Context-Aware Clustering

Junsong Yuan
EECS Dept., Northwestern Univ.
Evanston, IL, USA
j-yuan@northwestern.edu

Ying Wu
EECS Dept., Northwestern Univ.
Evanston, IL, USA
yingwu@eecs.northwestern.edu

Abstract

Most existing methods of semi-supervised clustering introduce supervision from outside, e.g.*, manually label some data samples or introduce constrains into clustering results. This paper studies an interesting problem: can the supervision come from inside,* i.e.*, the unsupervised training data themselves? If the data samples are not independent, we can capture the contextual information reflecting the dependency among the data samples, and use it as supervision to improve the clustering. This is called context-aware clustering. The investigation is substantialized on two scenarios of (1) clustering primitive visual features (*e.g.*, SIFT features) with help of spatial contexts, and (2) clustering '0'–'9' hand written digits with help of contextual patterns among different types of features. Our context-aware clustering can be well formulated in a closed-form, where the contextual information serves as a regularization term to balance the data fidelity in original feature space and the influences of contextual patterns. A nested-EM algorithm is proposed to obtain an efficient solution, which proves to converge. By exploring the dependent structure of the data samples, this method is completely unsupervised, as no outside supervision is introduced.*

1. Introduction

Unsupervised clustering is largely settled by the distance metric that measures the dissimilarity or affinity between two data points. This can be regarded as the internal force driving the clustering. Typical examples include the k-means clustering and spectral clustering. In practice, as it is generally quite difficult to choose the right distance metric in advance, we tend to learn a good metric by imposing supervision. Acting as a constraint, supervised information can be regarded as the external force that balances or adjusts the effect of the internal force. In this way, we can say the distance metric is tuned or learned. Supervision is generally introduced from outside, *e.g.*, manually labeling some samples as constraints to perform constrain-based clustering [12], or to perform co-training among multiple modalities [3], or to perform metric tuning [10]. Then, here is an interesting question: can the supervision come from inside, *i.e.*, the training data themselves? If so, it is still unsupervised, and can be called self-supervised clustering.

This is possible when training data are not independent. The dependency among data is the contextual information. Let's take web-page grouping as an example. The links among web-pages provide information on dependency. We group web-pages not only based on if they have similar contents (*features*) but also if they share similar link pages (*contexts*). Contextual information brought by data dependency provides an important clue for data mining [8]. In computer vision research, many recent work showed that contextual information can be utilized to resolve the ambiguities and uncertainties in many applications, including image search [6], recognition [9] [1] [2], metric learning [11], and image modeling [16].

If the data dependency can be well captured by the contextual patterns, which describe the co-occurrences of specific type of data samples in a higher level, it is possible to use it as the supervision to improve clustering. Because the contextual information is discovered from the unsupervised training data themselves, we call such a self-supervised clustering as *context-aware clustering*. We substantialize it on two case studies where (1) we cluster primitive visual features (*e.g.*, SIFT features) for finding local spatial patterns in images, and (2) we cluster '0'–'9' hand written digits with multiple features. By feeding back the contextual patterns as supervision which characterize the co-occurrence statistics, we can resolve the ambiguous samples based on the hints from their contexts.

The novelty of our work lies in two aspects. First of all, we give a closed-form formulation of context-aware clustering, where the contextual information serves as a regularization term in traditional k-means clustering. Secondly, due to the nice analytical properties of the new formulation, we present an efficient nested-EM algorithm for context-aware clustering, which proves to converge. Both simulation and real data validate the effectiveness of our method.

978-1-4244-2243-2/08/$25.00 ©2008 IEEE

2. Context-Aware Clustering

2.1. Motivating example: clustering visual primitives

We illustrate our context-aware clustering in a case study of clustering visual primitives. Each visual primitive is denoted as $v = (x, y, \mathbf{f})$, where (x, y) is its spatial location, and $\mathbf{f}$ denotes the feature vector describing v. In general $\mathbf{f} \in \mathbb{R}^d$ can be any possible visual features to characterize a local image region, like color histograms or SIFT-like features [4] [5]. An image is a collection of visual primitives, and we denote the visual primitive database as $\mathcal{D}_v = \{v_i\}_{i=1}^N$. After clustering these visual primitives into words, we can label each $v_i \in \mathcal{D}_v$ with $l(v_i) \in \mathbf{\Omega}$, where $\mathbf{\Omega}$ is the visual word lexicon of size $|\mathbf{\Omega}| = M$.

