A Unifying Approach to Hard and Probabilistic Clustering

Ron Zass and Amnon Shashua
School of Engineering and Computer Science,
The Hebrew University,
Jerusalem 91904, Israel

Abstract

We derive the clustering problem from first principles showing that the goal of achieving a probabilistic, or "hard", multi class clustering result is equivalent to the algebraic problem of a completely positive factorization under a doubly stochastic constraint. We show that spectral clustering, normalized cuts, kernel K-means and the various normalizations of the associated affinity matrix are particular instances and approximations of this general principle. We propose an efficient algorithm for achieving a completely positive factorization and extend the basic clustering scheme to situations where partial label information is available.

1. Introduction

The focus of this paper is twofold. On one hand we address the problem of clustering with "side information" which is the problem of how to guide the clustering process when partial labeling or pairwise label-consistency data is available externally. On the other hand we allow the continuum from having some external guidance to having none. Therefore, we also examine as a particular case the normal clustering problem starting from some distance-based similarity measures ending with a partitioning of the data into k clusters.

We will start with deriving a general algebraic model of clustering which contains as a particular case the well known K-means and spectral clustering approaches [16, 8, 11, 10, 21] which gained much popularity over the last number of years. In a nutshell, we will show that clustering a data set into $k \geq 2$ clusters (whether hard or probabilistic assignments) is equivalent to finding a *completely positive* (CP) factorization of the input affinity matrix under a doubly stochastic constraint. Spectral clustering, for example, is a crude approximation of the CP principle.

2. Clustering and Complete Positivity

Let $\mathbf{x}_i \in R^d$, $i = 1, ..., n$, be points which we wish to assign to k clusters $C_1, .., C_k$ and let $y_i \in \{1, ..., k\}$ be the associated (unknown) labels. There are two versions to the problem: the "hard" clustering scenario where the k clusters are mutually exclusive and the probabilistic assignments ("soft" clustering) where the goal is to assign a probability $P(y_i = j \mid \mathbf{x}_i)$ of point $\mathbf{x}_i$ belonging to cluster C_j.

Spectral clustering and K-means methods are based on pairwise "affinity" measures, which can be interpreted as the probability of pairs of points to be associated with the same cluster. Let $K_{ij} = \phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j) = \kappa(\mathbf{x}_i, \mathbf{x}_j)$ be a symmetric positive-semi-definite affinity function, e.g. $K_{ij} = \exp^{-\|\mathbf{x}_i - \mathbf{x}_j\|^2/\sigma^2}$, providing a measure in the interval $(0, 1]$ with high values associated with pairs of points that are likely to be clustered together (such as based on Euclidean distance). The function $\phi(\mathbf{x})$ is known as a "feature" map responsible for representing the input vectors in some high dimensional space — for the sake of simplicity we need only to assume that a function $\kappa(\mathbf{x}_i, \mathbf{x}_j)$ exists such that the pair-wise similarities form a positive-semi-definite matrix K. The clustering problem is to make assignments (either hard or probabilistic) given the affinity matrix K as input.

We will begin with the hard clustering version. The (kernel) K-means framework seeks an assignment which is compact around centers $\mathbf{u}_1, ..., \mathbf{u}_k$:

$$\min_{\substack{C_1, ..., C_k \\ \mathbf{u}_1, ..., \mathbf{u}_k}} \sum_{r=1}^{k} \sum_{\mathbf{x}_i \in C_r} \|\phi(\mathbf{x}_i) - \mathbf{u}_r\|^2 \qquad (1)$$

where $\mathbf{u}_r = (1/n_r) \sum_{\mathbf{x}_i \in C_r} \phi(\mathbf{x}_i)$ are the centers of the clusters where $|C_r| = n_r$. The affinity matrix enters into play by substituting the definition of $\mathbf{u}_r$ into the optimization function which then, after some algebraic manipulations, becomes:

$$\max_{C_1, ..., C_k} \sum_{r=1}^{k} \frac{1}{n_r} \sum_{\mathbf{x}_i, \mathbf{x}_j \in C_r} \kappa(\mathbf{x}_i, \mathbf{x}_j) \qquad (2)$$

In order to "algebralize" the optimization criterion, let F be an $n \times n$ matrix whose entries are $F_{ij} = 1/n_r$ if

$(i,j) \in C_r$ and $F_{ij} = 0$ otherwise. In other words, if we sort the points $\mathbf{x}_i$ according to cluster membership, then F is a block diagonal matrix with blocks $F_1, ..., F_k$ where $F_r = (1/n_r)\mathbf{1}\mathbf{1}^\top$. Then the optimization function eqn. 2 becomes:

$$\max_F trace(KF) \tag{3}$$

In order to solve this optimization problem we need to spell out the proper constraints on F. There are three types of constraints:

1. F is completely positive (CP) with $cp - rank(F) = k$, i.e., $F = GG^\top$, $G_{n\times k} \geq 0$ and $rank(G) = k$. To see why this is so, let $G = [\mathbf{g}_1, ..., \mathbf{g}_k]$ where $\mathbf{g}_r = (1/\sqrt{n_r})(0, .., 0, 1, ..., 1, 0, ..., 0)^\top$ where the 1s correspond to points $\mathbf{x}_i \in C_r$. Then $\mathbf{g}_r\mathbf{g}_r^\top$ is an $n \times n$ matrix with F_r as the r'th block along the diagonal and zero everywhere else, and $F = \sum_{r=1}^k \mathbf{g}_r\mathbf{g}_r^\top = GG^\top$.

2. $G^\top G = I$, i.e., G has orthonormal columns.

