

IFD: Iterative Feature and Data Clustering

Tao Li*
Computer Science Dept.
University of Rochester
Rochester, NY 14627-0226
taoli@cs.rochester.edu

Sheng Ma
IBM T.J. Watson Research Center
Hawthorne, NY 10532
shengma@us.ibm.com

ABSTRACT

In this paper, we propose a new clustering algorithm, *IFD*¹, based on a cluster model of data coefficients D and feature coefficients F . The coefficients denote the degree (or weights) of the data and features associated with the clusters. Clustering is performed via an iterative optimization procedure to **mutually reinforce** the relationships between the coefficients. The **mutually reinforcing** optimization exploits the duality of the data and features and enable a simultaneous clustering of both data and features. We have shown the convergence property of the clustering algorithm and discussed its connections with various existential approaches. Extensive experimental results on both synthetic and real data sets show the effectiveness of *IFD* algorithm.

Keywords: *IFD*, clustering, mutually reinforcing, data and feature coefficients, convergence

1. INTRODUCTION

The problem of clustering data arises in many disciplines and has a wide range of applications. Intuitively, the clustering problem can be described as follows: Let W be a set of n data points in a multi-dimensional space. Find a partition of W into classes such that the points within each class are *similar* to each other. Generally clustering problems are determined by four basic components: a) the (physical) representation of the given data set; b) the formal model for describing the generation of the data set; c) The criterion/objective function which the clustering solutions should aim to optimize; d) The optimization procedure. For a given data clustering problem, the four components are tightly coupled. The formal model is induced from the physical representation of the data, the formal model along with the objective function determines the clustering capability and the optimization procedure decides how efficiently and effectively the clustering results can be obtained. The choice of the optimization procedure depends on the first three components.

*The work is done during the author's 2003 summer internship at IBM T.J. Watson Research Center.

¹*IFD* stands for **I**terative **F**eature and **D**ata clustering.

The clustering problem has been studied extensively in machine learning, databases, and statistics from various perspectives and with various approaches and focuses. However, many methods suffer from serious drawbacks due to the following reasons. First, some methods based on the models that make simple assumptions on the data distributions e.g., Gaussian mixture models. Second, the criterion/objective function adopted by most methods are based on the distance functions between sample points. Hence most algorithms do not work efficiently in high dimensional spaces due to the *curse of dimensionality*. It has been shown that in a high dimensional space, the distance between every pair of points is almost the same for a wide variety of data distributions and distance functions [2]. Many feature selection techniques have been applied to reduce the dimensionality of the space. However, as demonstrated in [1], the correlations among the dimensions are often specific to data locality; in other words, some data points are correlated with a given set of features and others are correlated with respect to different features. As pointed out in [6], all methods that overcome the dimensionality problems use a metric for measuring neighborhoods, which is often implicit and/or adaptive. Third, the cluster interpretation and validity are also important concerns. Last but not least, efficient optimization procedures are also needed for better convergence.

In this paper, we propose a new clustering algorithm, *IFD*, which models cluster generation as two sets of coefficients: data coefficients D and feature coefficients F . The data (respectively, feature) coefficients denote the degree to which the corresponding data (respectively, feature) is associated with the clusters. The data clustering task is carried out by iteratively computing the two sets of coefficients based on **mutually reinforcing** updating rules derived from the objective criterion. The **mutually reinforcing** rules exploit the duality of the data and features, thus enable a simultaneous clustering of both data and features. By generating an explicit feature assignments, *IFD* produces interpretable descriptions of the resulting clusters as an added bonus. In addition, by iteratively reinforcing updating, *IFD* performs an implicit adaptive feature selection at each iteration and flexibly measures the distances between data points. Therefore it works well for high-dimensional data. We have proved the convergence property of *IFD* algorithm and conducted extensive experiments to show its effectiveness.

2. IFD CLUSTERING

In this section, we first describe the cluster model used in *IFD* clustering, followed by the formulation from matrix perspective. Then we derive the optimization criterion based on matrix norm minimization. Finally we present our **mutual reinforcing updating** rules as an efficient approximation to the optimization criterion.

2.1 The Cluster Model

We assume that the data is represented in a “flat-table” format where each row corresponds to a sample point and each column corresponds to an attribute. We first illustrate our model using binary data and will present the extensions to categorical and numerical data later. For binary dataset, each entry of the table is either 0 or 1.