The context of a visual primitive is its spatial neighbors in the image, *i.e.*, those visual primitives that collocate with it (Fig. 1). For each visual primitive $v_i \in \mathcal{D}_v$, we define its local spatial neighborhood, *e.g.* K-nearest neighbors (K-NN) or ϵ-nearest neighbors (ϵ-NN), as its *context group* $\mathcal{G}_i = \{v_i, v_{i_1}, v_{i_2}, \cdots, v_{i_K}\}$. The *context database* is denoted by $\mathbf{G} = \{\mathcal{G}_i\}_{i=1}^N$. Once the visual primitives are labeled by $\mathbf{\Omega}$, the context database $\mathbf{G}$ can be transfered to a *transaction database* with N records, where each record $\mathbf{t}_i \in \{0,1\}^M$ is a binary vector representation of $\mathcal{G}_i$ by indicating which words appear in group $\mathcal{G}_i$. This transaction database is a sparse binary matrix $\mathbf{T}_{M\times N}$, where each column is a context transaction $\mathbf{t}_i$. The entry $t_{ij} = 1$ indicates the j_{th} transaction contains the i_{th} word and $t_{ij} = 0$ otherwise. In the case of using spatial K-NN to define context group, we have $\sum_{i=1}^M t_{ij} = K, \ \forall \ j = 1, ..., N$, because each context group $\mathcal{G}_i$ contains K visual primitives.

An $N \times N$ sparse binary matrix $\mathbf{Q}$ can be used to describe the spatial context relations among the visual primitives, where $q_{ij} = 1$ denotes that v_i belongs to the context group of v_j, *i.e.* $v_i \in \mathcal{G}_j$; and $q_{ij} = 0$ otherwise. Matrix $\mathbf{Q}$ is symmetric when using ϵ-NN to define spatial neighbors, while an asymmetric matrix when using K-NN. The context matrix $\mathbf{Q}$ plays a central role in our context-aware clustering as it introduces extra relations among data samples other than in the feature space. Besides spatial contexts, $\mathbf{Q}$ can present any other possible contextual information among the N data samples. In Sec. 3.3, we give another example of applying contextual information from multiple features for clustering.

Based on the word lexicon $\mathbf{\Omega}$ ($|\mathbf{\Omega}| = M$), we can further define a *phrase lexicon* $\mathbf{\Psi} = \{\mathcal{P}_i\}_{i=1}^{\tilde{M}}$, where each phrase $\mathcal{P}_i \in \mathbf{\Psi}$ is a *contextual pattern* composed of a collection of words, *i.e.* $\mathcal{P}_i \subset \mathbf{\Omega}$. Compared with visual words which label visual primitives v, visual phrases label transactions $\mathbf{t}$. As visual phrases describes the spatial dependencies among visual words, they can be more meaningful patterns in a higher level [14]. For example in Fig. 1, the existence of a visual phrase $\mathcal{P} = \{a, b\}$ shows that two words $a, b \in \mathbf{\Omega}$ co-occur frequently in local image regions and may form a meaningful visual pattern. Each $\mathcal{P}_j \in \mathbf{\Psi}$ is presented by a binary vector $\tilde{\mathbf{u}}_j \in \{0,1\}^M$ which describes its word compositions, where $\tilde{\mathbf{u}}_j(i) = 1$ indicates that the i_{th} word is contained in $\mathcal{P}_j$. The matrix $\tilde{\mathbf{U}}_{M\times\tilde{M}}$ is further applied to represent $\mathbf{\Psi}$, where each column of $\tilde{\mathbf{U}}$ is a $\tilde{\mathbf{u}}_j$. Correspondingly, we use a real matrix $\mathbf{U}_{d\times M}$ to represent $\mathbf{\Omega}$, where each column is a feature vector to represent a word prototype $\mathbf{u}_j \in \mathbb{R}^d$. All of our notations are listed in table 1.

Symbol	Definition
d	dimensionality of the feature vector
N	number of visual primitives
M	number of visual words
$\tilde{M}$	number of visual phrases
$\mathbf{t}_{M\times 1}$	a context transaction
$\mathbf{T}_{M\times N}$	the transaction database
$\mathbf{Q}_{N\times N}$	spatial context relations of visual primitives
$\mathbf{u}_{d\times 1}$	prototype of a visual word
$\tilde{\mathbf{u}}_{M\times 1}$	prototype of a visual phrase
$\mathbf{U}_{d\times M}$	prototypes of M visual words
$\tilde{\mathbf{U}}_{M\times\tilde{M}}$	prototypes of $\tilde{M}$ visual phrases
$\mathbf{R}_{M\times N}$	label matrix of N primitives with M words
$\tilde{\mathbf{R}}_{\tilde{M}\times N}$	label matrix of N groups with $\tilde{M}$ phrases
$\mathbf{D}_{M\times N}$	distortion matrix of N primitives with M words
$\tilde{\mathbf{D}}_{\tilde{M}\times N}$	distortion matrix of N groups with $\tilde{M}$ phrases

Table 1. Notations of symbols. Bold upper case letters denote matrices and bold lower case letters denote vectors.

