Paper One

Multi-View Clustering via Joint Nonnegative Matrix Factorization

Motivation

A great number of real-world datasets are naturally comprised of many views. Therefore, some clustering algorithms working on single-view cannot work on multiple-view dataset directly. The authors want to promote the classic NMF method such that it adapt on multi-view datasets. They propose an effective method utilizing NMF and PLSA algorithms, which can give us a meaning and comparable clustering solutions.

Related Theory

Existing multi-view clustering algorithms can be roughly classified into three categories:

1) Optimize certain loss function in which multi-view information has been integrated.
2) Project multi-view data into a lower dimensional space and then cluster them through conventional clustering method.
3) Cluster each single-view data and get many corresponding results, and then fuse different clustering results into single one result.

NMF

The idea of NMF is to find two matrix such: $X \approx UV^T$, which is equal to

$$\min_{U,V} ||X - UV^T||_F^2,\ s.t. U \geq 0, V \geq 0$$

The above objective function is not convex. The updating rules are:

$$U_{i,k} \leftarrow U_{i,k}\frac{(XV)_{i,k}}{(UV^TV)_{i,k}}, V_{i,k} \leftarrow V_{i,k}\frac{(XU)_{i,k}}{(VU^TU)_{i,k}}$$

However, there could be many possible solutions so we need to enforce some additional constraint to ensure the uniqueness of the clustering result.

PLSA

PLSA is a topic model, considering that God chooses a subject in a certain probability when writing an article, and then chooses a word in a certain probability under the subject, and repeats the process to complete an article:

$$p(d_i, w_j) = p(z_k|d_i)p(w_j|z_k)$$

Where d, w, z indicates documents, words, and topics.

Therefore, the solution to PLSA model is to compute $p(z_k|d_i)$ and $p(w_j|z_k)$. The classic method is EM algorithm:

EM algorithm is an iterative algorithm which contains E-step and M-step.

E-step:

Firstly, compute $p(z_k \mid d_i, w_j)$ according to the following formula. $p(z_k \mid d_i)$ and $p(w_j \mid z_k)$ are obtained from the last M-step. The initial values of them are randomly given.

$$p(z_k|d_i, w_j) = \frac{p(z_k|d_i)p(w_j|z_k)}{\sum_k p(z_k|d_i)p(w_j|z_k)}$$

M-step:

$p(z_k \mid d_i)$ and $p(w_j \mid z_k)$ are obtained according to the following formulas, where $p(z_k \mid d_i, w_j)$ is calculated in the E-step.

$$p(z_k|d_i) = \frac{\sum_j p(z_k|d_i, w_j)}{\sum_j \sum_k p(z_k|d_i, w_j)} \quad p(w_j|z_k) = \frac{\sum_i p(z_k|d_i, w_j)}{\sum_i \sum_j p(z_k|d_i, w_j)}$$

Main Theory

For traditional NMF algorithm, V_j^v is the low dimensional expression of j node based on basic matrix U^v. And the measure of coefficient matrix V^v and consensus matrix V^* is defined as: $D(V^{(v)}, V^*) = ||V^{(v)} - V^*||_F^2$

Since then, our overall goal should be:

$$\sum_{v=1}^{n_v} ||X^{(v)} - U^{(v)}(V^{(v)})^T||_F^2 + \sum_{v=1}^{n_v} \lambda_v ||V^{(v)} - V^*||_F^2$$

With the constraints such that, $\forall 1 \le k \le K, ||U_{.,k}^{(v)}||_1 = 1 \text{且} U^{(v)}, V^{(v)}, V^* \ge 0$

Moreover, authors also introduce Q:

$$Q^{(v)} = Diag(\sum_{i=1}^{M} U_{i,1}^v, \sum_{i=1}^{M} U_{i,2}^v, \ldots, \sum_{i=1}^{M} U_{i,K}^v)$$

in order to constraint U.

Therefore, the final objective function is supposed to be:

$$O = \sum_{v=1}^{n_v} ||X^{(v)} - U^{(v)}(V^{(v)})^T||_F^2 + \sum_{v=1}^{n_v} \lambda_v ||V^{(v)}Q^{(v)} - V^*||_F^2$$

Optimization:

In order to minimize the above objective function, and get the consensus matrix V^*, basic matrices $\{U^{(1)}, U^{(2)}, \ldots, U^{(n_v)}\}$ and coefficient matrices $\{V^{(1)}, V^{(2)}, \ldots, V^{(n_v)}\}$, we firstly fix V^* to update U^v and V^v, then fix U^v and V^v to update V^*:

Input: Nonnegative matrix $\{X^{(1)}, X^{(2)}, \ldots, X^{(n_v)}\}$, parameters $\{\lambda_1, \lambda_2, \ldots, \lambda_{n_v}\}$, number of clusters K

