On the Equivalence of PLSI and Projected Clustering

[Position Paper]

Charu C. Aggarwal
IBM T. J. Watson Research Center
Yorktown Heights, NY
charu@us.ibm.com

ABSTRACT

The problem of *projected clustering* was first proposed in the ACM SIGMOD Conference in 1999, and the *Probabilistic Latent Semantic Indexing (PLSI)* technique was independently proposed in the ACM SIGIR Conference in the same year. Since then, more than two thousand papers have been written on these problems by the database, data mining and information retrieval communities, *along completely independent lines of work*. In this paper, we show that these two problems are essentially equivalent, under a probabilistic interpretation to the projected clustering problem. We will show that the EM-algorithm, when applied to the probabilistic version of the projected clustering problem, can be almost identically interpreted as the PLSI technique. The implications of this equivalence are significant, in that they imply the cross-usability of many of the techniques which have been developed for these problems over the last decade. We hope that our observations about the equivalence of these problems will stimulate further research which can significantly improve the currently available solutions for either of these problems.

1. INTRODUCTION

The problem of *projected clustering* (and the closely related problem of *subspace clustering)* were proposed over a decade ago for clustering high dimensional data [8, 1]. The main motivation of this problem formulation was to effectively solve the clustering problem in very high dimensional scenarios in which the data becomes increasingly sparse. Since then, this problem has been explored extensively by the database and data mining community in the context of a wide variety of scenarios and problem domains [6]. The projected clustering problem was first proposed in the database community, and much of the initial work in this area was performed within the core database conferences such as SIGMOD [1, 2, 5, 8].

At approximately the same time as the publication of the projected clustering work [1], the PLSI technique was independently proposed in the information retrieval community [15] for clustering and dimensionality reduction of text. This also led to a larger interest in the newly defined problem of *topic modeling*. A variety of subsequent methods for topic modeling such as LDA [10] have found very wide popularity and success for soft text clustering. We would like to emphasize that PLSI is a *technique*, whereas topic modeling is a *problem*. However, the interest and awareness of this very important problem arose out of the original PLSI paper [15]. Most of the work on PLSI and its variants has remained restricted to the information retrieval community, with a primary focus on text data. In fact, the original paper on PLSI was positioned [15] as an alternative to the latent-semantic indexing approach [11] for dimensionality reduction of documents, rather than providing a clustering solution. Subsequently, the importance of the broader problem formulation has been extensively exploited for soft clustering by the information retrieval community [10].

The differences between PLSI and projected clustering would seem to be significant at first sight. Most projected clustering problems are naturally defined as *deterministic* problems, in which cluster membership and dimension membership is absolute. On the other hand, PLSI is a soft variation, which allows soft membership of documents and words within the different clusters. Furthermore, the EM approach of PLSI implicitly uses the fact that most documents contain a small fraction of the lexicon, and have small non-negative frequencies. On the other hand, projected clustering is generally defined for highly ordered and quantitative attributes, which may be either positive or negative and the clusters are defined by the wide variations and correlations across these different attribute values. Straightforward applications of probabilistic methods to projected clustering do not necessarily yield PLSI. In fact, some probabilistic methods [17] have been proposed for projected clustering, but

are largely unrelated to PLSI, because of significant differences in the underlying data representations. This is because there are a variety of different ways to formulate projected clustering with a probabilistic approach. Finally, the two problems have largely been explored by two completely disjoint communities of researchers, and this has also lead to an artificial separation between these different problems.

On the other hand, the two problems also share a number of common characteristics. For example, both formulations explore the duality of points and dimensional clustering behavior simultaneously in order to determine the underlying patterns. As we will see later, a careful probabilistic modeling of the projected clustering problem, and an EM-based solution naturally leads to an algorithm which is essentially equivalent to PLSI, with an appropriate mapping between the feature space of the two problems. Furthermore, we will also explore the co-clustering model [13], the matrix factorization model [20], and the relationships of these models to both problems. Co-clustering and matrix factorization can essentially be considered deterministic versions of topic modeling, in that they provide a simultaneous understanding of the duality between documents and words, though not necessarily probabilistically. In this context, it is somewhat surprising that most of the work on these different clustering models are generally performed independently of one another, with little exploration and understanding of the relationships between the different variants.