3. F is doubly stochastic, i.e., $F \geq 0$, $F\mathbf{1} = \mathbf{1}$ and $F^\top\mathbf{1} = \mathbf{1}$.

In other words, if we find a matrix F that maximizes $trace(KF)$, under the constraints above, then we have found a solution to the original kernel K-means optimization setup. Spectral clustering, for example, satisfies only *some* of those constraints: substitute $F = GG^\top$ in $trace(KF)$ with the constraint $G^\top G = I$ and obtain:

$$\max_G trace(G^\top KG) \quad s.t.\ G^\top G = I,$$

whose solution is the leading k eigenvectors of K. Typically, the eigenvectors are treated as new coordinates in a k-dimensional subspace of R^n, followed by clustering heuristic approaches such as K-means, dynamic programing and exhaustive search (cf. [8, 21] and references therein).

Practitioners of spectral clustering have also found out that certain normalizations of K have significant positive effect on the clustering result. For example, the normalization $D^{-1/2}KD^{-1/2}$ is employed by the Normalized-Cuts [16] approach and by [8] where $D = diag(K\mathbf{1})$ is a diagonal matrix containing the sum of rows of K[1]. In fact, this normalization is an approximation to the process of replacing K by the closest (under relative entropy error) doubly stochastic matrix:

Proposition 1 *For any non-negative symmetric matrix $K^{(0)}$, iterating the process $K^{(t+1)} \leftarrow D^{-1/2}K^{(t)}D^{-1/2}$ with $D = diag(K^{(t)}\mathbf{1})$ converges to a doubly stochastic matrix.*

The proof is based on showing that the permanent increases monotonically, i.e. $perm(K^{(t+1)}) \geq perm(K^{(t)})$. Because the permanent is bounded the process must converge and if the permanent does not change (at the convergence point) the resulting matrix must be doubly stochastic. Due to lack of space we omit the proof but note this process is a symmetrization of the well known procedure for turning a non-negative matrix to a doubly stochastic matrix by alternating normalizing the row and column sums [17].

To conclude so far, spectral clustering maximizes $trace(KF)$ while enforcing $F = GG^\top$, $G^\top G = I$ and indirectly, via normalization of the affinity matrix, takes the first step in making the affinity matrix become closer to a doubly stochastic matrix. The point we will be making next is that *if one is forced to satisfy only partially the set of constraints (1)-(3), then $G^\top G = I$ is the least important* — its removal is equivalent to making a probabilistic clustering (as we shall see next). On the other hand, the non-negativity constraint $G \geq 0$ is crucial because without it the entries of G loose any physical meaning (which for that reason spectral clustering is well defined for $k = 2$ clusters and becomes heuristic in nature with larger number of clusters).

In the probabilistic setting, given the pairwise membership probabilities K_{ij} (e.g., $K_{ij} = e^{-\|\mathbf{x}_i - \mathbf{x}_j\|^2}$), we wish to recover $G_{ij} = P(y_i = j \mid \mathbf{x}_i)$ the probability that $\mathbf{x}_i$ belongs to the cluster C_j. Under conditional independence assumption $y_i \perp y_j \mid \{\mathbf{x}_i, \mathbf{x}_j\}$, we could evaluate the probability that $\mathbf{x}_i, \mathbf{x}_j$ are clustered together:

$$K_{ij} = \sum_{r=1}^k P(y_i = r \mid \mathbf{x}_i)P(y_j = r \mid \mathbf{x}_j).$$

It is not difficult to see by inspection that:

Proposition 2 *The input affinity matrix K is completely positive $K = GG^\top$, $G \geq 0$, where $G_{ij} = P(y_i = j \mid \mathbf{x}_i)$.*

The doubly stochastic constraint arises from the assumption of uniform priors $P(\mathbf{x}_i) = 1/n$ and $P(y_i = j) = 1/k$ using the Bayes rule as follows:

$$(G^\top\mathbf{1})_j = \sum_{i=1}^n P(y_i = j \mid \mathbf{x}_i) = \frac{n}{k}\sum_i P(\mathbf{x}_i \mid y_i = j) = \frac{n}{k},$$

thus $G^\top\mathbf{1} = (n/k)\mathbf{1}$. We have also: $(G\mathbf{1})_i = \sum_{j=1}^k P(y_i = j \mid \mathbf{x}_i) = 1$, thus $GG^\top\mathbf{1} = (n/k)\mathbf{1}$.

To conclude, the fundamental link between the input affinity matrix K and the output clustering result is that K is completely positive, $K = GG^\top$, $G \geq 0$. In addition, K should be doubly stochastic (or replaced by the closest doubly stochastic matrix) in order to respect the uniform prior assumption on the points and clusters. The doubly stochastic constraint replaces the normalization heuristic employed in the past by practitioners of spectral clustering. The orthonormality constraint $G^\top G = I$ is relevant

[1] We refer the reader to [19] for details on the relation between spectral clustering and normalized cuts

only for enforcing a hard clustering, i.e., a point can belong to one cluster only. In spectral clustering the orthonormality constraint receives a pivotal role (the eigen-decomposition arises solely from the orthonormality) — where in fact it should play a minor role given the analysis above. Instead, the non-negativity is the crucial missing ingredient in the factorization process. In other words, the clustering problem is directly tied to the algebraic problem of finding a completely positive factorization.

