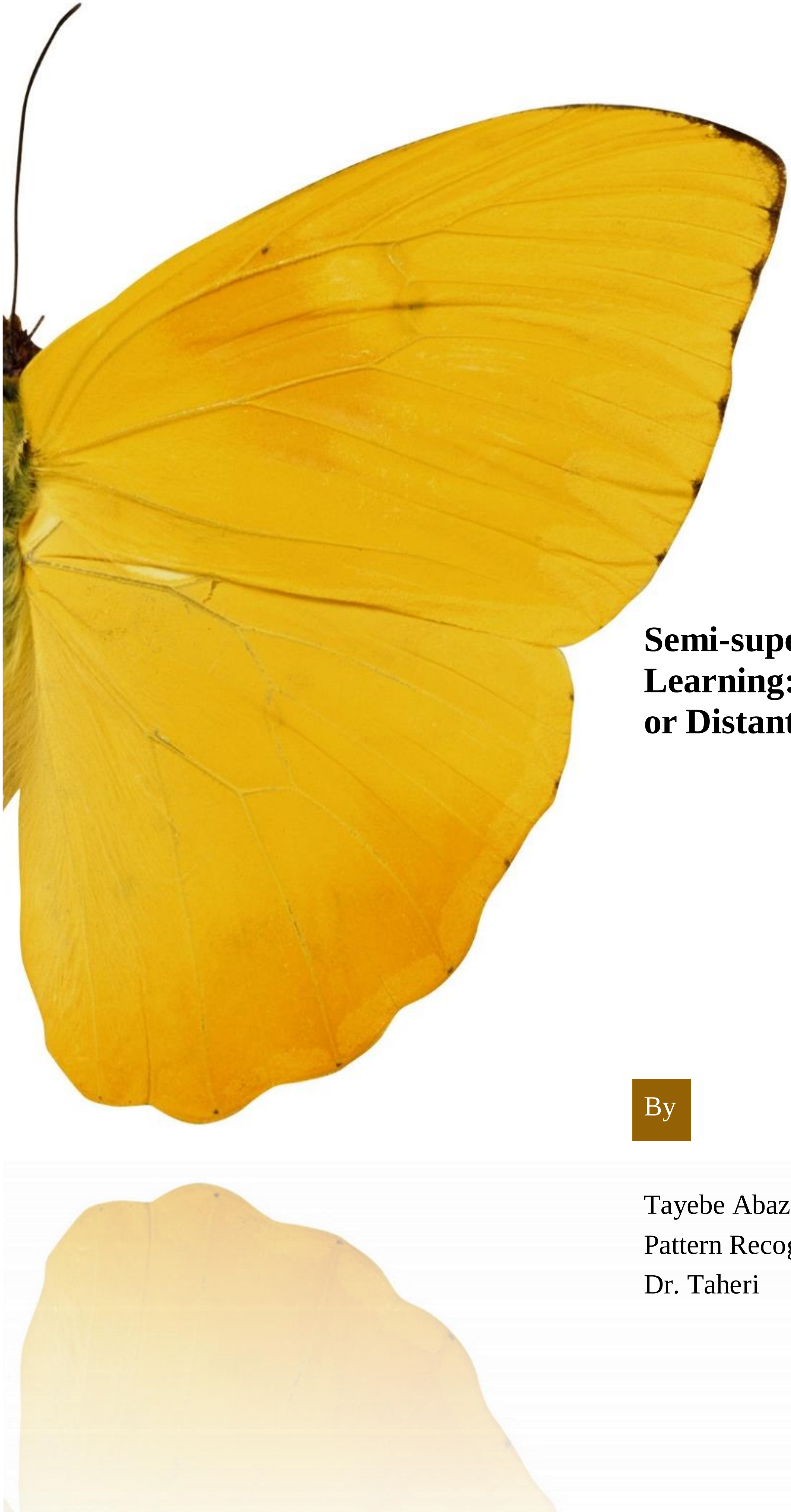


Semi-supervised Graph Learning: Near Strangers or Distant Relatives

By

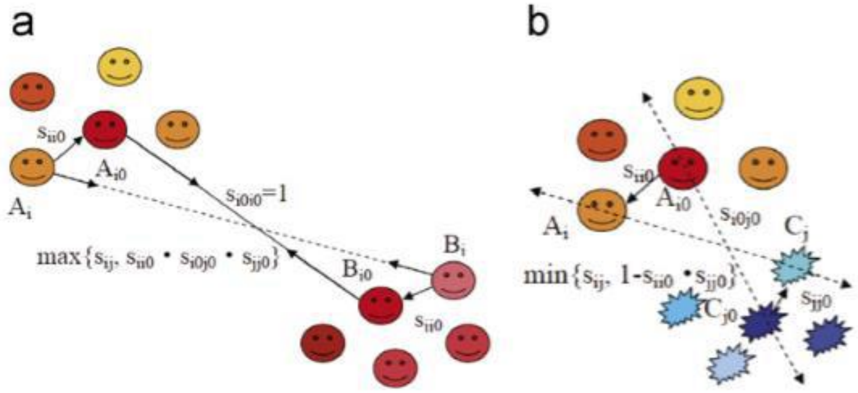
Tayebe Abazar
Pattern Recognition
Dr. Taheri

Cluster analysis or **clustering** is the task of grouping a set of objects in such a way that objects in the same group (called a **cluster**) are more similar (in some sense) to each other than to those in other groups (clusters).

In this paper we have two aims that are Clustering and dimensionality reduction. For achieve these, a semi-supervise approach using the most of prior knowledge from limited pairwise constraints proposed.

This paper introduces a ”Near Strangers or Distant Relatives” model for semi-supervised clustering. Under this model, neighbors of dissimilar instances are dissimilar, while neighbors of similar instances are also similar.

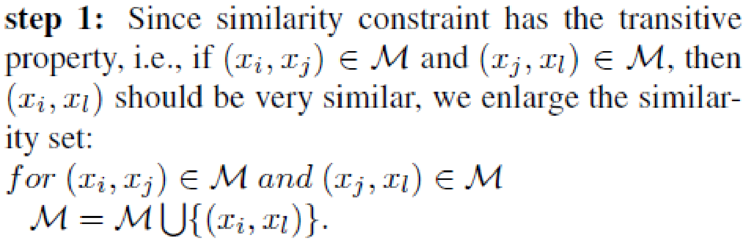
According to this model, if two instances are similar then these two instances and the near neighbors of these will be near to each other and if two instances are dissimilar then these two instances and the near neighbors of these, will be far away from each other. So by this approach we have a space-level constraint instead of instance-level constraint.



Suppose we have an undirected graph G (X, E, S) that X is a set of node, E is a set of Edges and S is a similarity matrix of graph

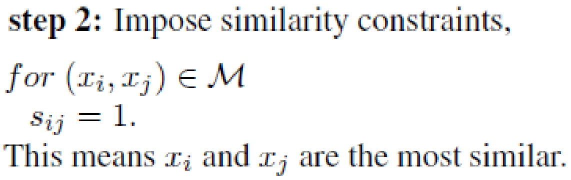
G. $s_{ij} = 0$ if there is no edge between x_i and x_j. Our goal is to modify the similarity matrix by instance-level constraints, along with the induced space-level constraints. This modification may involve a radical change in the topology of the original space, which is expected to improve the boundaries of mostly correct clusters.