The implications of this equivalence are significant for all these different models for clustering. Most projected clustering methods have been designed in an *absolute sense*, with a hard definition of the data points and the underlying dimensions. On the other hand, a probabilistic version of the problem lends itself to *immediate use of a decade of work in the information retrieval community*. Similarly, there is significant amount of work on pattern-based variations of projected clustering, which can be almost directly used by the information retrieval community for deterministic versions of topic modeling. Of course, significant effort may also be required in order to cross-test these methods across domains, though it is very likely that many of the methods in either domain will be useful for the other. A detailed cross-testing of the (decade of) methods across the two domains is beyond the scope of this position paper. The main purpose of this position paper is not to propose a specific algorithm for either problem, but to show the equivalence between the two problems. This is likely to stimulate a further direction of exploration for both communities.

This paper is organized as follows. In the next section, we will study the probabilistic version of the projected clustering problem. We will propose a probabilistic EM-algorithm for this problem. In section 3, we will interpret this solution in the context of the PLSI technique. We will also explore other variations of the PLSI method, which are related to this technique. In section 4, we will provide a discussion of the implications of the relationship between these different problems.

2. PROJECTED CLUSTERING: DEFINITION AND PROBABILISTIC VARIATION

We start off with the notations and definitions. We assume that we have a data set $\mathcal{D}$ with N records, and a dimensionality of d. We assume that the records in $\mathcal{D}$ are denoted by $\overline{X_1} \ldots \overline{X_N}$. The values on the individual dimensions of the j-th data point $\overline{X_j}$ are denoted by $(x_{j1} \ldots x_{jd})$.

The core idea in projected clustering is that the underlying data is sparse because of the curse of dimensionality [7, 9, 14]. In such cases, distance functions lose their discriminative behavior [5] in full dimensionality, and therefore meaningful clusters cannot always be defined in full dimensionality. Therefore, the problem of projected clustering is defined in order to simultaneously determine the clusters and the cluster-specific dimensions from the underlying data. The idea is that *locally* relevant dimensions can be helpful in defining clusters, because of the differential nature of the dimension relevance in different data localities. The output of a projected clustering algorithm is twofold:

- a $(k+1)$-way partition $\{\mathcal{C}_1, ..., \mathcal{C}_k\}$ of the data, such that the points in each partition element form a cluster.
- a possibly different orthogonal set $\mathcal{E}_i$ of dimensions for each cluster $\mathcal{C}_i$, $1 \leq i \leq k$, such that the points in $\mathcal{C}_i$ cluster well in the subspace defined by the dimensions in $\mathcal{E}_i$.

In order to define the probabilistic variation of the projected clustering problem, we will use kernel density estimation in order to define dimension-specific data localities. These dimension-specific localities are useful for defining the influence of each data point in different dimension-specific localities of the data in terms of a kernel density value.

Let μ_i and σ_i be the mean and standard deviation of the data along each dimension i. For each dimension i, we define $(m+1)$ equally spaced *anchor points* located at $\mu_i - \frac{3 \cdot m \cdot \sigma_i}{m}$, $\mu_i - \frac{3 \cdot (m-2) \cdot \sigma_i}{m}$

$\ldots$ $\mu_i + \frac{3\cdot(m-2)\cdot\sigma_i}{m}$, $\mu_i + \frac{3\cdot m\cdot\sigma_i}{m}$. In general, for the ith dimension, and rth dimension-specific locality, we define $Z(i,r)$ as follows:

$$Z(i,r) = \mu_i + \frac{3\cdot(m-2\cdot r)\cdot\sigma_i}{m} \qquad r \in \{0\ldots m\} \tag{1}$$

We note that the choice of the location of these anchor points ensures that the most relevant data space in $[\mu_i - 3\cdot\sigma_i, \mu_i + 3\cdot\sigma_i]$ (where most of the data is likely to be statistically located) also has well spaced anchor points in it. Correspondingly, we define the kernel density estimate $K(j,i,r)$ of the jth data point $\overline{X_j}$ along the rth locality in the ith dimension (denoted by $Z(i,r)$) as follows:

$$K(j,i,r) = \max\{e^{-\frac{2\cdot(x_{ji}-Z(i,r))^2}{(6\cdot\sigma_i/m)^2}} - \epsilon, 0\} \tag{2}$$