Finally, the connection between clustering and non-negativity has been raised in the past — but at a heuristic level. The technique of non-negative matrix factorization (NMF) [9, 7] has been used to cluster the document/word matrix into a non-negative set of basis vectors. Documents are then clustered by commonality with respect to the basis vectors used in the reconstruction [12] . The non-negativity constraint in these cases is motivated by achieving a *sparse* decomposition of the document/word matrix, meaning that each document is represented by a relatively small collection of features (word tuples). As we saw above, the fundamental link between non-negativity and clustering arises through a symmetric non-negative decomposition, rather than a general NMF. Furthermore, the sparsity argument (which may or may not make sense in NMF) does not really apply to a symmetric decomposition.

2.1 The CP Factorization Algorithm

Given the affinity matrix K we wish to find a matrix F which maximizes $trace(KF)$ under the constraints: (i) F is doubly stochastic, (ii) F is completely positive with cp-rank equal to k, i.e., $F = GG^\top$, $G_{n \times k} \geq 0$. We will do that in two phases, first replace K with the closest symmetric doubly stochastic matrix F, and then look for a CP factorization of F by minimizing the L_2 error: $\|F - GG^\top\|^2$ subject to $G_{n \times k} \geq 0$. From the discussion above (Proposition 1), if K is symmetric and non-negative we could iterate the process $D^{-1/2}KD^{-1/2}$ where $D = diag(K\mathbf{1})$ and define F that way. Later we will describe an alternative procedure based on an additive normalization which is relevant when K has negative values (which happens when side information is introduced to the clustering process). Thus, our focus in this section is to find a CP decomposition of F.

For non-negative positive semidefinite matrices V of rank k, the problem of *minimal* r for which $V = GG^\top$, $G \geq 0$ and $rank(G) = r$ has been studied at length in the literature (cf. [3]). This is the so called *cp-rank* problem. The cp-rank is greater or equal to the rank k with the latest upper bound standing at $cp - rank \leq k(k+1)/2 - 1$ [2]. Our case is special in the sense that the cp-rank is equal to the rank, but that on its own is of no help in finding the decomposition.

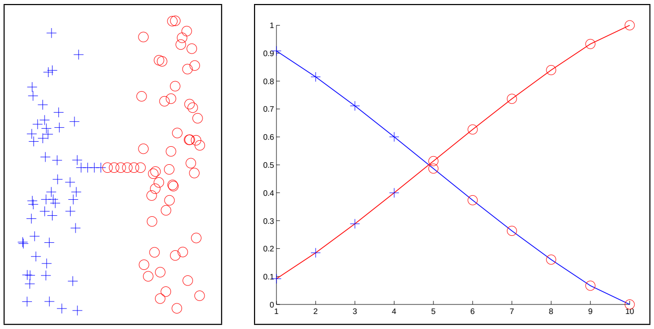


Figure 1: *The values G_{ij} is the probability that point $\mathbf{x}_i$ belongs to cluster C_j. The left-hand display contains 10 points bridging the two clear cut clusters. The right-hand display shows the two values (row of G) for each of the 10 points. One clearly sees that for the points at the extremes (where cluster membership is clear cut) one of the values (along the Y-axis) is indeed low (close to zero) and the other is high. The pair of values tends to even out as the points get closer to the middle where class membership is ambiguous.*

A similar decomposition task is the non-negative matrix factorization (NMF) which looks for an L_2 fit of WH, $W, H \geq 0$ and $W \in R^{l \times k}, H \in R^{k \times l}$ to a given non-negative matrix F. This problem has been studied as well with a simple and effective iterative algorithm proposed by [7]. One would hope that if F is symmetric then the resulting factorization WH would also come out symmetric and moreover that $W = H^\top$ if F is also CP. This regretfully is not the case as one can easily construct counter examples. A recent study by [4] concludes that NMF would generate a symmetric or CP decomposition only in very specialized situations. Therefore the NMF machinery is of no help as well.

What makes our task manageable is that the diagonal entries of K (the probability that $\mathbf{x}_i, \mathbf{x}_i$ are clustered together — which is an undefined measure) can be omitted from consideration. It is the diagonal elements that make the equation $\|F - GG^\top\|^2$ be of higher order than two — in other words, if we can keep the energy function to be of a second order then we could guarantee a converging positive preserving gradient update scheme. Removing the diagonal elements of K from consideration is also a particular instance of a *weighted* decomposition (see later).

We will derive now an iterative update rule and prove its convergence. Let $f()$ denote the optimization function $(1/2)\|F - \sum_j \mathbf{g}_j \mathbf{g}_j^\top\|^2$ where $\mathbf{g}_1, ..., \mathbf{g}_k$ are the columns of G. The differential df is derived below:

$$
\begin{aligned}
df &= d\frac{1}{2} < F - \sum_j \mathbf{g}_j \mathbf{g}_j^\top \ , \ F - \sum_j \mathbf{g}_j \mathbf{g}_j^\top > \\
&= < \sum_j \mathbf{g}_j \mathbf{g}_j^\top - F \ , \ d(\sum_j \mathbf{g}_j \mathbf{g}_j^\top) >
\end{aligned}
$$

$$= \quad < \sum_j \mathbf{g}_j\mathbf{g}_j^\top - F \; , \; \sum_j (d\mathbf{g}_j)\mathbf{g}_j^\top + \mathbf{g}_j(d\mathbf{g}_j)^\top >$$

where $< A, B >= \sum_{i,j} a_{ij}b_{ij}$ is the inner-product operator. The partial derivative with respect to g_{rs} (the s'th entry of $\mathbf{g}_r$) is:

$$\begin{aligned}\frac{\partial f}{\partial g_{rs}} &= < \sum_j \mathbf{g}_j\mathbf{g}_j^\top - F \; , \; \mathbf{e}_s\mathbf{g}_r^\top + \mathbf{g}_r\mathbf{e}_s^\top > \\ &= 2\sum_j g_{js}(\mathbf{g}_j^\top\mathbf{g}_r) - 2\mathbf{g}_r^\top\mathbf{f}_s\end{aligned}$$

where $\mathbf{f}_s$ is the s'th column of F. The partial derivative of the optimization function where the diagonal is weighted out becomes:

$$\frac{\partial f}{\partial g_{rs}} = 2\sum_{j=1}^{k} g_{js} \sum_{i=1, i\neq s}^{n} g_{ji}g_{ri} - 2\sum_{i=1, i\neq s}^{n} g_{ri}F_{si} \tag{4}$$