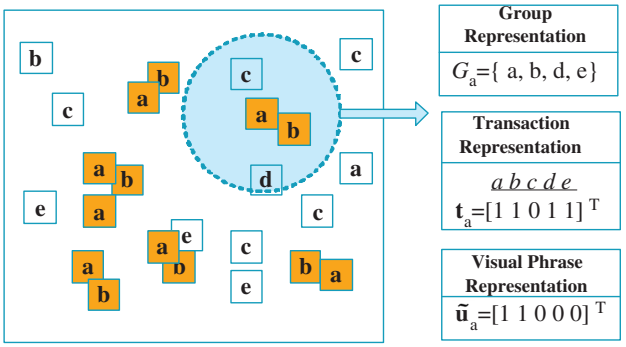


Figure 1. Illustration of spatial contexts: context group $\mathcal{G}$, transaction $\mathbf{t}$ and visual phrase $\mathcal{P}$. The left figure denotes an image and each rectangle denotes a visual primitive. We suppose the visual word lexicon contains 5 words: $\mathbf{\Omega} = \{a, b, c, d, e\}$ and each visual primitive is labeled by a word. The circle denotes a spatial context group generated by a visual primitive a. The highlighted visual primitives are instances of discovered visual phrase $\mathcal{P} = \{a, b\}$.

2.2. Problem Formulation

We first review the k-means clustering and its solution by the EM-algorithm. By performing traditional k-means clustering on a collection of visual primitives $v_i \in \mathcal{D}_v$, the following mean square distortion is minimized:

$$\mathbf{J}_1 = \sum_{i=1}^{N}\sum_{j=1}^{M} r_{ij}\|\mathbf{f}_i - \mathbf{u}_j\|^2 = tr(\mathbf{R}^T\mathbf{D}), \qquad (1)$$

where

- $\mathbf{f}_i$ is the $d \times 1$ feature vector, and $\mathbf{u}_j$ is the center of the cluster (prototype of visual words); $\|\cdot\|$ denotes the Euclidean distance and $tr(\cdot)$ denotes the matrix trace;
- $\mathbf{D}_{M\times N}$ denotes the distance matrix, where $d_{ij} = \|\mathbf{f}_j - \mathbf{u}_i\|^2$ denotes the distance between the j_{th} visual primitives and the i_{th} visual word prototype;
- $\mathbf{R}_{M\times N}$ denotes the label indicator matrix of the visual primitives, where $r_{ij} = 1$ if the j_{th} visual primitive is labeled with the i_{th} word; and $r_{ij} = 0$ otherwise.

Standard EM-algorithm can be performed to minimize the distortion in Eq. 1 by iteratively updating $\mathbf{R}$ (E-step) and $\mathbf{D}$ (M-step). By minimizing the objective function $\mathbf{J}_1$, k-means clustering tries to maximize the data likelihood under mixture Gaussian distribution and assumes all observation samples $v_i \in \mathcal{D}_v$ are independent from one another:

$$Pr(\mathcal{D}_v|\mathbf{\Omega}) = \prod_{i=1}^{N} Pr(v_i|\mathbf{\Omega}). \qquad (2)$$

However, such an independent assumption does not hold here because visual primitives have spatial dependency with each other. Thus they are not independent in the feature space. As a result, we need to take into consideration these spatial contextual information and cannot cluster visual primitives only based on their features $\mathbf{f}_i$. In order to consider both feature and contextual information for clustering, we propose a regularized objective function based on k-means:

$$\begin{aligned}\mathbf{J} &= \sum_{i=1}^{N}\sum_{j=1}^{M} r_{ij}\|\mathbf{f}_i - \mathbf{u}_j\|^2 + \lambda\sum_{i=1}^{N}\sum_{j=1}^{\tilde{M}} r'_{ij} d_H(\mathbf{t}_i, \tilde{\mathbf{u}}_j) \\ &= tr(\mathbf{R}^T\mathbf{D}) + \lambda \times tr(\tilde{\mathbf{R}}^T\tilde{\mathbf{D}}), \qquad (3)\end{aligned}$$

where

- $\lambda > 0$ is a positive constant for regularization;
- r'_{ij} is the binary label indicator of transactions, with $r'_{ij} = 1$ denoting that the i_{th} transaction is labeled with the j_{th} visual phrase; and $r'_{ij} = 0$ otherwise. Similar to $\mathbf{R}$, $\tilde{\mathbf{R}}_{N\times\tilde{M}}$ is a matrix to describe the clustering results of transactions $\mathbf{t}$. For deterministic clustering, we have the following constraints for $\mathbf{R}$ and $\tilde{\mathbf{R}}$:

$$\sum_{j=1}^{M} r_{ij} = 1, \qquad \sum_{j=1}^{\tilde{M}} \tilde{r}_{ij} = 1, \qquad \forall\ i = 1, ..., N. \quad (4)$$