The cluster model is determined by two sets of coefficients: data coefficients $D = (d_{ij})$ and feature coefficients $F = (f_{ij})$. The data (respectively, feature) coefficients denote the degree to which the corresponding data (respectively, feature) is associated with the clusters. Suppose the dataset W has n instances, having m features each. Then W can be viewed as a subset of R^m as well as a member of $R^{n \times m}$. Suppose W has k clusters. Then the data (resp. feature) coefficients can be represented as a matrix $D_{n \times k}$ (resp. $F_{m \times k}$) where d_{ij} (f_{ij}) indicates the degree to which data point i (resp. feature i) is associated with cluster j .

Note that clustering solutions can be easily obtained once the data or feature coefficients are obtained. Basically, D induces a hard clustering assignment by assign data point i to cluster x if $x = \operatorname{argmax}_j d_{ij}$. We also maintain the invariant that the weights of each data (feature) are normalized so their sums are equal to 1: $\sum_j d_{ij} = 1, \forall i = 1, 2, \dots, n, (\sum_j^k f_{ij} = 1, \forall i = 1, 2, \dots, m)$.

2.2 Matrix Perspective

Given a binary dataset $W = (w_{ij})_{n \times m}$ where $w_{ij} = 1$ if the j -th feature is present in the i -th instance and $w_{ij} = 0$ otherwise. Let \hat{W} be the normalized W , i.e., $\hat{W} = W \times L$ where L is a diagonal matrix whose (i, i) -th element is the norm of W 's i -th row. Each entry \hat{w}_{ij} of \hat{W} can then be interpreted as the probability that j -th feature is present in i -th instance. By abusing the notation, we will still use W to denote the normalized W .

Given representation (D, F) , basically, D denotes the likelihood of data points associated with clusters and F indicates the feature representations of clusters. The ij -th entry of DF^T then indicates the possibility that the j -th feature will be present in the i -instance, computed by the dot product of the i -th row of D and the j -th row of F . Hence after normalization, DF^T can be interpreted as the approximation of the original data W . Our goal is then to find a D and F that minimizes the squared error between W and its approximation DF^T .

$$\arg \min_{D, F} O = \frac{1}{2} \|W - DF^T\|_F^2, \quad (1)$$

where $\|X\|_F$ is the Frobenius norm of the matrix X , i.e., $\sqrt{\sum_{i,j} x_{ij}^2}$. With the formulation, we transform the data clustering problem into the computation of D and F that minimize the criterion O .

2.3 Optimization

The objective function can be rewritten as

$$\begin{aligned} O &= \frac{1}{2} \|W - DF^T\|_F^2 \\ &= \frac{1}{2} \operatorname{tr}((W - DF^T)(W - DF^T)^T) \\ &= \frac{1}{2} (\operatorname{tr}(WW^T) - 2\operatorname{tr}(WFD^T) + \operatorname{tr}(DF^T FD^T)) \end{aligned}$$

The above derivations used the matrix properties ² $\|A\|_F^2 = \operatorname{tr}(A^T A)$ and $\operatorname{tr}(AB) = \operatorname{tr}(BA)$. Although the function $\frac{1}{2} \|W - DF^T\|_F^2$ is not convex in both variables together (the Hessian is not positive definite), it is convex in D only or in F only. Therefore, the objective function can be minimized (local minima) by alternatively optimize one of D or F while fixing the other. Fixing D , O becomes a quadratic form of F which can be denoted as $O(F)$. Similarly, fixing F , O becomes a quadratic form of D which can be denoted as $O(D)$. Observe that

$$\frac{\partial O}{\partial D} = -WF + DF^T F \quad (2)$$

$$\frac{\partial O}{\partial F} = -W^T D + FD^T D \quad (3)$$

The above derivatives in Equation 2 and 3 would lead to the optimization via $D = WF(F^T F)^{-1}$ and $F = W^T D(D^T D)^{-1}$.

However, the computation of the the inverse matrix is usually expensive. In addition, if the data is ill-posed, the computation also faces the inverse problem: a small perturbation in the matrices may result in a big difference in its inverse. So it is natural to look for approximation methods.