Output: Basic matrices $\{U^{(1)}, U^{(2)}, \ldots, U^{(n_v)}\}$, coefficient matrices $\{V^{(1)}, V^{(2)}, \ldots, V^{(n_v)}\}$ and consensus matrix V^*

1. Normalize each view $X^{(v)}$ such that $\left\|X^{(v)}\right\|_1 = 1$
2. Initialize each $U^{(v)}$, $V^{(v)}$ and V^*
3. Repeat

   For v=1 to n_v do

   Repeat

   - Fix V^* and $V^{(v)}$, update $U^{(v)}$:

$$U_{i,k} \leftarrow U_{i,k} \frac{(XV)_{i,k} + \lambda_v \sum_{j=1}^{N} V_{j,k} V^*_{j,k}}{(UV^TV)_{i,k} + \lambda_v \sum_{l=1}^{M} U_{l,k} \sum_{j=1}^{N} V^2_{j,k}}.$$

   - Normalize $U^{(v)}$ and $V^{(v)}$ as:

$$U \leftarrow UQ^{-1},\ V \leftarrow VQ.$$

   - Fix V^* and $U^{(v)}$, update $V^{(v)}$:

$$V_{j,k} \leftarrow V_{j,k} \frac{(X^TU)_{j,k} + \lambda_v V^*_{j,k}}{(VU^TU)_{j,k} + \lambda_v V_{j,k}}.$$

   Until $\|X - UV^T\|_F^2 + \lambda_v \|VQ - V^*\|_F^2$ converge.

   End for

   - Fix $U^{(v)}$ and $V^{(v)}$, update V^*:

$$V^* = \frac{\sum_{v=1}^{n_v} \lambda_v V^{(v)} Q^{(v)}}{\sum_{v=1}^{n_v} \lambda_v} \geq 0.$$

   Until $O = \sum_{v=1}^{n_v} ||X^{(v)} - U^{(v)}(V^{(v)})^T||_F^2 + \sum_{v=1}^{n_v} \lambda_v ||V^{(v)}Q^{(v)} - V^*||_F^2$ converge.

Summary

The algorithm can produce a Interpretable result, where each element in V^* can be

viewed as sum of $P(d|k)^{(v)}$ with the weight of $P(d)^{(v)}$. What's more, V^* can be seen as a network representation, such that we can utilize some classic machine learning algorithm like KNN.

Paper Two

Partial Multi-View Clustering

Motivation

Previous multi-view clustering algorithms assumed that each example appears in all views while the assumption is not always satisfied. Therefore, the authors of this paper propose a new approach, PVC, which works on the dataset with some examples missing in some views. And the authors may possibly be the first ones to study on this issue. They call their work partial multi-view clustering.

Related Theory

Multi-view clustering methods can be roughly divided into two classifications: spectral approaches and subspace approaches.

- Spectral methods are the extension versions of single-view spectral clustering methods fused with similarity measures.
- Subspace methods try to learn a latent subspace across the multiple-views.

NMF

NMF has been introduced in the report of paper one. Its idea of NMF is to find two matrix such: $X \approx UV^T$. U matrix is the basic matrix and V matrix is the coefficient matrix, each row of which represents each data example's latent factor based on matrix U. The above objective function is not convex:

$$\min_{U,V} ||X - UV^T||_F^2,\ s.t. U \geq 0, V \geq 0$$

Main Theory

PVC algorithm defines the problem as:

For a number of two-view data, we learn a common latent subspace to satisfy that different instances of same examples are close and similar instances in the same view are close as well.

We assume that the number of examples presenting and only presenting in both views, the first view, and the second view is c, m and n. Both views: $\hat{\mathbf{X}}^{(1,2)}=[(\mathbf{x}_1^1,\mathbf{x}_1^2);\ldots;(\mathbf{x}_c^1,\mathbf{x}_c^2)]$, the first view: $\hat{\mathbf{X}}^{(1)}=[\mathbf{x}_{c+1}^1;\ldots;\mathbf{x}_{c+m}^1]$, the second view: $\hat{\mathbf{X}}^{(2)}=[\mathbf{x}_{c+m+1}^2;\ldots;\mathbf{x}_{c+m+n}^2]$. And we fuse them into two dataset corresponding to two

different views such that:

$\bar{\mathbf{X}}^{(1)} = [\mathbf{X}_c^{(1)}; \hat{\mathbf{X}}^{(1)}] \in \mathbb{R}^{(c+m)\times d_1}$ $\bar{\mathbf{X}}^{(2)} = [\mathbf{X}_c^{(2)}; \hat{\mathbf{X}}^{(2)}] \in \mathbb{R}^{(c+n)\times d_2}$