We note that the exponential term in the aforementioned expression is an un-normalized variation on the standard kernel density estimation technique [18], which is commonly used for density analysis. We have not used the constant multiplicative factors in the density expression for simplicity, and also because these factors do not affect the underlying computations or the result of the approach. The specific choice of the denominator in the exponent term (un-normalized bandwidth) is picked to ensure that the values of $K(j,i,r)$ will be significantly positive (in a given dimension and record $\overline{X_j}$) for only one or two anchors $Z(i,r)$. Specifically, two consecutive anchor points are $6\cdot\sigma_i/m$ units apart along dimension i, and therefore the square of this is used in the denominator of the value in the exponent. This ensures that the exponential term in the kernel density $K(j,i,r)$ is significant only for one or two neighboring anchor points of $\overline{X_j}$ along dimension i. The density values drop off exponentially for the other anchor points with increasing distance to that record. Therefore, by using a small value ϵ as a minimum threshold, it is possible to ignore very small values on the density and explicitly set them to 0. This is achieved in Equation 2, by subtracting the small value ϵ from every density, and setting any negative value to 0.

Next, we will define the probabilistic version of the projected clustering problem in terms of the dimension specific localities Z and the corresponding kernel function K.

Definition 1 (Prob. Proj. Clustering). *Given a data set $\mathcal{D}$, which is expressed in terms of dimension specific localities Z, and corresponding kernel densities K, determine a generative model for the data set with k partitions, in terms of the following parameters:*

- *Each data point $\overline{X_j}$ is associated with a partition with a probability that is learned in a data-driven manner. The sum of the probabilities over different partitions is 1.*
- *Each dimension-specific locality $Z(i,j)$ is associated with a partition with a probability that is learned in a data-driven manner. The sum of the probabilities over different partitions is 1.*

We note that this is a soft version of the projected clustering problem in which probabilities are associated with point-specific and dimension-specific membership. Furthermore, the probabilities are associated with dimension-specific *localities* rather than the dimensions themselves. If desired, it is possible to assign each data point to the partition with the highest probability of membership in order to create a strict partition. Similarly, it is possible to use a threshold on the probabilities which associate clusters with dimension locality. This will provide a set of the most relevant dimensions of projection *together with* the corresponding localities. In practice, the localities included for a particular partition are likely to be contiguous to one another (because of the natural smoothness of data distributions within cluster partitions). Furthermore, localities from many dimensions will not be included at all, when strong thresholds are used for picking cluster-specific dimension localities. This is almost identically a solution to deterministic projected clustering. Thus, by using thresholding, it is also possible to convert a solution to the probabilistic projected clustering problem into a complete solution of the deterministic version of the problem. Furthermore, we note that the probabilistic model allows for different levels of overlap and partitioning between clusters, depending upon how the soft clustering is converted into a hard one. For example, by using thresholds on the assignment probability (instead of assignment by largest probability value), it is possible to allow point overlaps among the different clusters. Similarly, it is also possible to force strict partitioning on the sets of dimension localities.

The afore-mentioned formulation requires us to learn point- and dimension-specific probabilities in a data-driven manner. This can be naturally solved with the use of the EM algorithm. In order to perform the modeling, a generative model is assumed for the different records in the database. We define random variables $Q_1 \ldots Q_k$ corresponding to the k different partitions, and each partition has its own set of generative probabilities for the dimension-specific localities. The probability $P(Z(i,r)|Q_s)$

represents the probability that the dimension-specific locality $Z(i,r)$ is included in the s-th partition. From an intuitive perspective, a high value of $P(Z(i,r)|Q_s)$ implies that data points which are close to this dimension-specific locality are very relevant to the partition Q_s. Correspondingly, such data points will have high non-zero density value of $K(j,i,r)$.

Similarly, the expression $P(Q_s|\overline{X_j})$ represents the probability that the s-th partition is most relevant, when the generated record happens to be $\overline{X_j}$. Clearly, these are the probabilities that need to be learned in a data-driven manner. These will also directly yield the probability distribution parameters which define the solution to the projected clustering problem.