2.4 Mutual Reinforcing Updating

Let's take a closer look at the two updating rules: $F = W^T D(D^T D)^{-1}$ and $D = WF(F^T F)^{-1}$. D (resp. F) indicates the weights of each point (resp. feature) associated with each cluster. Hence $D^T D$ (resp. $F^T F$) indicate the weight comparisons between the clusters. In other words, the ij -th entry of $D^T D$ equals to the dot product of the i -th column of D (corresponding to cluster i) with the j -th column of D (corresponding to cluster j). If the columns of F , i.e., the features associated with different clusters are orthogonal, then $F^T F$ becomes a diagonal matrix and doesn't change the directions of optimization. Similar situation holds for $D^T D$. So orthogonalization enables easy computation. On the other hand, for most clustering tasks, it is customary to assume that the features or data belonging to different clusters do not overlap ³. By imposing orthogonal requirements, we would obtain two simplified updating rules

$$F = W^T D \quad (4)$$

$$D = WF \quad (5)$$

In addition, we note that these simplified rules has a natural interpretation analogous to the HITS ranking algorithm [9]. Basically, the optimizing rules show the **mutually reinforcing relationship** between the data and the features. It is natural to express as follows: if a feature f is shared by many points that have high weights associated with a cluster c , then feature f has a high weight associated with c . On the other hand, if a data point d is shared by many points that have high weights associated with a cluster c , then the data point d has a high weight associated with c . To find the desired solution for D and F , one then apply the two rules in an alternative fashion and see whether a fixed point is reached. We have shown in Section 4 that the optimization procedure converges.

3. ALGORITHM DESCRIPTION

² $\operatorname{tr}(A)$ is the trace of matrix A .

³Extending the model for overlapped clustering problems is one of our future work.

The clustering procedure of clustering is described as Algorithm 1. Clustering results can be naturally obtained from D . In this procedure, we maintain that the features (data) belonging to different clusters do not overlap. This amounts to require that each row of the data (feature) coefficients has at most one entry with value 1 and all the rest are zeros. The mutual reinforcing update rules derived in Equation 4 and Equation 5 are then used in our procedure. The mutually reinforcing updating rules are realized in Step 2.1 and step 2.3.

The mutually reinforcing updating rules derived in Equation 4 and Equation 5 implicitly assume that all the features/data are beneficial to clustering assignments. In real applications, however, not all the features are characteristic features for the clusters and some of them are outliers. Similarly, there exist data points which do not belong to any cluster. Hence, step 2.1 and step 2.3 also perform outlier detections. Intuitively, a feature is a characteristic feature for a cluster if its associated degree to the cluster is significantly greater than that to any other cluster. Similarly, a data point belongs to a cluster if its associated degree to the cluster is significantly greater than that to any other cluster. In our implementation, if a feature (or data point) has similar associated degrees to multiple clusters, (i.e., the differences between the associated degrees are less than some predefined value), then it is viewed as an outlier at current stage. The identified outliers at one step do not contribute to the mutual reinforcing updating for the adjacent step⁴. The post-processing step performs normalization and thresholding operations to make the coefficients satisfying the requirements for orthogonal iterations. In particular, thresholding operations make sure that each row of the data (feature) coefficients has at most one entry⁵ with value 1 and all the rest are zeros.

Algorithm 1 *IFD*: clustering procedure

Input: (data points: $W_{n \times m}$, # of classes: k)

Output: D : cluster assignment;

begin

1. **Initialization:**

1.1 Set Initial feature coefficients F^0 ;

1.2 Set $t = 1$.;

2. **Iteration:**

begin

2.1 Compute D^t from F^{t-1} ;

2.2 Post-processing;

2.3 Compute F^t from D^t ;

2.4 Post-processing;

2.5 $t = t + 1$, repeat until converge.

end

3. **Return** $D = D^t$;

end

To summarize, the four components, as discussed in Section 1, of *IFD* clustering algorithm are listed in Table 1.

4. CONVERGENCE ANALYSIS

Basically, the objective function O is non-increasing under the updating rule derived from Equation 2 and Equation 3 [11]. We

⁴This is typically done by zeroing the corresponding rows in the coefficients.

⁵Usually the entry with maximal associated degree.