Objective function:

The PVC algorithm firstly employs NMF approach onto the above two dataset:

$$\min_{U^{(1)}\geq 0, \bar{P}^{(1)}\geq 0} \|\bar{\mathbf{X}}^{(1)} - \bar{P}^{(1)}U^{(1)}\|_F^2 + \lambda\Omega(\bar{P}^{(1)}),$$

$$\min_{U^{(2)}\geq 0, \bar{P}^{(2)}\geq 0} \|\bar{\mathbf{X}}^{(2)} - \bar{P}^{(2)}U^{(2)}\|_F^2 + \lambda\Omega(\bar{P}^{(2)})$$

Thus we can get $\bar{P}^{(1)} = [P_c^{(1)}; \hat{P}^{(1)}]$ and $\bar{P}^{(2)} = [P_c^{(2)}; \hat{P}^{(2)}]$. Since $P_c^{(1)}, P_c^{(2)}$ should be very close or the same in theoretical analysis, we enforce them equal. Therefore, the objective function is shown as followed:

$$\min_{\{U^{(v)},\bar{P}^{(v)}\}_{v=1}^2} O \equiv \left\| \begin{bmatrix} \mathbf{X}_c^{(1)} \\ \hat{\mathbf{X}}^{(1)} \end{bmatrix} - \begin{bmatrix} P_c \\ \hat{P}^{(1)} \end{bmatrix} U^{(1)} \right\|_F^2 + \lambda\|\bar{P}^{(1)}\|_1 + \left\| \begin{bmatrix} \mathbf{X}_c^{(2)} \\ \hat{\mathbf{X}}^{(2)} \end{bmatrix} - \begin{bmatrix} P_c \\ \hat{P}^{(2)} \end{bmatrix} U^{(2)} \right\|_F^2 + \lambda\|\bar{P}^{(2)}\|_1$$

And we can get the homogeneous feature representation $P = [P_c; \hat{P}^{(1)}; \hat{P}^{(2)}]$ by solving the above model.

Algorithm:

Input: data set $\{X^{(1,2)}, X^{(1)}, X^{(2)}\}$, parameters t, λ

Output: Basis matrices $\{U^{(1)}, U^{(2)}\}$, latent representations $\{P_c, P^{(1)}, P^{(2)}\}$

1. Initialize $U^{(1)}, U^{(2)}$ by using conventional NMF method on both-view data X_c:

$$\min_{U^{(1)},U^{(2)},P_c} O_{init} = \left\|X_c^{(1)} - P_c U^{(1)}\right\|_F^2 + \left\|X_c^{(2)} - P_c U^{(2)}\right\|_F^2 + \lambda\left\|P_c\right\|_1$$

2. Repeat

   Fix $U^{(1)}, U^{(2)}$, update $\{P_c, P^{(1)}, P^{(2)}\}$:

   - $\min_{\hat{P}^{(1)}\geq 0} O(\hat{P}^{(1)}) \equiv \|\hat{\mathbf{X}}^{(1)} - \hat{P}^{(1)}U^{(1)}\|_F^2 + \lambda\|\hat{P}^{(1)}\|_1,$
   - $\min_{\hat{P}^{(2)}\geq 0} O(\hat{P}^{(2)}) \equiv \|\hat{\mathbf{X}}^{(2)} - \hat{P}^{(2)}U^{(2)}\|_F^2 + \lambda\|\hat{P}^{(2)}\|_1$
   - $\min_{P_c} O(P_c) = \left\|X_c^{(1)} - P_c U^{(1)}\right\|_F^2 + \left\|X_c^{(2)} - P_c U^{(2)}\right\|_F^2 + \lambda\left\|P_c\right\|_1$

   Fix $\{P_c, P^{(1)}, P^{(2)}\}$, update $U^{(1)}, U^{(2)}$:

   - $\min_{U^{(1)}\geq 0} O(U^{(1)}) \equiv \|\bar{\mathbf{X}}^{(1)} - \bar{P}^{(1)}U^{(1)}\|_F^2,$
   - $\min_{U^{(2)}\geq 0} O(U^{(2)}) \equiv \|\bar{\mathbf{X}}^{(2)} - \bar{P}^{(2)}U^{(2)}\|_F^2.$

   Until Objective function converges

Summary

The PVC method is based on traditional NMF algorithm working on multi-view clustering. By firstly minimizing the objective function on different two views separately, and then enforce the latent factor corresponding to the same examples equal, to build the PVC model's goal. This idea is of simplicity but produces a powerful result. However, PVC algorithm is still limited in the number of views. It can only work on two-views clustering effectively, but is computationally expensive when working on more-views dataset.