Then, we can also express the probability of a dimension-specific locality $Z(i,r)$ occurring within the record $\overline{X_j}$ as follows with the use of this generative model:

$$P(Z(i,r)|\overline{X_j}) = \sum_{s=1}^{k} P(Z(i,r)|Q_s) \cdot P(Q_s|\overline{X_j}) \quad (3)$$

The above relationship is key to the EM algorithm, because we also have the data, which tells us the *true instantiations* of $P(Z(i,r)|\overline{X_j})$. Therefore, we will define matrices for the point- and dimension-specific probability parameters and attempt to learn them with the EM algorithm.

Thus, for each term $Z(i,r)$ and record $\overline{X_j}$, we can generate a $N \times [(m+1) \cdot d]$ matrix of probabilities, which represent the probability that the dimension-specific locality, $Z(i,r)$ is relevant to (or has a high kernel density estimate for) record $\overline{X_j}$. The rows in this matrix corresponds to the N different records, and the number of columns corresponds to the number of dimension-specific localities $(m+1) \cdot d$. The $[i*(m+1)+r]$-th column of this matrix corresponds to the probability for $Z(i,r)$. We also assume that we have a matrix of similar size, which provides us the *actual data* about the kernel densities directly from the underlying database $\mathcal{D}$. We refer to this as the kernel density matrix Y. For $l = i*(m+1)+r$, the entry $Y(j,l)$ is equal to the kernel density value $K(j,i,r)$. Thus, the maximum likelihood estimation process can be used, by maximizing the product of the dimension-specific localities (with non-zero kernel density), which are observed in each record in the database $\mathcal{D}$ containing the different records $\overline{X_j}$.

Specifically, the maximum likelihood estimation algorithm maximizes the product of the generative probabilities of dimension-specific localities, that are actually observed to be of non-zero value in the underlying kernel density matrix. As is the case for the maximum-likelihood approach in EM algorithms, we would like to maximize the logarithm of this estimated probability. This can be expressed as a weighted sum of the logarithm of the terms on the left hand side in Equation 3. The weight of the (j,l)th term is the density value $Y(j,l)$. This is a constrained optimization problem. Specifically, from the EM framework, we need to optimize the value of the log likelihood probability $\sum_{i,j,r} Y(j,l) \cdot \log(P(Z(i,r)|\overline{X_j}))$ subject to the constraints that the probability values over each of the point-specific and dimension-specific values must sum to 1:

$$\sum_{i,r} P(Z(i,r)|Q_s) = 1 \quad \forall Q_s \quad (4)$$

$$\sum_{j} P(Q_s|\overline{X_j}) = 1 \quad \forall \overline{X_j} \quad (5)$$

The value of $P(Z(i,r)|\overline{X_j})$ in the objective function can be expanded and expressed in terms of the model parameters with the use of Equation 3. We note that a Lagrangian method can be used to solve this constrained problem. The Lagrangian solution essentially leads to a set of iterative update equations for the corresponding parameters which need to be estimated. It can be shown that these parameters can be estimated [12] with the iterative update of two matrices $[P_1]_{k \times N}$ and $[P_2]_{d \cdot (m+1) \times k}$ containing the point-specific probabilities and dimension-specific probabilities respectively for the clustering process. We start off by initializing these matrices randomly, and normalize each of them so that the probability values in their columns sum to one. Then, we iteratively perform the steps on each of P_1 and P_2 respectively, as discussed in Figure 1. The first step is the E-step, which updates P_1 by computing the expected probabilities of membership of a point in a cluster. This is done by using the dimension-specific localities in the point, and

Algorithm *ProjectedClusteringEM*
begin
Initialize matrices $P1$ and $P2$;
repeat
(E-Step) Update P_1 to correspond to probabilities of assignment of records to clusters;
Normalize each column of P_1 to sum to 1;
(M-Step) Compute P_2 based on the weighted frequency of each dimension-specific locality in each cluster;
Normalize each column of P_2 to sum to 1;
until convergence;
end

Figure 1: Application of the EM Framework for Probabilistic Projected Clustering

the matrix P_2, which provides the probability distribution of the dimension specific localities in that cluster. The E-Step may use a variety of probability models (eg. bernoulli model) for computing cluster assignment probabilities from the dimension-specific localities present in records. The second step is the M-step, which optimizes the parameters, assuming the current assignments. This corresponds to computing the probability of the dimension-specific locality in each cluster. Thus, this iterative two-step process continuously updates the matrices P_1 and P_2, which provides the final output of the algorithm.