Four Components	<i>IFD</i>
Data Representation	W : flat table
Data Generation Model	(D, F) : Data and Feature Coefficients
Optimization Criterion	Matrix Perspective (Or Maximum Likelihood)
Optimization Procedure	Iterative Mutual Reinforcing Data and Feature Assignments

Table 1: *IFD* Summary.

will show that under certain conditions, our optimization procedure would converge to the subspace spanned by the k dominant eigenvectors.

Suppose the sizes of the k clusters are n_1, n_2, \dots, n_k respectively and $n = \sum_{i=1}^k n_i$. Note that WW^T is a $n \times n$ matrix and each entry computes the number of matches (i.e., the denominator of the similarity coefficients) between data points. We could then normalize the entries of WW^T such that the sum of each row equals to one. To simplify the analysis, we ordered the points in W according to which cluster they are in, so that all points belonging to the first cluster appear first and the second cluster next, etc. The permutation of the matrices does not change the spectral properties. Since data points inside each cluster are similar to each other while they are quite different among clusters, WW^T can be regarded as the addition of two matrices: $WW^T = L + E$ where $L = \begin{pmatrix} X_1 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & 0 & X_k \end{pmatrix} \in R^{n \times n}$, $X_i = \begin{pmatrix} 1 & \dots & 1 \\ \vdots & \vdots & \vdots \\ 1 & \dots & 1 \end{pmatrix} \in R^{n_i \times n_i}$ and $E \in R^{n \times n}$ is a matrix with a small value in each entry, i.e., $E = O(\epsilon)$.

Denote $S = WW^T$, set:

$$S(\epsilon) = S(0) + \epsilon S^{(1)} + \epsilon^2 S^{(2)} + \dots,$$

where $S(0) = L$ is the unperturbed part of S . It then follows from perturbation theory [8, 3] that the spectrum of $S(\epsilon)$ can be divided into two parts:

1. the Perron cluster including the Perron root $\lambda_1 = 1$ and the $k - 1$ eigenvalues $\lambda_2(\epsilon), \dots, \lambda_k(\epsilon)$ approaching 1 for $\epsilon \rightarrow 0$.
2. the remaining part of the spectrum, bounded away from 1 for $\epsilon \rightarrow 0$.

It can then be shown that our optimization procedure performs an approximate orthogonal iteration to compute the k largest eigenvalues of WW^T . Due to the space limit, we omit the detailed proof here.

5. EXPERIMENTS

We have applied *IFD* on a variety of datasets. Due to space limit, we only present experimental results on the **CSTR** dataset. **CSTR** is the dataset of the abstracts of technical reports published in the Department of Computer Science at the University of Rochester between 1991 and 2002. The TRs are available at <http://www.cs.rochester.edu/trs>. It has been used in [13] for text categorization. The dataset contained 476 abstracts, which were divided into four research areas: Natural Language Processing(NLP), Robotics/Vision, Systems, and Theory. We represent the abstracts

using binary vector-space model where each document is a binary vector in the term space and each element of the vector indicates the presence of the corresponding term. To pre-process the dataset, we remove the stop words use a standard stop list and perform stemming using a porter stemmer, all HTML tags are skipped and all header fields except subject and organization of the posted article are ignored. Finally we select the top 1000 words by mutual information with class labels.

Table 2 gives the confusion matrices built from the clustering results on CSTR dataset. The columns of the confusion matrix are NLP, Robotics/Vision, Systems and Theory respectively. The result shows Systems and Theory are much different from each other, and different from NLP and Robotics/Vision; NLP and Robotics/Vision are similar to each other; AI is more similar to SYSTEMS than ROBOTICS is.

Output	Input	1	2	3	4
A		68	0	8	0
B		8	1	4	120
C		0	0	160	0
D		25	70	6	6

Table 2: Confusion matrix of technical reports by IFD.