- $d_H(\mathbf{t}_i, \tilde{\mathbf{u}}_j)$ denotes the Hamming distance between two binary vectors: a transaction $\mathbf{t}_i$ and a context pattern $\tilde{\mathbf{u}}_j$, where $\mathbf{1}$ is the $M \times 1$ all 1 vector:

$$\begin{aligned} d_H(\mathbf{t}_i, \tilde{\mathbf{u}}_j) &= M - \left[\mathbf{t}_i^T\tilde{\mathbf{u}}_j + (\mathbf{1} - \mathbf{t}_i)^T(\mathbf{1} - \tilde{\mathbf{u}}_j)\right] \\ &= \mathbf{t}_i^T\mathbf{1} + \tilde{\mathbf{u}}_j^T\mathbf{1} - 2\mathbf{t}_i^T\tilde{\mathbf{u}}_j. \qquad (5)\end{aligned}$$

Given the objective function in Eq. 3 with M, $\tilde{M}$ and λ are fixed parameters, our objectives are two-fold: (1) clustering all the visual primitives v_i into M classes (word lexicon $\mathbf{\Omega}$) and simultaneously (2) clustering all the context transactions $\mathbf{t}_i \in \mathbf{T}$ into $\tilde{M}$ classes (phrase lexicon $\mathbf{\Psi}$). The clustering results are presented by $\mathbf{R}$ and $\tilde{\mathbf{R}}$ respectively. Since each visual primitive can generate a spatial context group, we finally end up with two labels for every primitive: (1) the word label of itself and (2) the phrase label of the spatial group it generates. Compared with k-means clustering which assumes convex (*e.g.* Gaussian) shape for each cluster in the feature space, our regularization term can modify the cluster into an arbitrary shape by considering the influences from the higher phrase level. Similar to the k-means clustering, this formulation is also a chicken-and-egg problem where we cannot estimate $\mathbf{D}$, $\tilde{\mathbf{D}}$, $\mathbf{R}$ and $\tilde{\mathbf{R}}$ simultaneously.

2.3. Iterative Solution: a Nested-EM algorithm

The objective function in Eq. 3 can be partitioned into two parts:

$$\mathbf{J} = \underbrace{tr(\mathbf{R}^T\mathbf{D})}_{\mathbf{J}_1} + \underbrace{\lambda \times tr(\tilde{\mathbf{R}}^T\tilde{\mathbf{D}})}_{\mathbf{J}_2},$$

where $\mathbf{J}_1 = tr(\mathbf{R}^T\mathbf{D})$ and $\mathbf{J}_2 = \lambda \times tr(\tilde{\mathbf{R}}^T\tilde{\mathbf{D}})$ correspond to the quantization distortions of visual primitives and context groups respectively. Although it looks we could minimize $\mathbf{J}$ by minimizing $\mathbf{J}_1$ and $\mathbf{J}_2$ separately, *e.g.*, through two independent EM-processes, this is actually infeasible because $\mathbf{J}_1$ and $\mathbf{J}_2$ are coupled. By further analyzing $\mathbf{J}_1$ and $\mathbf{J}_2$, we find that although visual primitive distortions $\mathbf{D}$ only depends on $\mathbf{R}$, the context group distortions $\tilde{\mathbf{D}}$ depends on *both* visual primitive labels $\mathbf{R}$ and context group labels $\tilde{\mathbf{R}}$. Thus it is infeasible to minimize $\mathbf{J}_1$ and $\mathbf{J}_2$ separately due to their correlation. In the following, we show how to decouple the dependency between $\mathbf{J}_1$ and $\mathbf{J}_2$ and propose our nested-EM algorithm.

Initialization:

1. Clustering all visual primitives $\{v_i\}_{i=1}^N$ into M classes, *e.g.* through k-means clustering, based on the Euclidean distance.
2. Obtaining the visual primitives lexicon $\mathbf{\Omega}$ (represented by $\mathbf{U}$) and the distortion matrix $\mathbf{D}$.
3. Clustering all context groups $\{\mathcal{G}_i\}_{i=1}^N$ into $\tilde{M}$ classes based on the Hamming distance, and obtain the visual phrase lexicon $\mathbf{\Psi}$ (represented by $\tilde{\mathbf{U}}$), as well as the distortion matrix $\tilde{\mathbf{D}}$.

E-step:
The task is to label visual primitives v_i and context groups $\mathcal{G}_i$ with $\mathbf{\Omega}$ and $\mathbf{\Psi}$, namely to update $\mathbf{R}$ and $\tilde{\mathbf{R}}$ given $\mathbf{D}$ and

$\tilde{\mathbf{D}}$, where $\mathbf{D}$ and $\tilde{\mathbf{D}}$ can be directly computed from $\mathbf{U}$ and $\tilde{\mathbf{U}}$ respectively. Based on the analysis above, we need to optimize $\mathbf{R}$ (corresponding to $\mathbf{J}_1$) and $\tilde{\mathbf{R}}$ (corresponding to $\mathbf{J}_2$) *simultaneously* to minimize $\mathbf{J}$, because $\mathbf{J}_1$ and $\mathbf{J}_2$ are correlated.