We will be using a ”positive preserving” gradient descent scheme $g_{rs} \leftarrow g_{rs} - \delta_{rs}\partial f/\partial g_{rs}$. Following [7] we set the gradient step size δ_{rs} as follows:

$$\delta_{rs} = \frac{g_{rs}}{2\sum_{j=1}^{k} g_{js}\sum_{i\neq s}^{n} g_{ji}g_{ri}} \tag{5}$$

After substitution of eqn. 5 into the gradient descent equation we obtain a multiplicative update rule:

$$g_{rs} \leftarrow \frac{g_{rs}\sum_{i\neq s}^{n} g_{ri}F_{si}}{\sum_{j=1}^{k} g_{js}\sum_{i\neq s}^{n} g_{ji}g_{ri}} \tag{6}$$

The update rule preserves positivity, i.e., if the initial guess for G is non-negative and F is symmetric and non-negative, then all future updates will maintain non-negativity. We will now prove that this update rule reduces the optimization energy. Let $f(g_{rs})$ be the energy as a function of g_{rs} (all other entries of G remain constant) and let g'_{rs} be the updated value according to eqn. 6. We wish to show that if we make a gradient descent with a step size δ_{rs} given by eqn. 5 (which as we saw leads to a positive-preserving update), then $f(g'_{rs}) \leq f(g_{rs})$. They key is that δ_{rs} is smaller than the inverse second derivative:

Proposition 3 *The update scheme $g'_{rs} = g_{rs} - \delta_{rs}\partial f/\partial g_{rs}$, with δ_{rs} given by eqn. 5 and the partial first derivative is given by eqn. 4, reduces the optimization function, i.e., $f(g'_{rs}) \leq f(g_{rs})$.*

Proof: The second derivative is:

$$\frac{\partial^2 f}{\partial g_{rs}\partial g_{rs}} = 2\sum_{i=1; i\neq s}^{n} g_{ri}^2,$$

and the step size δ_{rs} satisfies:

$$\begin{aligned}\delta_{rs} = \frac{g_{rs}}{2\sum_{j=1}^{k} g_{js}\sum_{i\neq s}^{n} g_{ji}g_{ri}} &\leq \frac{g_{rs}}{2g_{rs}\sum_{i\neq s}^{n} g_{ri}^2} \\ &= \frac{1}{\partial^2 f/\partial g_{rs}\partial g_{rs}}\end{aligned}$$

The Taylor expansion of $f(g_{rs}+h)$ with $h = -\delta_{rs}\partial f/\partial g_{rs}$ is:

$$f(g'_{rs}) = f(g_{rs}) - \delta_{rs}(\frac{\partial f}{\partial g_{rs}})^2 + \frac{1}{2}\delta_{rs}^2(\frac{\partial f}{\partial g_{rs}})^2\frac{\partial^2 f}{\partial g_{rs}\partial g_{rs}},$$

from which follows:

$$f(g_{rs}) - f(g'_{rs}) = \delta_{rs}(\frac{\partial f}{\partial g_{rs}})^2(1 - \frac{1}{2}\delta_{rs}\frac{\partial^2 f}{\partial g_{rs}\partial g_{rs}}) \geq 0,$$

since $\delta_{rs}\partial^2 f/\partial g_{rs}\partial g_{rs} \leq 1$. □

We apply the update rule in a Gauss-Seidel fashion according to a row-major raster scan of the entries of G (a row-major raster scan has the advantage of enabling efficient caching). Since the energy is lower-bounded, twice differentiable, and is monotonically decreasing via the update rule, yet cannot decrease beyond the lower bound (i.e., positive preserving), then the process will converge onto a local minimum of the optimization function $\|F - GG^\top\|^2$ with the diagonal removed.

Finally, a weighted decomposition can be performed with a straightforward modification. Given a symmetric non-negative weight matrix W we wish to recover $G \geq 0$ that minimizes $\|F - GG^\top\|_W^2$, where $\|A\|_W^2 = \sum_{ij} w_{ij}a_{ij}^2$ is a weighted norm. The need for a weighted decomposition typically arises when not all entries of the affinity matrix are computed, i.e., only a sample of the pairs $\mathbf{x}_i, \mathbf{x}_j$ are drawn and thus K (and F) is sparse. This could happen with very large datasets where it becomes too expensive to consider all possible pairs of points. Since the vanishing entries of K *do not indicate* a zero probability that the corresponding pair of points belong to the same cluster, a weight matrix (in this case binary) is necessary to weight-out the vanishing entries of F from the decomposition. Note that the preceding analysis was a special case with $W = \mathbf{1}\mathbf{1}^\top - I$ (matrix of all 1s with a zero diagonal). The partial derivative becomes:

$$\frac{\partial f}{\partial g_{rs}} = 2\sum_{j=1}^{k} g_{js}\sum_{i=1}^{n} w_{si}g_{ji}g_{ri} - 2\sum_{i=1}^{n} w_{si}g_{ri}F_{si}$$

and the update rule becomes:

$$g_{rs} \leftarrow \frac{g_{rs}\sum_{i=1}^{n} w_{si}g_{ri}F_{si}}{\sum_{j=1}^{k} g_{js}\sum_{i=1}^{n} w_{si}g_{ji}g_{ri}} \tag{7}$$

