] [19] and multi-view data [4] [18].

## 3 Formulations

### 3.1 Notations

Suppose we have $n$ data points $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$ and the number of features is $D$. Let $f_p$ denote the $p$-th feature ($p = 1, 2, \ldots, D$) and $\mathbf{x}_{ip}$ denotes the value of $p$-th feature of $\mathbf{x}_i$. Our goal is to select $d$ ($d < D$) discriminative features.

As other similarity based feature selection methods [5], we first construct a kNN similarity graph $\mathcal{G}$ from $\mathbf{X}$ (e.g., by cosine similarity). In this graph $\mathcal{G}$, each data instance is connected with its $k$ nearest neighbors. For each data instance $\mathbf{x}_i$ ($i = 1, \ldots, n$), we denote its neighbors as $\mathcal{N}(\mathbf{x}_i)$ and the set of non-neighbors as $\mathcal{NN}(\mathbf{x}_i)$.

### 3.2 Discriminatively Exploiting Similarity

In general, we want to select features highly indicative of certain classes/topics. In supervised setting, labels provide clear guidance for feature selection: those features that

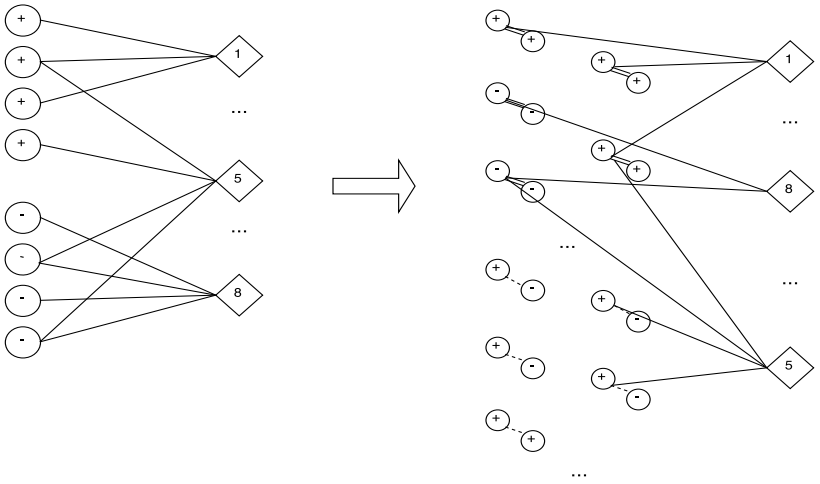



Fig. 1: Supervised Scenario with real labels (left) v.s. Unsupervised Scenario with SimLabels (right). Circles are data points and diamonds are terms/features. Link between data instance and terms indicates the occurrence of term. On the left, '+' and '-' denote two classes. Double-edged line denotes similar pair and dashed line denotes dissimilar pair

lead to good separability of different classes should be good features. If a term is usually shared by data points from the same class and rarely shared by data points from different classes, then this term is likely to be a discriminative one. In comparison, generic terms are shared indiscriminatively by data points from different classes. Consider the example shown in Figure 1. Feature $1$ and $8$ are discriminative ones but feature $5$ is not discriminative.

Our goal is to exploit this intuition in unsupervised case. Based on the kNN similarity graph, we first define *SimLabel* to divide pairs of data points into to classes: must-link (similar pairs) and cannot-link (dissimilar pairs).

**Definition 1 SimLabel***: Given the kNN graph $\mathcal{G}$ constructed from pairwise similarity, we label each pair of data points $(\mathbf{x}_i, \mathbf{x}_j)$ $(i, j \in \{1, 2, \ldots, n\}, i \neq j)$, with SimLabel $l^s_{ij}$ defined as below:*

$$l^s_{ij} = \begin{cases} 1 & \textit{if } \mathbf{x}_i \in \mathcal{N}(\mathbf{x}_j) \textit{ or } \mathbf{x}_j \in \mathcal{N}(\mathbf{x}_i) \\ -1 & \textit{otherwise} \end{cases} \tag{1}$$

We refer to pairs with $l^s = 1$ as *must-links (similar pairs)* and pairs with $l^s = -1$ as *cannot-links (dissimilar pairs)*. Let us denote the set of similar pairs and set of dissimilar pairs as $\Omega_s$ and $\Omega_d$, respectively. We also use $\Omega = \Omega_s \cup \Omega_d$ to denote the set of all pairs. Note that some non-neighbors are not necessarily very different (e.g. data instance $i$ and its $(k+1)$ closest data instance), but most of them are not as close as neighbors.