For measuring similarity constraints, a method proposed to impose and propagate similarity constraint. Steps of these method are:



step 1: Since similarity constraint has the transitive property, i.e., if $(x_i, x_j) \in \mathcal{M}$ and $(x_j, x_l) \in \mathcal{M}$, then (x_i, x_l) should be very similar, we enlarge the similarity set:

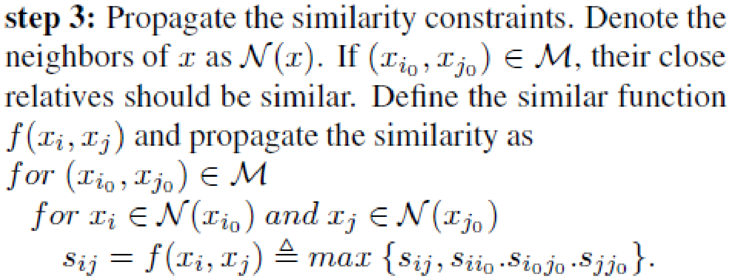
```
$for\ (x_i, x_j) \in \mathcal{M}\ and\ (x_j, x_l) \in \mathcal{M}$
  $\mathcal{M} = \mathcal{M} \bigcup \{(x_i, x_l)\}.$
```



step 2: Impose similarity constraints,

```
$for\ (x_i, x_j) \in \mathcal{M}$
  $s_{ij} = 1.$
```

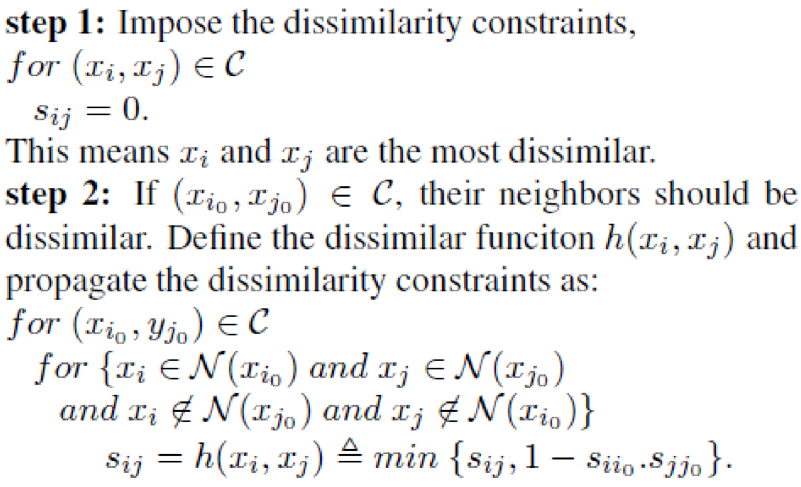
This means x_i and x_j are the most similar.



step 3: Propagate the similarity constraints. Denote the neighbors of x as $\mathcal{N}(x)$. If $(x_{i_0}, x_{j_0}) \in \mathcal{M}$, their close relatives should be similar. Define the similar function $f(x_i, x_j)$ and propagate the similarity as

```
$for\ (x_{i_0}, x_{j_0}) \in \mathcal{M}$
  $for\ x_i \in \mathcal{N}(x_{i_0})\ and\ x_j \in \mathcal{N}(x_{j_0})$
    $s_{ij} = f(x_i, x_j) \triangleq max\ \{s_{ij}, s_{ii_0}.s_{i_0 j_0}.s_{jj_0}\}.$
```

For measuring dissimilarity constraints, a method proposed to impose dissimilarity constraint. Steps of these method are:



step 1: Impose the dissimilarity constraints,

```
$for\ (x_i, x_j) \in \mathcal{C}$
  $s_{ij} = 0.$
```

This means x_i and x_j are the most dissimilar.

step 2: If $(x_{i_0}, x_{j_0}) \in \mathcal{C}$, their neighbors should be dissimilar. Define the dissimilar funciton $h(x_i, x_j)$ and propagate the dissimilarity constraints as:

```
$for\ (x_{i_0}, y_{j_0}) \in \mathcal{C}$
  $for\ \{x_i \in \mathcal{N}(x_{i_0})\ and\ x_j \in \mathcal{N}(x_{j_0})$
    $and\ x_i \notin \mathcal{N}(x_{j_0})\ and\ x_j \notin \mathcal{N}(x_{i_0})\}$
      $s_{ij} = h(x_i, x_j) \triangleq min\ \{s_{ij}, 1 - s_{ii_0}.s_{jj_0}\}.$
```