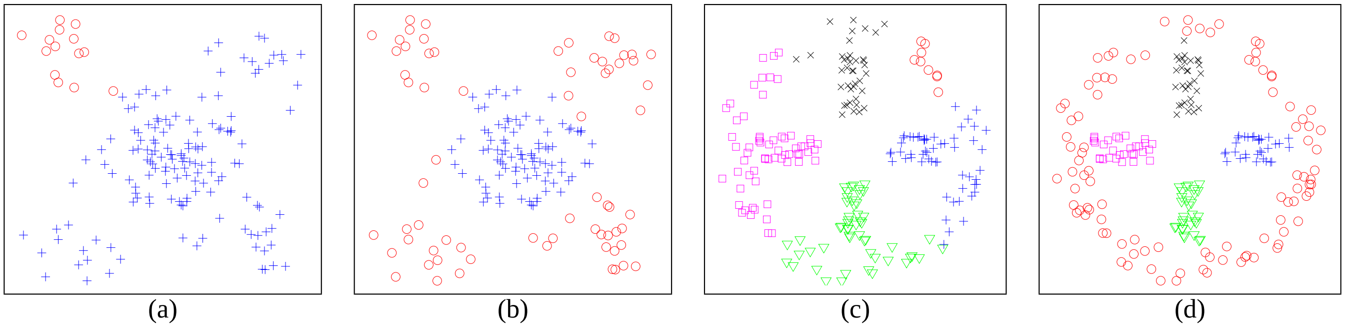


Figure 2: *Comparing multiplicative versus additive normalization (approximations of doubly stochastic) (a) Ncuts using the conventional multiplicative normalization for $k = 2$ clusters compared to (b) using the additive normalization. (c) and (d) are with $k = 5$ clusters. Note that in (d) the outer rim is assigned into a single cluster. The results of the CP factorization are almost identical to (b) and (d).*

2.2 Clustering With Side Information

In this section we will introduce additional considerations when partial label information is added to the mix — also known as ”side” information. For example, a user may indicate that a certain pair of points in the dataset are judged to be similar and a certain other pair of points are judged to arise from separate clusters. This type of information is useful not only as a way to simplify a difficult clustering task but also in some cases the data contains multiple clustering solutions. For example, face images can be clustered based on person identity or illumination variation or facial expressions, documents can be clustered based on topic, authorship or writing style, and so forth.

This type of problem was addressed in the past from various standpoints. In [20] the pairwise label consistency data was used to shape the Euclidean distance metric by looking for an optimal weighting matrix which parameterizes the Mahalanobis distances in the data space. Others (cf. [15]) incorporate the label consistency into a k-means or more generally a Gaussian mixture EM model to find a partitioning that reflect the user's notion of meaningful clusters, or use the side information for extracting meaningful features from the data [5, 14].

We define a ”label consistency” symmetric matrix R such that $R_{ij} = 1$ if it is known apriori that $\mathbf{x}_i, \mathbf{x}_j$ are clustered together, $R_{ij} = -1$ if they are known *not* to be clustered together, and $R_{ij} = 0$ if label consistency information is not given. Replace the affinity matrix K with $K' = K + \alpha R$ with $0 < \alpha \leq 1$ is a balancing factor. The non-vanishing entries of R provide external (user based) information which could support or suppress the local affinities.

The combined local and external affinity matrix K' is symmetric but *not necessarily* non-negative. As we saw in the previous section, the doubly-stochastic constraint is an important ingredient in the success of the clustering process, however, the multiplicative normalization $D^{-1/2}K'D^{-1/2}$ (whether as a single step or by iterating the normalization until convergence) is not appropriate here. The multiplicative normalization will not turn a negative matrix to a non-negative one, thus, instead we will introduce an additive normalization by setting up the following optimization function: $\min_F \|F - K'\|^2$ subject to a scaled doubly stochastic constraint: $F^\top \mathbf{1} = F\mathbf{1} = \beta\mathbf{1}$, and $F = F^\top$. We leave the non-negativity constraint $F \geq 0$ to later. Since $\|F - K'\|^2 = trace((F - K')^\top (F - K'))$ we obtain the optimization problem below (using $K' = K'^\top$):

$$\max_F \quad trace(K'F - \frac{1}{2}F^\top F) \qquad \text{s.t. } F^\top \mathbf{1} = F\mathbf{1} = \beta\mathbf{1}, F = F^\top. \tag{8}$$

Setting the derivative of the Lagrangian of eqn. 8 (noting that F should be symmetric) to zero yields:

$$F = K' + \mu\mathbf{1}^\top + \mathbf{1}\mu^\top, \tag{9}$$

where μ is the vector of Lagrange multipliers. Multiply both sides by $\mathbf{1}$ noting that $F\mathbf{1} = \beta\mathbf{1}$ we obtain:

$$\begin{aligned} \mu &= (\mathbf{1}\mathbf{1}^\top + nI)^{-1}(\beta I - K')\mathbf{1} \\ &= \frac{1}{n}(I - \frac{1}{2n}\mathbf{1}\mathbf{1}^\top)(\beta I - K')\mathbf{1} \end{aligned}$$

Substituting μ back into eqn. 9 we obtain:

$$F = K' + \left(\frac{\beta}{n}I + \frac{\mathbf{1}^\top K'\mathbf{1}}{n^2}I - \frac{1}{n}K'\right)\mathbf{1}\mathbf{1}^\top - \frac{1}{n}\mathbf{1}\mathbf{1}^\top K' \tag{10}$$

The parameter β is now set so that $F \geq 0$. In other words, the non-negativity requirement is circumvented by having a scaled doubly stochasticity. We then proceed with the CP decomposition process (looking for $G \geq 0$ that minimizes $\|F - GG^\top\|^2$) described in the previous section.