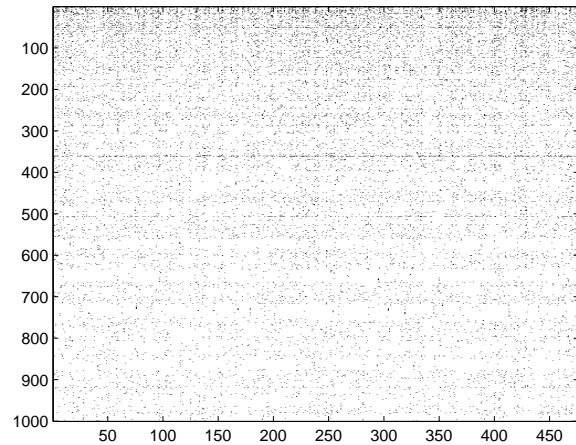
We now try to visualize the cluster structure that might be discovered by *IFD* algorithm. Figure 1 shows the original word-document matrix of CSTR and the reordered matrix obtained by arranging rows and columns based on the cluster assignments. The figure reveals the hidden sparsity structure of both the document and word clusters. The four block diagonals in Figure 1(b) correspond to the four clusters and the dense region at the bottom of the figure identifies the feature outliers (which are distributed uniformly across the technical reports). The rough block diagonal sub-structures observed indicate the cluster structure relation between documents and words. Hence, by exploiting the duality of the data and features and incorporating the feature information in data clustering at all stages, The *IFD* algorithm tends to yield better clustering solution than one-dimensional clustering approaches, especially for high dimensional sparse datasets. The dense region (corresponding feature outliers) also reflects the feature selection ability of *IFD*.

IFD comes with an important by-product that the resulting classes can be easily described in terms of features, since the algorithm employs a dual procedure and also returns feature clusters. In Table 3, we show the four word clusters obtained when applying *IFD* on CSTR dataset. It can be easily seen that these words are meaningful and are often representatives of the associated document cluster. For example, *shared* and *multiprocessor* are representatives of the *Systems* cluster⁶. Similarly, *turing* and *reduction* highlight the theory research efforts at Rochester. An interesting and also important implication is the interpretability of the clustering results. The document clusters could be well explained using its associated feature(word) clusters.

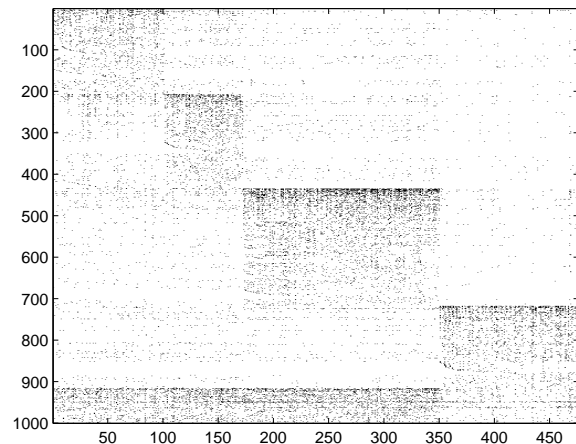
6. RELATED WORK

Traditional clustering techniques has focused on one-sided clustering and they can be classified into partitionial, hierarchical, density-

⁶This conforms with the fact that system research at Rochester's computer science has traditionally focused on shared and parallel system processing. See <http://www.cs.rochester.edu/dept/systems/>.



(a) Original Dataset



(b) Dataset after Reordering

Figure 1: Visualization of the original document-data matrix and the reordered document-data matrix. The shaded region represents non-zero entries.

Cluster 1	Cluster 2	Cluster 3	Cluster 4
shared	trains	tracking	turing
multiprocessors	spoken	freedom	reduction
cache	dialogue	movements	nondeterministic
synchronization	discourse	perception	collapse
locality	plan	calibration	boolean
remote	speaker	target	oracle
load	utterance	sensor	bound
latency	corpus	eye	prove
contention	conversational	filters	reducible
locks	parser	cameras	counting
operating	act	behaviors	circuit
block	inferences	manipulator	few
message	semantic	motor	pspace
butterfly	disambiguation	robotic	relativized
caches	linguistic	arm	string
policies	reason	stage	membership
page	lexicon	reconstructing	sat
busy	phrase	indoor	equivalent
wait	coverage	acquiring	automata
multiprogramming	deductive	geometry	polynomially

Table 3: The four word clusters obtained using IFD on CSTR dataset. Cluster 1, 2, 3, and 4 represent Systems, Natural Language Processing, Robotics/Vision, and Theory respectively. For each cluster, only top 20 words based on the associated degree in the final feature coefficients are included.