According to the Hamming distance in Eq. 5, we can derive the matrix form of context groups distortions:

$$\tilde{\mathbf{D}} = -2 \times \tilde{\mathbf{U}}^T\mathbf{T} + \mathbf{1}_T\mathbf{T} + \tilde{\mathbf{U}}^T\mathbf{1}_{\tilde{U}},$$

where $\mathbf{1}_T$ is an $\tilde{M} \times M$ all 1 matrix and $\mathbf{1}_{\tilde{U}}$ is an $M \times N$ all 1 matrix.

Moreover, transaction database $\mathbf{T}$ can be determined by

$$\mathbf{T} = \mathbf{RQ},$$

because each transaction column can be obtained as

$$\mathbf{t}_j = \sum_{i=1}^{N} q_{ij}\mathbf{r}_i{}^1,$$

where q_{ij} is a binary indicator of whether primitive v_i belongs to the context group of v_j, and $\mathbf{r}_i$ denotes the i_{th} column of $\mathbf{R}$ which describe the word label of v_i. Based on the above, we derive Eq. 3 as follows:

$$\begin{aligned}
&\mathbf{J}(\mathbf{D}, \tilde{\mathbf{D}}, \mathbf{R}, \tilde{\mathbf{R}}) = tr(\mathbf{R}^T\mathbf{D}) + \lambda \times tr(\tilde{\mathbf{R}}^T\tilde{\mathbf{D}}) && (6)\\
=\ & tr(\mathbf{R}^T\mathbf{D}) + \\
& \quad \lambda \times tr[\tilde{\mathbf{R}}^T(-2\tilde{\mathbf{U}}^T\mathbf{T} + \mathbf{1}_T\mathbf{T} + \tilde{\mathbf{U}}^T\mathbf{1}_{\tilde{U}})] && (7)\\
=\ & tr(\mathbf{R}^T\mathbf{D}) + \\
& \quad \lambda \times tr[\tilde{\mathbf{R}}^T(-2\tilde{\mathbf{U}}^T\mathbf{RQ} + \mathbf{1}_T\mathbf{RQ} + \tilde{\mathbf{U}}^T\mathbf{1}_{\tilde{U}})] \\
=\ & tr(\mathbf{R}^T\mathbf{D}) + \\
& \quad \lambda \times tr[\tilde{\mathbf{R}}^T(-2(\tilde{\mathbf{U}}^T - \frac{1}{2}\mathbf{1}_T)\mathbf{RQ} + \tilde{\mathbf{U}}^T\mathbf{1}_{\tilde{U}})] \\
=\ & tr(\mathbf{R}^T\mathbf{D}) - 2\lambda \times tr[\tilde{\mathbf{R}}^T(\tilde{\mathbf{U}}^T - \frac{1}{2}\mathbf{1}_T)\mathbf{RQ}] + \\
& \quad \lambda \times tr(\tilde{\mathbf{R}}^T\tilde{\mathbf{U}}^T\mathbf{1}_{\tilde{U}}) && (8)\\
=\ & tr(\mathbf{R}^T\mathbf{D}) - 2\lambda \times tr[\mathbf{Q}^T\mathbf{R}^T(\tilde{\mathbf{U}}^T - \frac{1}{2}\mathbf{1}_T)^T\tilde{\mathbf{R}}] + \\
& \quad \lambda \times tr(\tilde{\mathbf{R}}^T\tilde{\mathbf{U}}^T\mathbf{1}_{\tilde{U}}) && (9)\\
=\ & tr(\mathbf{R}^T\mathbf{D}) - 2\lambda \times tr[\mathbf{R}^T(\tilde{\mathbf{U}}^T - \frac{1}{2}\mathbf{1}_T)^T\tilde{\mathbf{R}}\mathbf{Q}^T] + \\
& \quad \lambda \times tr(\tilde{\mathbf{R}}^T\tilde{\mathbf{U}}^T\mathbf{1}_{\tilde{U}}) && (10)\\
=\ & tr[\mathbf{R}^T(\mathbf{D} - 2\lambda \times (\tilde{\mathbf{U}}^T - \frac{1}{2}\mathbf{1}_T)^T\tilde{\mathbf{R}}\mathbf{Q}^T)] + \\
& \quad \lambda \times tr(\tilde{\mathbf{R}}^T\tilde{\mathbf{U}}^T\mathbf{1}_{\tilde{U}}), && (11)
\end{aligned}$$

where we apply three properties of matrix trace: for square matrix $\mathbf{A}$ and $\mathbf{B}$, we have (1) $tr(\mathbf{A}) = tr(\mathbf{A}^T)$ (Eq. 9), (2) $tr(\mathbf{AB}) = tr(\mathbf{BA})$ (Eq. 10), and (3) $tr(\mathbf{A}) + tr(\mathbf{B}) = tr(\mathbf{A} + \mathbf{B})$ (Eq. 11).

Based on the above analysis, we propose an E-step to iteratively update $\mathbf{R}$ and $\tilde{\mathbf{R}}$ to decrease $\mathbf{J}$. Recall that $\mathbf{R}$ and $\tilde{\mathbf{R}}$ are label indicator matrices constrained by Eq. 4.