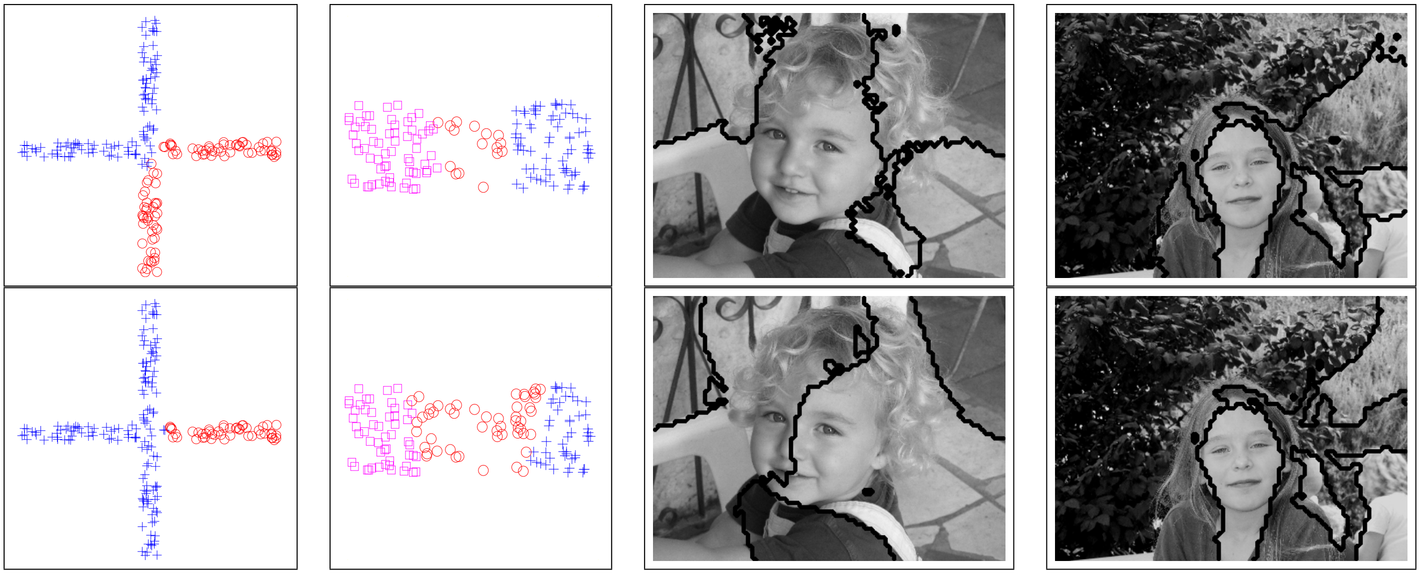


Figure 3: *CP factorization (top row) versus Ncuts (bottom row). Additive normalization used for the dot patterns and multiplicative for the real image examples.*

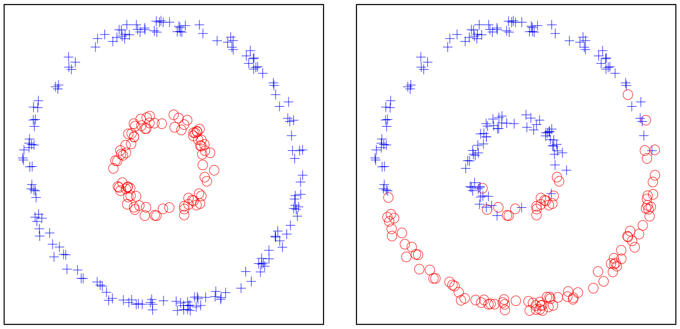
3 Experiments

Our algorithm (update rule eqn. 6 or eqn. 7 in the weighted case) produces a probabilistic assignment matrix G whose value G_{ij} is the probability that point $\mathbf{x}_i$ belongs to cluster C_j (recall that the omission of $G^\top G = I$ makes G represent probabilistic assignments). A ”hard” clustering can be made by assigning $\mathbf{x}_i$ to the cluster with highest probability of assignment (maximum over the corresponding row of G). Consider the point-set of Fig. 1 where most of the points clearly fall into one of the two clusters except for a subset of 10 points bridging between the two point sets along a straight line path. The matrix G has two columns, thus for every point (row of G) we have two values — one should be high and the other should be low if indeed there is a clear cut membership assignment. The right-hand display of Fig. 1 shows the pair of values associated with each of those 10 bridging points. One clearly sees that for the points at the extremes (where cluster membership is clear cut) one of the values (along the Y-axis) is indeed low (close to zero) and the other is high. The pair of values tends to even out as the points get closer to the middle where class membership is ambiguous.

Next we look into comparative results with spectral clustering (using Normalized-Cuts as the representative of the class of algorithms). There are two issues to consider: one is the choice of normalization of the affinity matrix K, i.e., multiplicative $D^{-1/2}KD^{-1/2}$ or additive (eqn. 10); and the second is the choice of the clustering algorithm to see whether the CP decomposition yields more accurate results than spectral clustering.