3. INTERPRETATION AS PLSI

Upon examining the iterative update equations of Figure 1 in more detail, and comparing to the PLSI algorithm in [15], it becomes evident that *the steps in the two algorithms are virtually identical*. The main difference is that the densities of dimension specific localities are used to perform the updates in the EM-algorithm instead of the word-specific frequencies in the PLSI algorithm. More generally, the probabilistic projected clustering algorithm becomes identical to PLSI when words are interpreted as dimension-specific localities.

This is quite logical because both algorithms are derived from an EM-based approach, the kernel density-based transformation provides a feature representation which is friendly to PLSI. We note that such an approach also opens up other possibilities for projected clustering with the use of other methods such as *co-clustering* and *matrix-factorization* on the representation.

- We can apply matrix-factorization [20] to the kernel density matrix Y in order to yield the k projected clusters. Specifically, let U be a $N \times k$ non-negative matrix, and V is a $d \cdot (m+1) \times k$ non-negative matrix. Then, we can factorize the matrix Y as follows in order to yield the point- and dimension components U and V:

$$Y \approx U \cdot V^T \tag{6}$$

  The columns of V provide the k-different basis-vectors for the dimension-specific localities for each of the clusters. These can also be regarded as k (non-negative) basis vectors which correspond to the k different clusters. As in the case of PLSI, one can use thresholding on these basis vectors to decide which dimension-specific locality is relevant to which cluster. Specifically, a basis vector has $(m+1) \cdot d$ components, and the value of each component is an indicator of its relevance to that cluster. Therefore, by thresholding out the low values, the relevant dimensions may be determined.

  Similarly, the $N \times k$ matrix U provides information about the level of relevance of the N different data points to each of the k clusters. A strict partition may be obtained by assigning each data point to the cluster for which it has the highest relevance. Thus, it is possible to use non-negative matrix factorization for projected clustering, an approach which has rarely been used in the literature.

- Co-clustering [13] is defined on sparse non-negative matrices for clustering both rows and columns simultaneously. A wide variety of graph-based and information-theoretic techniques are available for solving this problem. The kernel-based representation can be used directly in conjunction with any co-clustering approach for this problem. Specifically, co-clustering can be applied to the $N \times d \cdot (m+1)$ matrix Y in order to provide a simultaneous clustering of the points and dimension-specific localities. This can be used in order to re-construct the projected clusters effectively. Yet, such methods have been rarely used for projected clustering of multi-dimensional (quantitative) data, and have largely been restricted to sparse matrices such as text.

The work in [19] explores the relationship of matrix factorization models to PLSI. However, it does not explore the relationship of the projected clustering problem to PLSI. Furthermore, all models discussed in [19] are implicitly designed for sparse non-negative matrices.

4. POTENTIAL AND RESEARCH DIRECTIONS

The implications of these equivalence observations are significant for both communities. First of all, this problems are explored independently by the different communities over a decade, and a huge number of algorithms have been constructed for different variations of these problems. For example, the original PLSI technique has been extended to more advanced techniques such as LDA [10], or other dynamic methods for topic modeling in streaming scenarios [4]. Instead of applying a probabilistic EM framework for projected clustering, it is possible to use any of these more advanced methods for the problem. While EM algorithms can also be directly applied to projected clustering, the parameter fitting process does not behave well with increasing dimensionality for general multidimensional data.

The kernel-density based transformation creates a representation which enhances the locality specific behavior of distances between records. This is known to be effective for the high dimensional case, as suggested in section 4 of [5]. Furthermore, any of these methods can be made immediately available for different contexts and scenarios such as projected clustering of high dimensional data streams [3]. We also showed that numerous other techniques such as co-clustering and matrix-factorization can also be used in the context of this framework.