1. We first fix $\mathbf{R}$ and update $\tilde{\mathbf{R}}$. Based on Eq. 8, let

   $$\tilde{\mathbf{H}} \triangleq \lambda \times (-2\tilde{\mathbf{U}}^T\mathbf{RQ} + \mathbf{1}_T\mathbf{RQ} + \tilde{\mathbf{U}}^T\mathbf{1}_{\tilde{U}}),$$

   we have

   $$\mathbf{J} = \underbrace{tr(\mathbf{R}^T\mathbf{D})}_{\mathbf{J}_1} + \underbrace{tr(\tilde{\mathbf{R}}^T\tilde{\mathbf{H}})}_{\mathbf{J}_2}. \qquad (12)$$

   Therefore we only need to minimize $\mathbf{J}_2 = tr(\tilde{\mathbf{R}}^T\tilde{\mathbf{H}})$ as $\mathbf{J}_1 = tr(\mathbf{R}^T\mathbf{D})$ is a constant given $\mathbf{R}$ and $\mathbf{U}$. Because each column of $\tilde{\mathbf{R}}$ contains a single 1 (Eq. 4), we update $\tilde{\mathbf{R}}$ to minimize $\mathbf{J}_2$ based on the following criterion, $\forall\ j = 1, 2, ...N$:

   $$\tilde{r}_{ij} = \begin{cases} 1 & i = \arg\min_k \tilde{h}_{kj} \\ 0 & otherwise \end{cases}, \qquad (13)$$

   where $\tilde{h}_{kj}$ is the element of $\tilde{\mathbf{H}}$ and $\tilde{r}_{ij}$ is the element of $\tilde{\mathbf{R}}$. $\tilde{\mathbf{H}}$ can be calculated based on $\mathbf{Q}$, $\tilde{\mathbf{U}}$ and $\mathbf{R}$ which are all given.

2. Similar to the above step, now we fix $\tilde{\mathbf{R}}$ and update $\mathbf{R}$. Based on Eq. 11, let

   $$\mathbf{H} \triangleq \mathbf{D} - 2\lambda \times (\tilde{\mathbf{U}}^T - \frac{1}{2}\mathbf{1}_T)^T\tilde{\mathbf{R}}\mathbf{Q}^T$$

   We get another representation of $\mathbf{J}$:

   $$\mathbf{J} = \underbrace{tr(\mathbf{R}^T\mathbf{H})}_{\mathbf{J}_3} + \underbrace{\lambda \times tr(\tilde{\mathbf{R}}^T\tilde{\mathbf{U}}^T\mathbf{1}_{\tilde{U}})}_{\mathbf{J}_4}, \qquad (14)$$

   where $\mathbf{J}_4 = \lambda \times tr(\tilde{\mathbf{R}}^T\tilde{\mathbf{U}}^T\mathbf{1}_{\tilde{U}})$ is a constant given $\tilde{\mathbf{R}}$ and $\tilde{\mathbf{U}}$. Therefore, only $\mathbf{J}_3$ can be minimized. We update $\mathbf{R}$ to minimize $\mathbf{J}_3$ as follows, $\forall\ j = 1, ...N$:

   $$r_{ij} = \begin{cases} 1 & i = \arg\min_k h_{kj} \\ 0 & otherwise \end{cases}, \qquad (15)$$

   where h_{kj} is the element of $\mathbf{H}$ and r_{ij} is the element of $\mathbf{R}$.

The above E-step itself is an EM-like process because we need to update $\mathbf{R}$ and $\tilde{\mathbf{R}}$ iteratively until $\mathbf{J}$ converges. The objective function $\mathbf{J}$ decreases monotonically at each step.

M-step:
After knowing the labels of visual primitives and visual groups ($\mathbf{R}$ and $\tilde{\mathbf{R}}$), we want to estimate better visual lexicons $\mathbf{\Omega}$ and $\mathbf{\Psi}$. From Eq. 3, $\mathbf{D}$ and $\tilde{\mathbf{D}}$ are not interlaced and thus $\mathbf{U}$ and $\tilde{\mathbf{U}}$ can be optimized separately. We apply the following two steps to update $\mathbf{U}$ and $\tilde{\mathbf{U}}$ separately:

[1]Strictly, $\mathbf{t}_j$ is a binary vector only if it contains distinguishable primitives, *i.e.* each primitive belongs to a different word in $\mathbf{t}_j$. However, our solution is generic and do not need $\mathbf{t}_j$ to be binary, as long as we apply the distortion measure as in Eq. 5.