As for normalization, in all our experiments the additive normalization worked better than the conventional multiplicative. The additive normalization is the result of an L_2 fit of a doubly stochastic matrix to K whereas the multiplicative normalization is an *approximation* to a relative entropy fit (Prop. 1). Since both the spectral and the CP factorization rely on an L_2 error model, having the normalization step fall under a different error model introduces a loss in the overall accuracy — this is our conjecture. Fig. 2 illustrates this with two different data sets (a),(b) and (c),(d). The result of Ncuts (with the conventional multiplicative normalization) with $k = 2$ is displayed in (a) and with the additive normalization in (b). Likewise for (c) and (d) with $k = 5$ clusters. Note that in (d) the points on the outer rim are clustered together, unlike in (c). The performance of the CP factorization was almost identical to (b) and (d), respectively. Iterating the multiplicative normalization (Prop. 1) had only a minor improvement to (a) and (c).

The disadvantage of the additive normalization is that sparsity of K is not preserved, i.e., vanishing entries in K do not necessarily vanish followed by the normalization. This is especially important when applying the clustering scheme to grey-level images with the affinity matrix designed such that the affinity between distant pixels are attenuated — thus effectively creating a sparse K [16]. In those cases we used the multiplicative normalization. Fig. 3 shows additional patterns and real image segmentation with the resulting clustering by CP factorization (top row) and Ncuts (bottom row). The dot patterns examples were applied with the additive normalization and the real image examples with the multiplicative and with the affinity matrix



(a) (b)

Figure 5: *(a) donut pattern clustered using K-means and CP factorization. Both algorithms were subject to 1000 runs starting from random initial guesses. The CP produced the same result shown here for all trials where K-means succeeded only 9 out of the 1000 trials. (b) same pattern but with only 10% of the pairwise distances sampled — the remaining entries of K were given zero values. The CP clustering produced the correct result (shown in (a)) whereas the Ncuts produced the result shown in (b).*

design described in [16]. Fig. 4 illustrates the use of label consistency information for improving a segmentation.

We also compared the accuracy of the clustering to the K-means result over 1000 trials starting with random initial guesses. Unlike spectral clustering, both K-means and CP factorization rely on an iterative scheme bound to settle on a local optimum. Fig. 5a shows a donut pattern used for this type of experiment. The resulting clustering was consistently achieved on all 1000 trials of the test whereas only 9 out of 1000 K-means runs succeeded in converging to this solution (all other solutions roughly divided the pattern across the diameter of the big circle).

Finally, we used the same pattern to test the weighted version of the CP factorization (eqn. 7). We sampled 10% of the pairs of points in the donut pattern and run the weighted CP factorization using zero weights for all the un-sampled entries in K. The resulting clustering was identical to Fig. 5a whereas the Ncuts running on the same (sparse) K produced the result in Fig. 5b. The poor result of Ncuts is not surprising because the scheme has no way to know that the vanishing entries of K should not be considered. A weighted version of spectral decomposition is fairly complex (see [18]) whereas the weighted version of CP factorization requires only a straightforward modification of the original scheme.

4 Discussion

To achieve a good clustering result, the quality of the ”affinity” matrix K, which measures the probability K_{ij} that the pair $\mathbf{x}_i, \mathbf{x}_j$ are clustered together, is by far the crucial ingredient. This is similar to the roles played by feature measurements and classification engines in pattern recognition — given good discriminatory features even a naive classifier will do. In this paper we focused on the clustering engine and this is where most of the past research has been conducted as well.

What we have shown in this paper is that all the past research on this topic can be condensed into a single principle which is the *complete positivity* of the affinity matrix $K = GG^\top$, $G \geq 0$. We have shown how to start from the K-means scheme and arrive to the CP principle and also how to start with a probabilistic scheme based on the notion of conditional independence and arrive to the same CP framework. We have seen that the normalization attempts used in the past are related to a doubly stochastic constraint on K arising from uniform priors on the data points. We have shown that the difference between the kernel K-means setup and the probabilistic setup is that with K-means one adds an orthonormality constraint $G^\top G = I$ which forces ”hard” clustering. The hard clustering desire is responsible for the spectral clustering algorithms which as a result give an (unwarranted) pivotal role to orthonormality instead of the more crucial constraint of non-negativity. In other words, starting with an admissible K (should be doubly stochastic), a completely positive factorization $K = GG^\top$, $G \geq 0$ will provide a probabilistic clustering, whereas an orthonormal factorization $K = GG^\top$, $G^\top G = I$ (without non-negativity) has no physical meaning (and which becomes worse beyond the special case of two clusters). Nevertheless, from a practical standpoint, although all our experiments have shown that the CP factorization generates better clusters (especially with $k > 2$), the spectral clustering results were not that far behind. This brings us back to the beginning of this section: the design and normalization of the affinity matrix is by far more critical than anything else in the clustering process.

Finally, the CP principle naturally extends to higher dimensions. In many clustering situations it is more natural to compute affinities among n-tuples, rather than among pairs of data points. This has been pointed out recently in [13, 1, 6]. Let $K_{i_1,...,i_n}$ be an n-way array (tensor) representing the (known) probability that points $\mathbf{x}_{i_1}, ..., \mathbf{x}_{i_n}$ are clustered together, and let G be defined as before, i.e., $G_{ij} = P(y_i = j \mid \mathbf{x}_i)$. Then, using the same arguments as above (conditional independence), we have:

$$\begin{aligned} K_{i_1,...,i_n} &= \sum_{r=1}^{k} P(y_{i_1} = r \mid \mathbf{x}_{i_1}) \cdots P(y_{i_n} = r \mid \mathbf{x}_{i_n}) \\ &= \sum_{r=1}^{k} G_{i_1,r} \cdots G_{i_n,r} \end{aligned}$$

which means that $K = \sum_{r=1}^{k} \mathbf{g}_r \otimes ... \otimes \mathbf{g}_r = \sum_r \mathbf{g}_r^{\otimes n}$ where $G = [\mathbf{g}_1, ..., \mathbf{g}_k]$. In other words, the affinity tensor must be a super-symmetric non-negative rank=k tensor.