To make better use of similarity for selecting discriminative features, we exploit the following intuition: two neighbors are more likely to be of the same class than

two random non-neighbors. If a term is shared more often by neighbors than by non-neighbors, it is likely to be discriminative for certain class. In this sense, whether two data points are similar or not can serve the similar functionality of class labels to guide feature selection. We refer to this approach as Discriminatively Exploiting Similarity (DES). Consider feature $1$, $5$ and $8$ in Figure 1 for example. Discriminative features such as $1$ and $8$ are are shared more often by similar pairs, while generic feature $5$ is shared by almost equal amount of similar pairs and dissimilar pairs. For a real-world example, one can consider a collection of research papers on different topics (e.g., Machine Learning, Database and OS). Discriminative terms such as *SVM* and *classification* appear more often in pairs of similar papers. In comparison, generic terms such as *compare* and *propose* appear equally likely in similar papers and dissimilar papers.

In supervised feature selection, each data instance is an instance for learning. In the framework of DES, a pair of data points, rather than a single data instance, becomes the basic instance for learning. Since the instance in DES is a pair of data points, data features cannot be directly used for feature selection. We derive *SimFeatures* from for each pair.

**Definition 2 SimFeature***: Given a pair of data points $(\mathbf{x}_i, \mathbf{x}_j)$, the $p$-th SimFeature is defined as the product of corresponding feature values:*

$$(\mathbf{x}_i, \mathbf{x}_j)_p = \mathbf{x}_{ip} \cdot \mathbf{x}_{jp} \tag{2}$$

For example, for data points with binary features (e.i., term occurrence), the $p$-th SimFeature for pair $(\mathbf{x}_i, \mathbf{x}_j)$ is whether two data points $\mathbf{x}_i$ and $\mathbf{x}_j$ both have term $p$.

Our approach uses neighbor/non-neighbor relationship to serve as a proxy of class-belonging information. We aim to exploit the contrast between similar and dissimilar pairs rather than preserving the similarity itself. The contrast can provide useful information for selecting discriminative features. Compared with pseudo label based approaches, we do not explicitly construct labels and therefore the number of classes does not need to be specified explicitly. By discriminatively preserving similarity, DES combines the strength of similarity based approach and pseudo-label based approach.

## 4 Instantiations of the DES

In previous section, we introduce the general idea of DES. In this section, we adapt ideas from supervised feature selection and combine them with DES.

### 4.1 Hypothesis test based DES (HT-DES)

There are statistical test based supervised feature selection methods, such as Chi-square test [9]. Chi-square tests the null hypothesis that the given feature is independent of the class label. The features with higher test statistics are selected since this indicates the null hypothesis should be rejected and hence such features are highly correlated with the class label.

Inspired by Chi-square test, we propose a hypothesis testing based approach to exploit the difference of similar/dissimilar pairs. We test whether a feature has higher

proportion in similar pairs than in dissimilar pairs. If a feature appears more often in similar pairs than in dissimilar pairs, it is likely to be an informative feature.

Specifically, we perform two proportion one-tailed z-test. For a feature, we use $p_s$ to denote the proportion of its presence in similar pairs and $p_d$ the proportion of this feature in dissimilar pairs. The null hypothesis and alternative hypothesis can be formed as follows.

$$\begin{aligned} H_0 &: p_s = p_d \\ H_1 &: p_s > p_d \end{aligned} \tag{3}$$

**Pooled sample proportion.** Since the null hypothesis assumes $p_s = p_d$, we use a pooled proportion $\hat{p}$ to calculate the standard error.

$$\hat{p} = \frac{p_s \cdot n_s + p_d \cdot n_d}{n_s + n_d} \tag{4}$$

where $n_s$ and $n_d$ are the numbers of sampled similar and dissimilar pairs, respectively.
**Standard error** With the pooled proportion $p$, we can compute the standard error.

$$SE = \sqrt{\hat{p}(1-\hat{p})(\frac{1}{n_s} + \frac{1}{n_d})} \tag{5}$$

**Test statistics** We use the following *z-score* as test statistic for the difference in proportions.

$$z = \frac{p_s - p_d}{SE} \tag{6}$$

Features with high z-scores are shared significantly more often by similar pairs than by dissimilar pairs. Low z-score means that the feature is almost equally possible to appear in both similar and dissimilar pairs and hence less discriminative. For the example in Figure 1, feature 1 and 8 will have high z-score and feature 5 will have low z-scores. To obtain high-quality features, we can select the top features with high z-scores.

### 4.2 Classification-based DES (CL-DES)

HT-DES evaluates features individually and the z-score of one feature is not influenced by the z-scores of other features. So the selected subset of features can be highly redundant. Such redundancy would lead to higher computational cost and potentially degenerated performance.