1. Recalculate the cluster centroid for each visual word class $\{\mathbf{u}_i\}_{i=1}^{M}$ like traditional k-means algorithm, with Euclidean distance. Update $\mathbf{U}$ and $\mathbf{D}$ to decrease $\mathbf{J}$.
2. Recalculate the cluster centroid for each phrase $\{\tilde{\mathbf{u}}_i\}_{i=1}^{\tilde{M}}$, with Hamming distance (see the Appendix for the update details). Update $\tilde{\mathbf{U}}$ and $\tilde{\mathbf{D}}$ to decrease $\mathbf{J}$.

Both of the above steps guarantee that $\mathbf{J}$ is decreasing, therefore the whole M-step decreases $\mathbf{J}$ monotonically. Our method is called a nested-EM algorithm because there are two nested EM processes, where the E-step itself is an EM process. We describe the nested-EM algorithm in Alg. 1.

Algorithm 1: Nested-EM Algorithm.

input : visual primitive database $\mathcal{D} = \{v_i\}$, contextual relations $\mathbf{Q}$, parameters: M, $\tilde{M}$, λ

output : visual word and phrase lexicons: $\mathbf{\Omega}$ and $\mathbf{\Psi}$; clustering results $\mathbf{R}$ and $\tilde{\mathbf{R}}$

Init: (1) clustering visual primitives to get $\mathbf{\Omega}$ and $\mathbf{U}$;
(2) based on $\mathbf{\Omega}$, clustering visual groups to get $\mathbf{\Psi}$ and $\tilde{\mathbf{U}}$;
while $\mathbf{J}$ *is decreasing* **do**
  E-step: fix $\mathbf{U}$ and $\tilde{\mathbf{U}}$, update $\mathbf{R}$ and $\tilde{\mathbf{R}}$
    nested-E step: fix $\mathbf{R}$, update $\tilde{\mathbf{R}}$ (Eq. 13)
    nested-M step: fix $\tilde{\mathbf{R}}$, update $\mathbf{R}$ (Eq. 15)
  if $\mathbf{J}$ *is decreasing* **then**
    goto **E-setp**
  else
    Goto **M-step**
  M-step: fix $\mathbf{R}$ and $\tilde{\mathbf{R}}$, update $\mathbf{U}$ and $\tilde{\mathbf{U}}$ separately.
return $\mathbf{U}$, $\tilde{\mathbf{U}}$, $\mathbf{R}$, $\tilde{\mathbf{R}}$.

Because the solution spaces of $\mathbf{R}$ and $\tilde{\mathbf{R}}$ are discrete and finite, according to the monotonic decreasing of $\mathbf{J}$ at each step of our nested-EM algorithm, we have theorem 1.

Theorem 1 convergence of the nested-EM algorithm *The nested-EM algorithm can converge in finite steps.*

3. Experiments

3.1. Simulation results

To illustrate the idea of our context-aware clustering, we synthesize a spatial dataset for simulation. A concrete example of this spatial dataset can be an image. All the samples have two representations with regarding to (1) feature domain, $\mathbf{f} \in \mathbb{R}^2$ and (2) spatial domain $(x, y) \in \mathbb{N} \times \mathbb{N}$ as shown in Fig. 3 (a) and (b). In our case, we have 5 different types of visual primitives labeled as: '$\square$', '$\star$', 'O', '∇', or '$\times$'. In the spatial domain, $\{\times, \square\}$ is generated together to form a co-occurrent contextual pattern, while $\{O, \star, \nabla\}$ is the other contextual pattern. In the feature domain, each of the 5 clusters has different number of samples and are generated based on Gaussian distributions of different means and variances. Based on the feature domain only, clustering is a challenging problem because some of these Gaussian distributions are heavily overlapped, for example, clusters '$\times$', 'O', and '$\star$' are heavily overlapped. Our tasks are (1) clustering visual primitives into words, and (2) recover the visual phrases $\mathcal{P}_1 = \{\times, \square\}$ and $\mathcal{P}_2 = \{O, \star, \nabla\}$.

We compare the performances of the context-aware clustering with different choices of λ ($\lambda = 0, 400, 2000$) in Fig. 3 (c),(d) and (e), where $\lambda = 0$ gives the same results as the k-means clustering. The major differences of the clustering results appear from the cluster '$\times$'. Although '$\times$' heavily overlaps with clusters 'O' and '$\star$', most of its samples are still correctly labeled based on the help from its spatial context: cluster '$\square$'. For example, although it is difficult to determine a sample v located in the overlapped regions of '$\times$' and 'O' in the feature space, we can resolve the ambiguity by observing the spatial contexts of v. If a '$\square$' is found in its spatial context, then v should be labeled as '$\times$' because discovered visual phrase $\{\square, \times\}$ supports such a label.