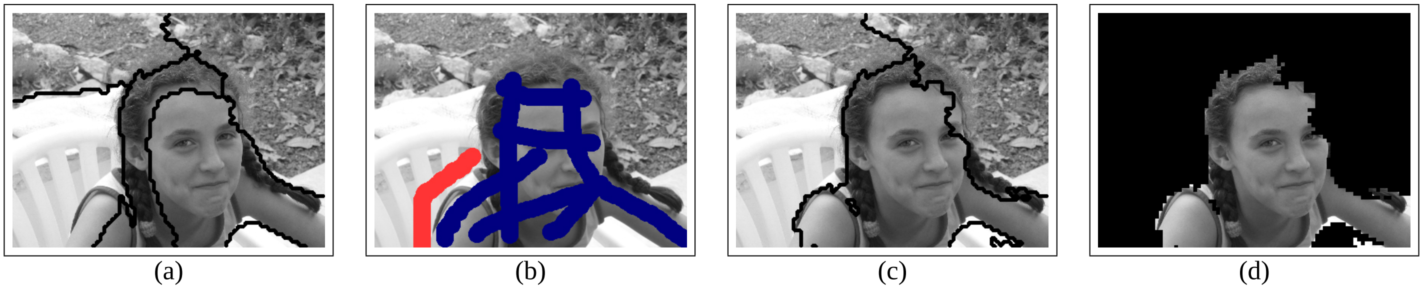


Figure 4: *Use of label consistency to improve clustering. (a) CP segmentation with $k = 5$, (b) foreground (head and shoulders) and background (part of the chair) strips. Points in the foreground are "label consistent", $R_{ij} = 1$, and points between foreground and background are "label inconsistent", $R_{ij} = -1$. (c) CP clustering with side information, (d) the resulting "head and shoulders" cluster.*

The update rule devised for the case $n = 2$ can be naturally extended to the general $n > 2$ (left for future publication - but for preliminary examples see [13]). Extending spectral clustering and the analogy to graph cuts to higher dimensions requires the use of hyper-graphs or ways of flattening down the affinity tensor to a matrix followed by a spectral decomposition [1, 6] — our comments above on the apparent excessive role of orthonormality in spectral clustering should be even further amplified in higher dimensions.

References

[1] S. Agrawal, J. Lim, L. Zelnik-Manor, P. Perona, D. Kriegman, and S. Belongie. Beyond pairwise clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[2] F. Barioli and A. Berman. The maximal cp-rank of rank k completely positive matrices. *Linear Algebra Appl.*, 363:17–33, 2003.

[3] A. Berman and N. Shaked-Monderer. *Completely Positive Matrices*. World Scientific Publishing Co., 2003.

[4] M. Catral, L. Han, M. Neumann, and R.J. Plemmons. On reduced rank for symmetric nonnegative matrices. *Linear Algebra and its Applicatoins*, 393:107–126, 2004.

[5] G. Chechik and N. Tishby. Extracting relevant structures with side information. In *Proceedings of the conference on Neural Information Processing Systems (NIPS)*, volume 15, 2002.

[6] V.M. Govindu. A tensor decomposition for geometric grouping and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[7] D. Lee and H. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791, 1999.

[8] A.Y. Ng, M.I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Proceedings of the conference on Neural Information Processing Systems (NIPS)*, 2001.

[9] P. Paatero and U. Tapper. Positive matrix factorization: a non-negative factor model with optimal utilization of error estimates of data values. *Envirometrics*, 5:111–126, 1994.

[10] M. Pavan and M. Pelillo. A new graph-theoretic approach to clustering and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2003.

[11] P. Perona and W. Freeman. A factorization approach to grouping. In *Proceedings of the European Conference on Computer Vision*, 1998.

[12] F. Shahnaz, M. Berry, P. Pauca, and R. Plemmons. Document clustering using nonnegative matrix factorization. *Journal on Information Processing and Management, to appear*, 2004.

[13] A. Shashua and T. Hazan. Non-negative tensor factorization with applications to statistics and computer vision. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2005.

[14] A. Shashua and L. Wolf. Kernel feature selection with side data using a spectral approach. In *Proceedings of the European Conference on Computer Vision*, 2004.

[15] N. Shental, A. Bar-Hillel, T. Hertz, and D. Weinshall. Computing gaussian mixture models with em using equivalence constraints. In *Proceedings of the conference on Neural Information Processing Systems (NIPS)*, 2003.

[16] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8), 2000.

[17] R. Sinkhorn. A relationship between arbitrary positive matrices and doubly stochastic matrices. *Ann. Math. Statist.*, 35:876–879, 1964.

[18] N. Srebro and T. Jaakola. Weighted low-rank approximations. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2003.

[19] Y. Weiss. Aegmentation using eigenvectors: a unifying view. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1999.

[20] E.P. Xing, A.Y. Ng, M.I. Jordan, and S. Russel. Distance metric learning, with applications to clustering with side information. In *Proceedings of the conference on Neural Information Processing Systems (NIPS)*, volume 15, 2002.

[21] S.X. Yu and J. Shi. Multiclass spectral clustering. In *Proceedings of the International Conference on Computer Vision*, 2003.