In this section, we present a classification based approach to evaluate features jointly. The intuition is that discriminative features should be able to distinguish similar pairs from dissimilar pairs. Class labels establish the difference between data instances. In our DES framework, similarity can establish the difference between instance pairs and acts similarly as the class labels do in supervised setting. If a feature is highly indicative of similarity relationship, it is likely to be a useful feature.

To perform feature selection, we first introduce a weight vector $\mathbf{w}$ as features' importance scores since not all features are equally important. By using $\mathbf{w}$, we define *Weighted Similarity* between two instances:

**Definition 3 Weighted Similarity**: *For two instances* $(\mathbf{x}_i, \mathbf{x}_j)$ $(i, j \in \{1, 2, \ldots, n\}, i \neq j)$*, their weighted similarity* $s_{ij}^{\mathbf{w}}$ *w.r.t weight vector* $\mathbf{w}$ *is defined as:*

$$s_{ij}^{\mathbf{w}} = \mathbf{x}_i^T diag(\mathbf{w}) \mathbf{x}_j \tag{7}$$

*where* $diag(\mathbf{w})$ *is the diagonal matrix with* $\mathbf{w}$ *as diagonal elements.*

It is desirable that the weighted similarity can distinguish similar pairs from dissimilar pairs:

$$s_{ij}^{\mathbf{w}} \cdot l_{ij}^s > 1, \ \forall (i, j) \in \Omega \tag{8}$$

This objective makes our formulation essentially different from the similarity preserving framework, since our goal is to separate similar pairs from dissimilar pairs rather than preserving similarity itself.

In supervised feature selection, $L_1$-regularization [16] is able to take into consideration the redundancy among features and achieves great success due to its simplicity and effectiveness. To get sparse weight vector $\mathbf{w}$, we add L1 regularization to our DES framework:

$$\begin{aligned} \min_{\mathbf{w}} \ & \|\mathbf{w}\|_1 \\ \text{s.t.} \ & s_{ij}^{\mathbf{w}} \cdot l_{ij}^s \geq 1, \forall (i, j) \in \Omega \end{aligned} \tag{9}$$

However, similar/dissimilar pairs may not always be separable given the weight vector $\mathbf{w}$, since the original similar/dissimilar pairs constructed from features can be noisy. So, to address this issue, we add an slack variable $\mu_{ij}$ to impose soft margin.

$$\begin{aligned} \min_{\mathbf{w}} \ & \frac{1}{|\Omega|} \sum_{(i,j) \in \Omega} \mu_{ij} + \lambda \|\mathbf{w}\|_1 \\ \text{s.t.} \ & s_{ij}^{\mathbf{w}} \cdot l_{ij}^s \geq 1 - \mu_{ij}, \forall (i, j) \in \Omega \end{aligned} \tag{10}$$

Eq (10) can also be interpreted as $L_1$ regularized SVM on pair-wise instances with SimLabel and SimFeatures. Discriminative features are more likely to appear in similar pairs and would have relatively larger positive weights. Indiscriminative features have little utility in differentiating similar pairs and dissimilar pairs. As a result, the weights of such features are close to zero or negative. If we rank the features w.r.t their weights, we can select the top ones as high quality features.

## 5 Optimization

For HT-DES, the optimization is straightforward: one can simply calculate the z-scores of each feature and select the top ones. There are $O(nk)$ similar pairs and $O(n(n-k))$ dissimilar pairs. So the number of dissimilar pairs is much larger than number of similar pairs since $k \ll n$. To avoid imbalanced distribution of SimLabels, we employ a bootstrapping based approach to sample equal amounts of similar and dissimilar pairs for HT-DES and CL-DES.

For CL-DES, the objective function is not differentiable due to hinge loss and $L_1$ regularization, we calculate subgradient for the objective function and optimize it by

stochastic subgradient descent. For a data instance pair $(\mathbf{x}_i, \mathbf{x}_j)$, the subgradient w.r.t $w_p$ $(p = 1, \ldots, D)$ is:

$$\frac{\partial \mathcal{L}(\mathbf{x}_i, \mathbf{x}_j)}{\partial w_p} = \frac{\partial}{\partial w_p}\mu_{ij} + \lambda \cdot \text{sign}(w_p) \tag{11}$$

where the subdifferential $\frac{\partial}{\partial w_p}\mu_{ij}$ can be calculated as follows.

$$\frac{\partial}{\partial w_p}\mu_{ij} = \begin{cases} x_{ip} \cdot x_{jp} \cdot l^s_{ij}, & \text{if } s^{\mathbf{w}}_{ij} \cdot l^s_{ij} < 1 \\ 0, & \text{otherwise} \end{cases} \tag{12}$$

**Algorithm 1** Stochastic Subgradient Descent Algorithm for CL-DES

```
w^0 ← [0, 0, ..., 0]
for (t in 1..T) do
    Generate random number α ∈ (0, 1)
    if α > 0.5 then
        Sample a similar pair (x_i, x_j) (x_j ∈ 𝒩(x_i))
    else
        Sample a dissimilar pair (x_i, x_j) (x_j ∈ 𝒩𝒩(x_i))
    end if
    Update w^t using the sampled pair with formula (11)
end for
Sort features w.r.t. w[i] and output the top d features
```

The Stochastic Sub-gradient Descent method is shown in Algorithm 1 and the time complexity is $O(mT)$, where $m$ is the average number non-zero features in each data instance and $T$ is the total number of iterations.