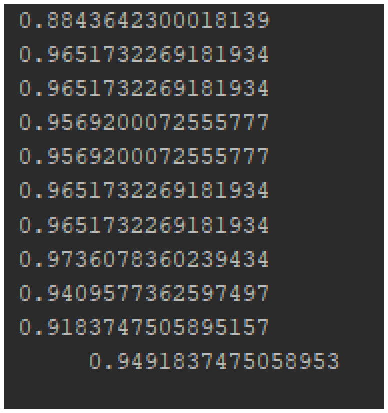
And then used a semi-supervise clustering method called Ncut for cluster the instances using similarity matrix. The goal of Ncut is to minimize between-cluster associations and maximize degrees (or volumes) of clusters simultaneously. The hole algorithm named NSDR-Ncut that pseudo code of this algorithm is as follow:

```
Inputs: Data set $X = \{x_1, x_2, \dots, x_n\}$, mustlink $\mathcal{M}$,
        cannotlink $\mathcal{C}$, number of clusters k,
        parameter $\sigma$, K nearest neighborhood.
Outputs: Cluster assignments of the points.
1. Construct a K nearest neighborhood graph $G$ over $X$.
2. Define the distance $d(x_i, x_j)$ as the shortest path
   connecting $x_i$ and $x_j$ in $G$.
3. Calculate the $S = (s_{ij} = exp(-d^2(x_i, x_j)/2\sigma^2))$.
4. Impose and propagate constraints $\mathcal{M}$ and $\mathcal{C}$.
5. Define $D = Diag(S\mathbf{1})$ and $W = D^{-1}S$.
6. Find the k largest eigenvectors $y_1, y_2, \dots, y_k$ of $W$,
   and form the matrix $Y = [y_1 y_2 \dots y_k]$.
7. Treating each row of $Y$ as a point in $\mathbb{R}^k$, cluster
   them into k clusters via k-Means.
8. Assign the original point $x_i$ to cluster $j$ if and only
   if row $j$ of $Y$ is assigned to cluster $j$.
```

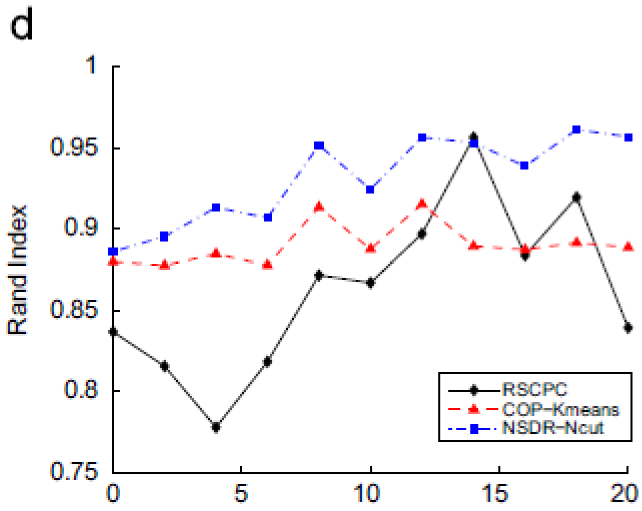
And finally the results of our implementation

Res:

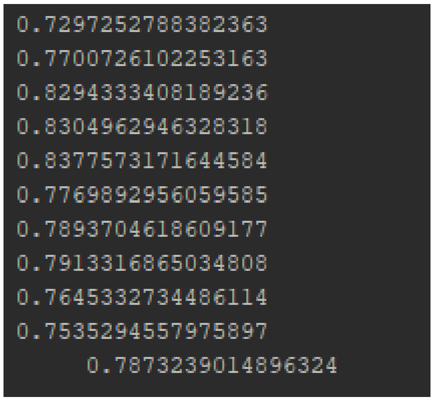
Iris - 10



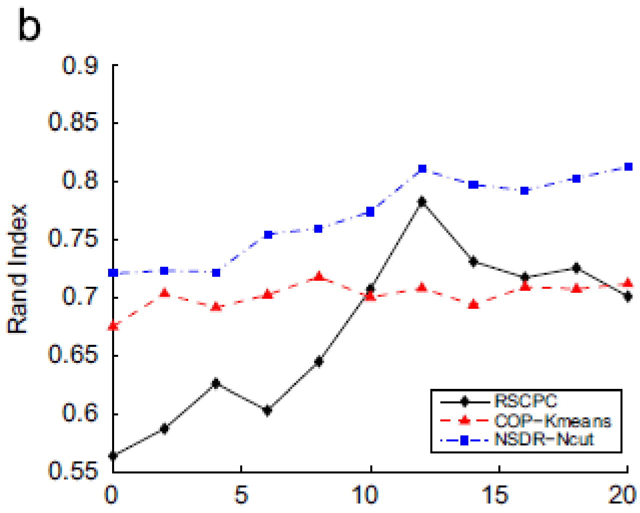
```
0.8843642300018139
0.9651732269181934
0.9651732269181934
0.9569200072555777
0.9569200072555777
0.9651732269181934
0.9651732269181934
0.9736078360239434
0.9409577362597497
0.9183747505895157
        0.9491837475058953
```



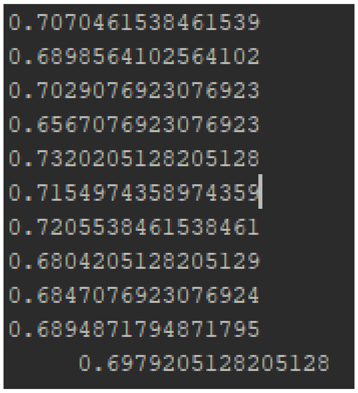
Dermatology - 10



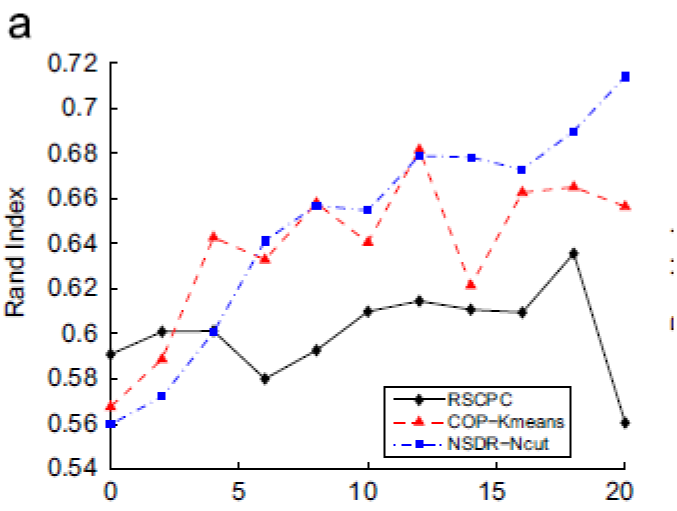
```
0.7297252788382363
0.7700726102253163
0.8294333408189236
0.8304962946328318
0.8377573171644584
0.7769892956059585
0.7893704618609177
0.7913316865034808
0.7645332734486114
0.7535294557975897
      0.7873239014896324
```



Balance scale – 10



```
0.7070461538461539
0.6898564102564102
0.7029076923076923
0.6567076923076923
0.7320205128205128
0.7154974358974359
0.7205538461538461
0.6804205128205129
0.6847076923076924
0.6894871794871795
      0.6979205128205128
```


a
Rand Index
RSCPC
COP−Kmeans
NSDR−Ncut