On the other hand, numerous variations of projected clustering such as pattern-based clustering are closely related to the techniques designed for co-clustering and matrix factorization. In particular, probabilistic algorithms for the bi-clustering problem [16] and for pattern-based clustering can be adapted to the information retrieval domain, to achieve similar goals of examining the duality between words and clusters. The use of these equivalences to further test the potential of these different problems is likely to be a fruitful direction of work for both domains.

We also note that many general versions of both problems are not equivalent to one another, and therefore cannot be captured by either framework. For example, the generalized projected clustering [2] problem defines the relevant dimensions of projection in arbitrary dimensions in the data space. Such scenarios cannot be easily modeled with PLSI- or matrix-factorization models, because the latter models implicitly work with axis-parallel representations. Similarly, projected clustering techniques cannot achieve the same goal as more sophisticated topic modeling methods such as LDA [10]. Nevertheless, such techniques in either domain also suggest the possibility of developing more generalized methods in the other domain. Therefore, it is evident that significant similarities exist between the problems at the formulation level. These should therefore, be leveraged for advancement of the techniques in both fields.

5. REFERENCES

[1] C. C. Aggarwal, C. Procopiuc, J. Wolf, P. Yu, J.-S. Park. Fast Algorithms for Projected Clustering. *ACM SIGMOD Conference*, 1999.

[2] C. C. Aggarwal, P. S. Yu. Finding Generalized Projected Clusters in High Dimensional Space, *ACM SIGMOD Conference*, 2000.

[3] C. C. Aggarwal, J. Han, J. Wang, P. Yu. A Framework for Projected Clustering of High Dimensional Data Streams, *VLDB* , 2004.

[4] C. C. Aggarwal, C. Zhai. A Survey of Text Clustering Algorithms, *Mining Text Data*, Springer, 2012.

[5] C. C. Aggarwal. Re-designing Distance Functions and Distance-based Applications for High Dimensional Data, *ACM SIGMOD Record*, March, 2001.

[6] C. C. Aggarwal, C. Reddy. Data Clustering: Algorithms and Applications, *CRC Press*, 2013.

[7] C. C. Aggarwal, A. Hinneburg, D. Keim. On the Surprising Behavior of Distance Metrics in High Dimensional Space, *ICDT*, 2001.

[8] R. Agrawal, J. Gehrke, P. Raghavan, D. Gunopulos. Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications, *SIGMOD Conference*, 1998.

[9] K. Beyer, J. Goldstein, R. Ramakrishnan, U. Shaft. When is nearest neighbor meaningful? *ICDT Conference*, 1999.

[10] D. Blei, A. Ng, M. Jordan. Latent Dirichlet allocation, *Journal of Machine Learning Research*, 3: pp. 993–1022, 2003.

[11] S. T. Deerwester, S. T. Dumais, G. Furnas, R. Harshman. Indexing by Latent Semantic Analysis, *JASIS*, 1990.

[12] A. P. Dempster, N. M. Laird and D. B. Rubin. "Maximum Likelihood from Incomplete Data via the EM Algorithm", *Journal of the Royal Statistical Society*, B, vol. 39, no. 1, pp. 1–38, 1977.

[13] I. Dhillon. Co-clustering Documents and Words using bipartite spectral graph partitioning, *ACM KDD Conference*, 2001.

[14] A. Hinneburg, C. Aggarwal, D. Keim. What is the nearest neighbor in high dimensional space? *VLDB Conference*, 2000.

[15] T. Hoffman. Probabilistic Latent Semantic Indexing, *ACM SIGIR Conference*, 1999.

[16] S. C. Madeira, A. L. Oliveira. Bi-clustering Algorithms for Biological Data Analysis: A Survey, *IEEE/ACM Transactions on Computational Biology*, 1(1), pp. 24–35, 2004.

[17] G. Moise, J. Sander, M. Ester. P3C: A Robust Projected Clustering Algorithm, *ICDM Conference*, 2006.

[18] B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, 1986.

[19] A. Singh, G. Gordon. A Unified View of Matrix Factorization Models, *ECML/PKDD Conference*, 2008.

[20] W. Xu, X. Liu, Y. Gong. Document Clustering based on non-negative matrix factorization, *ACM SIGIR Conference*, 2003.