## 6 Experiment

In this section, we conduct experiments on six publicly available datasets. We compare DES with several state-of-the-art approaches.

### 6.1 Baselines

We compared our approach to four unsupervised feature selection methods and one supervised method. LS is a similarity-preserving approach. UDFS and NDFS are regression based methods which also consider the similarity information.

- All Features: It uses all the features for evaluation.
- Laplacian Score (LS): Laplacian score [5] selects the features which can best preserve the local manifold structure of data points.
- UDFS: Unsupervised Discriminative Feature Selection [22] exploits the local structure with $L_{2,1}$ norm regularized subspace learning.

– RSFS: Robust Spectral Feature Selection [12] selects features by the robust spectral analysis with $L_{2,1}$ norm regularized regression.
– FSASL: A recently proposed approach [2] which performs joint local structure learning and feature selection based on $L_{2,1}$ norm.
– Mutual Information (MI): We also include a widely-used supervised feature selection method which evaluates features by their mutual information with class labels. Since there are multiple classes, for each feature we use its average mutual information with different classes.

### 6.2 Datasets

We use six publicly available datasets: CNN dataset[1], Handwritten digits Dataset[2], BBCSport dataset[3], Guardian dataset[4], BlogCatalog[5] blog-posts dataset, Newsgroup [6]. The baseline methods UDFS, RSFS and FSASL are prohibitively slow for large datasets. The original data of the latter two datasets are too large and therefore we sample a subset of them.

– CNN: CNN Web news with 7 classes (the category information contained in the RSS feeds for each news article can be viewed as reliable ground truth). Titles, abstracts, and text body contents are extracted as the text features.
– Handwritten Digits: 2000 images of handwritten digits $0 \sim 9$ and we use the image pixels as features.
– BBCSport: It consists of 737 documents from the BBC Sport website corresponding to sports news articles in five topical areas from 2004-2005. The dataset has 5 classes: *athletics, cricket, football, rugby, tennis*.
– Guardian: It consists of 302 news stories from Guardian during the period February - April 2009. Each story is annotated with one of the six topical labels based on the dominant topic: *business, entertainment, health, politics, sport, tech*.
– Newsgroup: A subset of Newsgroup dataset on four topics: *comp.graphics, rec.sport.baseball, rec.motorcycles, sci.electronics*.
– BlogCatalog: A subset of users' blogposts from BlogCatalog in the following categories (100 posts are sampled for each category): *cycling, military, architecture, commodities/futures, vacation rentals*.

The statistics of six datasets are summarized in Table 1.

### 6.3 Experimental Setting

In this section, we evaluate the quality of selected features by their clustering performance. Following the typical setting of evaluation for unsupervised feature selection

[1] https://sites.google.com/site/qianmingjie/home/datasets/cnn-and-fox-news
[2] https://archive.ics.uci.edu/ml/datasets/Multiple+Features
[3] http://mlg.ucd.ie/datasets/bbc.html
[4] http://mlg.ucd.ie/datasets/3sources.html
[5] http://dmml.asu.edu/users/xufei/datasets.html
[6] http://www.cs.umb.edu/∼smimarog/textmining/datasets/

[8] [20] [22], we use Accuracy and Normalized Mutual Information (NMI) to evaluate the result of clustering. Accuracy is defined as follows.

$$Accuracy = \frac{1}{n}\sum_{i=1}^{n}\mathcal{I}(c_i = map(p_i)) \tag{13}$$

where $p_i$ is the clustering result of data instance $i$ and $c_i$ is its ground truth label. $map(\cdot)$ is a permutation mapping function that maps $p_i$ to a class label using Kuhn-Munkres Algorithm.