based and grid-based [7]. Most of these algorithms use distance functions as objective criteria and are not effective in high dimensional spaces. The IFD algorithm has connections with various recent clustering algorithms such as co-clustering [4], information bottleneck (IB) [14], CoFD [12], non-negative matrix factorization (NMF) [11], spectral clustering [15], binary matrix decomposition [10], subspace clustering [1], adaptive feature selection [5] and etc. The related work can be briefly summarized in Figure 2. The iterative dual optimization in IFD is similar to co-clustering, and the cluster model with data and feature coefficients in IFD is similar to that the data and feature maps in [12]. The optimization procedure of IFD converges to the span of dominant eigenvectors and this share the spirit of spectral clustering [15]. By iteratively reinforcing updating, IFD performs an implicit adaptive feature selection at each iteration and has some common ideas with adaptive feature selection methods. Since each cluster obtained in IFD is associated with some features, IFD can then be regarded as adaptive subspace clustering.

7. CONCLUSIONS

In this paper, we introduced a new cluster model based on data and feature coefficients and then proposed a mutually reinforcing optimization procedure to iteratively cluster both data and features. We also gave a theoretical analysis on the convergence property of the iterative procedure. Experimental results suggested that IFD is a viable and competitive clustering algorithm.

8. REFERENCES

[1] Aggarwal, C. C., Wolf, J. L., Yu, P. S., Procopiuc, C., & Park, J. S. (1999). Fast algorithms for projected clustering. *ACM SIGMOD Conference* (pp. 61–72).

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

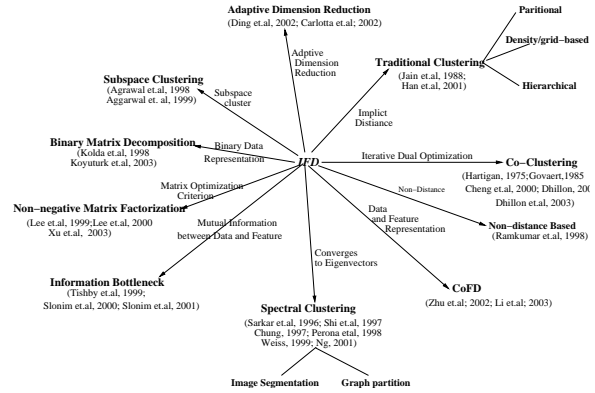


Figure 2: Summary of related work. The words beside the arrows describe connections between the methods.

[3] Deuffhard, P., Huisinga, W., Fischer, A., & Schutte, C. (2000). Identification of almost invariant aggregates in reversible nearly coupled markov chain. *Linear Algebra and Its Applications*, 39–59.

[4] Dhillon, I. S., Mallela, S., & Modha, S. S. (2003). Information-theoretic co-clustering. *SIGKDD'03* (pp. 89–98).

[5] Ding, C., He, X., Zha, H., & Simon, H. (2002). Adaptive dimension reduction for clustering high dimensional data. *Proc. ICDM 2002* (pp. 107–114).

[6] Hastie, T., Tibshirani, R., & Friedman, J. (2001). *The elements of statistical learning: Data mining, inference, prediction*. Springer.

[7] Jain, A. K., & Dubes, R. C. (1988). *Algorithms for clustering data*. Prentice Hall.

[8] Kato, L. (1995). *Perturbation theory for linear operators*. Springer.

[9] Kleinberg, J. M. (1999). Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46, 604–632.

[10] Koyuturk, M., & Grama, A. (2003). PROXIMUS: a framework for analyzing very high dimensional discrete-attributed datasets. *SIGKDD'03* (pp. 147–156).

[11] Lee, D. D., & Seung, H. S. (2000). Algorithms for non-negative matrix factorization. *NIPS* (pp. 556–562).

[12] Li, T., Zhu, S., & Ogihara, M. (2003a). Algorithms for clustering high dimensional and distributed data. *Intelligent Data analysis*, 7, 305–326.

[13] Li, T., Zhu, S., & Ogihara, M. (2003b). Efficient multi-way text categorization via generalized discriminant analysis. *CIKM'03* (pp. 317–324).

[14] Tishby, N., Pereira, F. C., & Bialek, W. The information bottleneck method. *Proc. of the 37-th Annual Allerton Conference on Communication, Control and Computing* (pp. 368–377).

[15] Weiss, Y. (1999). Segmentation using eigenvectors: A unifying view. *ICCV (2)* (pp. 975–982).