Normalized Mutual Information (NMI) is another popular metric for evaluating clustering performance. Let $C$ be the set of clusters from the ground truth and $C'$ obtained from a clustering algorithm. Their mutual information $MI(C, C')$ is defined as follows:

$$MI(C, C') = \sum_{c_i \in C, c'_j \in C'} p(c_i, c'_j) \log \frac{p(c_i, c'_j)}{p(c_i)p(c'_j)} \tag{14}$$

where $p(c_i)$ and $p(c'_j)$ are the probabilities that a random data instance from the data set belongs to $c_i$ and $c'_j$, respectively, and $p(c_i, c'_j)$ is the joint probability that the data instance belongs to the cluster $c_i$ and $c'_j$ at the same time. In our experiments, we use the normalized mutual information.

$$NMI(C, C') = \frac{MI(C, C')}{max(H(C), H(C')} \tag{15}$$

where $H(C)$ and $H(C')$ are the entropy of $C$ and $C'$. Higher value of NMI indicates better quality of clustering.

We set $k = 5$ for the kNN neighbor size in both our approach and the baseline methods following previous evaluation convention [8]. For $\lambda$ in CL-DES, we set it to be $10^{-4}$ for all datasets, since in preliminary experiments we found the performance is not sensitive to $\lambda$ when it is in $(10^{-6}, 10^{-3})$. For HT-DES and CL-DES, we sample $40000$ pairs for the optimization, as we observe sampling more pairs usually have similar performance.

For the number of pseudo-classes/latent dimensions in UDFS, RSFS and FSASL, we use the ground-truth number of classes. Note that it actually benefits these pseudo-label based baselines with extra information about the data (and therefore certain advantages) since our approach does not need the number of classes as input. For the parameter to enforce the orthogonal constraint in pseudo-label methods [12], we use $10^8$ as in the original papers. However, UDFS, RSFS and FSASL also require specifying the values of several other regularization parameters. In supervised learning, one can perform grid search for the parameters on a validation dataset; but there is no good way to determine the parameter values in unsupervised learning since we assume class labels are not available. In their original papers, all the class labels are used to find the best parameters. However, this violates the assumption of no supervision and favors the methods with best overfitting ability. Nonetheless, we perform grid search in the range of $\{0.1, 1, 10\}$ for the regularization parameters in UDFS, RSFS and FSASL (except for  in FSASL, for which we do grid search in $\{0.001, 0.01, 0.1\}$ since $\gamma$ should be a

value in the range of $0 \sim 1$, as suggested in [2]). Besides the best performance, we also report the median performance for them

Following the convention in previous work [1] [22], we use KMeans[7] with cosine similarity for clustering evaluation. Since KMeans is affected by the initial seeds, we repeat the experiment for $20$ times and report the average performance. We vary the number of features $d$ in the range of $\{100, 200, 300, 400, 600\}$ (except for Handwritten digit dataset, which only has $240$ features).

### 6.4 Clustering Results

The clustering performance on six datasets is shown in Figure 2 and Figure 3. The performance of baseline methods shown in Figure 2 and Figure 3 are under their best parameter values. HT-DES and Mutual Information cannot handle continuous feature values and hence are not applied to the handwritten dataset.

The experimental results show that feature selection is a very effective technique for clustering. With much less features, DES can obtain better accuracy and NMI than using all the features. For instance, compared with using all $4612$ features, CL-DES with only $100$ features improves the clustering accuracy by $28\%$ on BBCSport dataset. Another thing worth noting is that, when the number of selected features is small (such as $100$ and $200$), the improvement of DES over using all the features is also significant. This means that DES is capable of ranking high-quality features at the top. Besides the improved accuracy and NMI, using selected features rather than all features can also lead to better interpretability for human to analyze.

Among the two DES instantiations, CL-DES tends to have better clustering performance than HT-DES. This demonstrates the importance of evaluating features in a joint manner. HT-DES does not take into consideration correlation between features and there could be more redundancy in selected features.

When comparing DES with other unsupervised baseline methods, we observe that DES methods (especially CL-DES) with fixed $\lambda$ perform better than baseline methods (with best parameter settings) in terms of both accuracy and NMI on most datasets. For example, on BlogCatalog dataset, CL-DES outperforms the most competitive baseline by $16\%$ with $200$ features.

Although HT-DES does not evaluate feature jointly, it still outperforms most unsupervised baseline methods substantially. This illustrates the power of exploiting the implicit class information contained in similar/dissimilar pairs. The baseline methods also utilize similarity in certain ways. For example, LS attempts to preserve the local manifold. RSFS generates pseudo-labels through spectral clustering on the similarity graph. But the inaccurate clustering labels can be viewed as a lossy compression of similarity information and may mislead feature selection. DES directly exploits the implicit class information embedded in similarity pairs without generating intermediate labels. The experimental results show that DES is a much more effective way for utilizing similarity information.

[7] We use the code at `http://www.cad.zju.edu.cn/home/dengcai/Data/Clustering.html`

If we compare DES with the supervised method MI, we can see the performance of CL-DES is close to MI. It is usually very difficult for an unsupervised method to achieve performance comparable to a supervised method. This further illustrates strength of Discriminatively Exploiting Similarity (DES).

Table 1: Statistics of datasets

| Statistics | CNN | Handwritten Digits | BBC Sport | BlogCatalog | Guardian | Newsgroup |
|---|---|---|---|---|---|---|
| # of instances | 2107 | 2000 | 737 | 500 | 302 | 1575 |
| # of features | 6262 | 240 | 4612 | 4547 | 3631 | 2849 |
| # of classes | 7 | 10 | 5 | 5 | 6 | 4 |

Table 2: Relative performance (%) of median Accuracy/NMI (percentage) compared to the performance reported in Figure 2 and 3

| Statistics | CNN | Handwritten | BBCSport | BlogCatalog | Guardian | Newsgroup |
|---|---|---|---|---|---|---|
| UDFS | -3.56/-6.74 | -3.96/-2.06 | -10.67/-23.91 | -8.74/-18.33 | -11.72/-25.33 | -10.9/-31.64 |
| RSFS | -19.18/-34.07 | -7.68/-4.80 | -7.55/-11.07 | -16.37/-37.49 | -6.39/-9.50 | -19.08/-38.97 |
| FSASL | -14.19/-17.72 | -8.06/-12.15 | -6.41/-4.30 | -9.49/-16.95 | -6.84/-12.40 | -12.17/-24.49 |
| CL-DES | -0.53/-0.68 | -0.55/-0.47 | -0.32/-0.19 | -0.06/-0.03 | -0.75/-0.46 | +0.09/-0.48 |

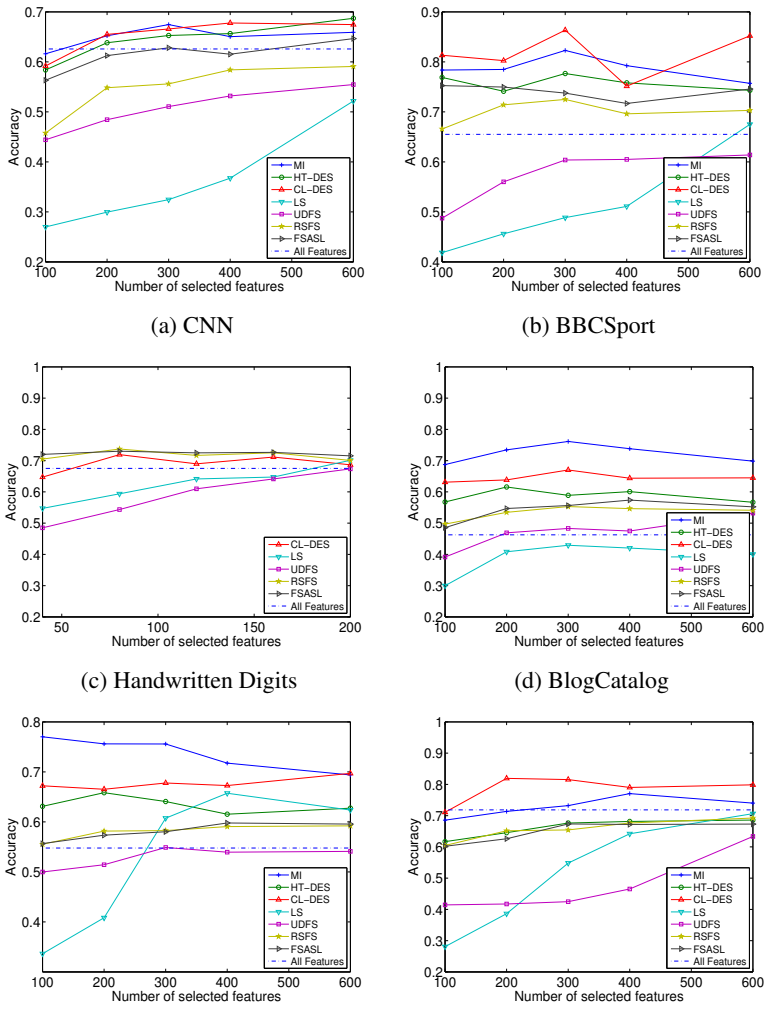
### 6.5 Sensitivity Analysis

CL-DES has one regularization parameter $\lambda$ and we study how this parameter affects the quality of selected features. In Figure 4, we can observe that CL-DES performs consistently well as long as $\lambda$ is smaller than $10^{-3}$.

We also show in Table 2 how the median performance for UDFS, RSFS and FSASL compares with the best performance shown in Figure 2. It can be observed that these baseline methods are sensitive to the parameter values and the median performance is usually $5\% \sim 40\%$ lower than their best performance. Since one cannot know the best parameter combination for these methods in unsupervised setting, the median performance is more realistic to expect in practice. In contrast, we also report the median perform for CL-DES from $\lambda = \{10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}\}$ and we observe that the median performance is very close to performance of $\lambda = 10^{-4}$. This makes the proposed method more practical for real-world applications.

## 7 Conclusion

In this paper, we propose a new perspective for unsupervised feature selection which considers the similarity relationship as pseudo must-link/cannot-links. This new perspective enables us to adapt classic supervised feature selection ideas into our pair-wise

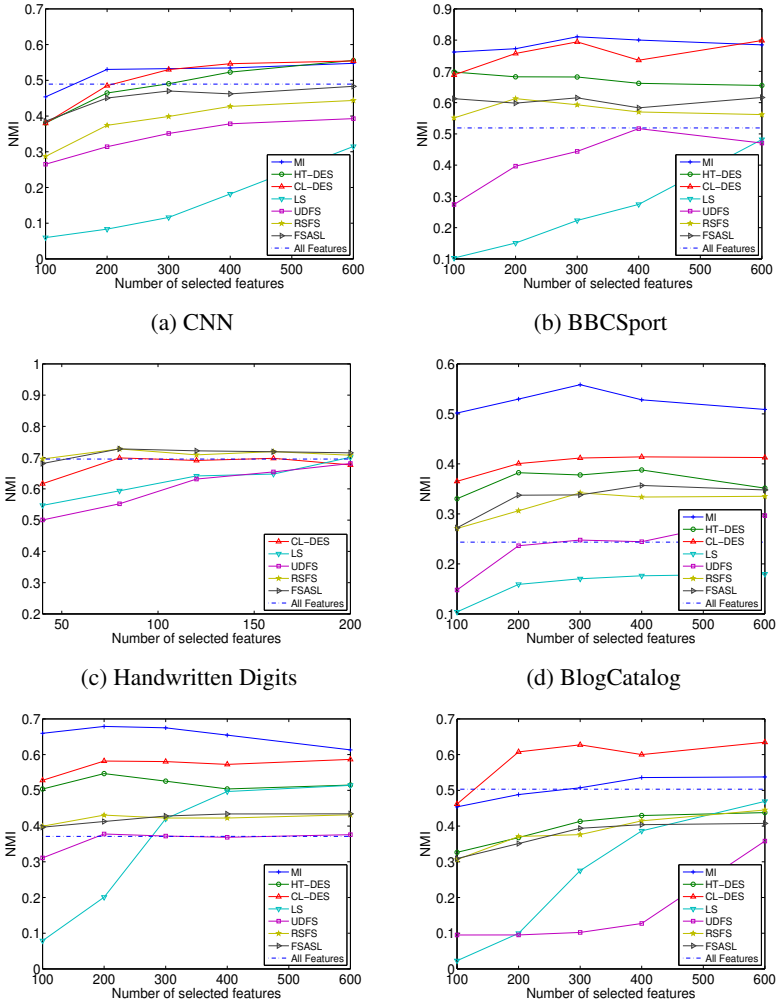



(a) CNN (b) BBCSport

(c) Handwritten Digits (d) BlogCatalog

(e) Guardian (f) Newsgroup

Fig. 2: Accuracy of clustering results



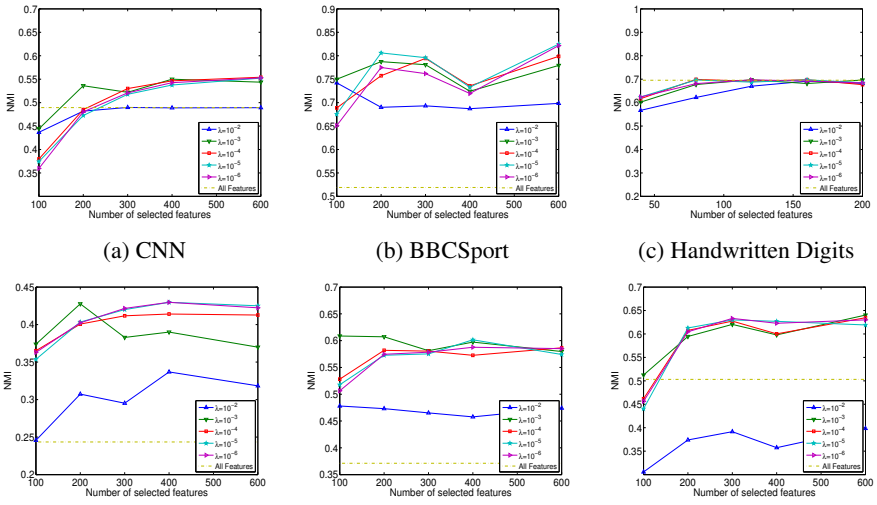

(a) CNN (b) BBCSport

(c) Handwritten Digits (d) BlogCatalog

(e) Guardian (f) Newsgroup

Fig. 3: NMI of clustering results

formulation. We present hypothesis testing based and classification based approaches as instantiations of our framework. Empirical results show that the proposed method, although frustratingly simple, can select more discriminative features than state-of-the-art unsupervised approaches and even achieve comparable performance as the supervised mutual information approach.




(a) CNN (b) BBCSport (c) Handwritten Digits

(d) BlogCatalog (e) Guardian (f) Newsgroup

Fig. 4: NMI of clustering results with features selected by CL-DES under different values of $\lambda$

## References


1. D. Cai, C. Zhang, and X. He. Unsupervised feature selection for multi-cluster data. In *KDD*, pages 333–342, 2010.
2. L. Du and Y.-D. Shen. Unsupervised feature selection with adaptive structure learning. In *KDD*, 2015.
3. R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley, 2 edition, 2001.
4. Y. Feng, J. Xiao, Y. Zhuang, and X. L. 0002. Adaptive unsupervised multi-view feature selection for visual concept recognition. In *ACCV (1)*, volume 7724, pages 343–357, 2012.
5. X. He, D. Cai, and P. Niyogi. Laplacian score for feature selection. In *NIPS*, 2005.
6. J. Li, X. Hu, L. Jian, and H. Liu. Toward time-evolving feature selection on dynamic networks. In *IEEE 16th International Conference on Data Mining (ICDM), December 12-15, 2016, Barcelona, Spain*, pages 1003–1008, 2016.
7. J. Li, J. Tang, and H. Liu. Reconstruction-based unsupervised feature selection: An embedded approach. In *IJCAI*, 2017.
8. Z. Li, Y. Yang, J. Liu, X. Zhou, and H. Lu. Unsupervised feature selection using nonnegative spectral analysis. In *AAAI*, 2012.

9. H. Liu and R. Setiono. Chi2: Feature selection and discretization of numeric attributes. In *In Proceedings of 7th IEEE Int'l Conference on Tools with Artificial Intelligence*, 1995.
10. F. Nie, H. Huang, X. Cai, and C. H. Q. Ding. Efficient and robust feature selection via joint l2, 1-norms minimization. In *NIPS*, pages 1813–1821, 2010.
11. M. Qian and C. Zhai. Robust unsupervised feature selection. In *IJCAI*, 2013.
12. L. Shi, L. Du, and Y.-D. Shen. Robust spectral learning for unsupervised feature selection. In *ICDM*, 2014.
13. L. Song, A. J. Smola, A. Gretton, K. M. Borgwardt, and J. Bedo. Supervised feature selection via dependence estimation. In *ICML*, volume 227, pages 823–830. ACM, 2007.
14. L. Sun, Z. Li, Q. Yan, W. Srisa-an, and Y. Pan. Sigpid: significant permission identification for android malware detection. In *Malicious and Unwanted Software (MALWARE), 2016 11th International Conference on*, pages 1–8. IEEE, 2016.
15. J. Tang and H. Liu. Unsupervised feature selection for linked social media data. In *KDD*, pages 904–912, 2012.
16. R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society (Series B)*, 58:267–288, 1996.
17. X. Wei, B. Cao, and P. S. Yu. Unsupervised feature selection on networks: A generative view. In *AAAI*, pages 2215–2221, 2016.
18. X. Wei, B. Cao, and P. S. Yu. Multi-view unsupervised feature selection by cross-diffused matrix alignment. In *International Joint Conference on Neural Networks, (IJCNN)*, pages 494–501, 2017.
19. X. Wei, S. Xie, and P. S. Yu. Efficient partial order preserving unsupervised feature selection on networks. In *SDM*, pages 82–90, 2015.
20. X. Wei and P. S. Yu. Unsupervised feature selection by preserving stochastic neighbors. In *AISTATS*, 2016.
21. J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31:210–227, 2009.
22. Y. Yang, H. T. Shen, Z. Ma, Z. Huang, and X. Zhou. l2, 1-norm regularized discriminative feature selection for unsupervised learning. In *IJCAI*, pages 1589–1594, 2011.
23. Z. Zhao and H. Liu. Spectral feature selection for supervised and unsupervised learning. In *ICML*, volume 227, pages 1151–1157, 2007.
24. Z. Zhao, L. Wang, and H. Liu. Efficient spectral feature selection with minimum redundancy. In *AAAI*, 2010.