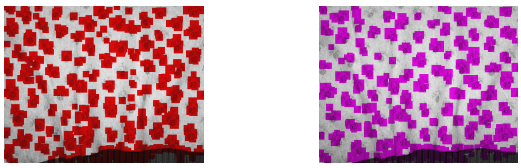
Figure 3(f) shows the iterations of nested-EM algorithm with $\lambda = 2000$. Each iteration corresponds to an individual E-step or an M-step until converge. We decompose the objective function into $\mathbf{J} = \mathbf{J}_1 + \mathbf{J}_2$, where $\mathbf{J}$, $\mathbf{J}_1$, $\mathbf{J}_2$ are the red, black and pink curves respectively. All these three curves are normalized by $\mathbf{J}_{max} = \mathbf{J}^0$, which is the $\mathbf{J}$ value at the initialization step. Compared to k-means clustering which minimizes distortions $\mathbf{J}_1$ in feature space only, our context-aware clustering sacrifices $\mathbf{J}_1$ to gain larger decrease of distortion $\mathbf{J}_2$ in the context space, which gives a smaller total distortion $\mathbf{J}$. The error rate curve (blue) describes the percentage of samples that are wrongly labeled at each step, and we notice it decreases consistently with our objective function $\mathbf{J}$.

In terms of clustering errors, the context-aware clustering ($e = 0.12$ when $\lambda = 2000$) performs significantly better than the k-means method ($e = 0.25$). The parameter λ balances the two clustering criteria: (1) clustering based on visual features $\mathbf{f}$ ($\mathbf{J}_1$) and (2) clustering based on spatial contexts ($\mathbf{J}_2$). The smaller the λ, the more faithful the clustering results follow the feature space, where samples have similar features are grouped together. An extreme case is $\lambda = 0$ when no regularization is applied in Eq. 3 by ignoring the feedback from contexts. In such a case, our context-aware clustering is equal to k-means clustering. On the other hand, a larger λ favors the clustering results that support the discovered context patterns (*e.g.* visual phrases), thus samples have similar contexts are more likely to be grouped together.

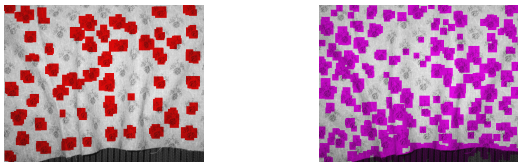
3.2. Image texton discovery

To validate whether the discovered visual phrases can really capture common spatial image patterns [13], we test a

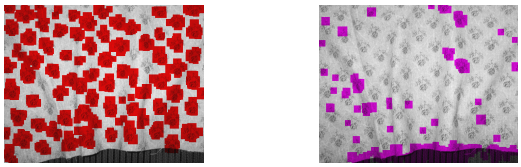
collection of texture images[2]. and an example is presented in Fig. 3.2 Given an image, we first detect SIFT points [7] and treat keypoints of scale ranges between 1 and 2 as visual primitives: $\mathcal{D}_v = \{v_i\}$. We apply spatial K-NN groups to build the group databases $\mathcal{G}$, with $K = 10$. We let $\lambda = \tau \mathbf{J}_1^0/\mathbf{J}_2^0$, where $\mathbf{J}_1^0$ and $\mathbf{J}_2^0$ are the initialization value of $\mathbf{J}_1$ and $\mathbf{J}_2$ respectively; $\tau > 0$ is the parameter to balance the distortions between SIFT features (word level) and contextual patterns (phrase level).



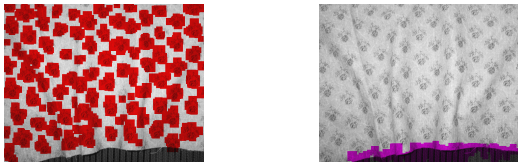
k-means clustering of visual primitives (k=2).



Context-aware clustering: initialization of visual phrases.



Context-aware clustering: after the 1_{st} full EM iteration.



Context-aware clustering: final results (19 full EM iter).

Figure 2. The 1_{st} row shows 2 visual words (red and purple) formed through k-means clustering. From the 2_{nd} to 4_{th} row, we show 2 visual phrases (red and purple) discovered through context-aware clustering. There exist two types of near-regular textures in the image. One is the flower pattern located in the clothes (with deformations) and the other is the regular textures located in the right bottom. We notice that k-means clustering of visual primitives cannot distinguish from two different textures. Parameters used are $M = 25$, $\tilde{M} = 2$, $\tau = 0.5$.

For an image of size 1024×768 and containing 1000-2000 visual primitives, the nested-EM algorithm can converge within 40 full EM iterations. It is interesting to notice that the discovered visual phrases are of spatial structures, such as flowers in Fig. 3.2. In Fig. 3.2, we also show how our nested-EM algorithm corrects the imperfect clustering results iteratively, by using the spatial contextual information as the feedback. In comparison, conventional k-means clustering cannot obtain satisfactory results if clustering visual primitives individually.

[2]Images are from source: http://vivid.cse.psu.edu/texturedb/gallery/. Please see supplementary materials for more results

3.3. Multiple-view clustering

Multiple-view clustering is another typical application of our context-aware clustering algorithm. In multiple-view clustering [15], each data sample $v = \{\mathbf{f}_1, \mathbf{f}_2, ..., \mathbf{f}_c\}$ is represented by different types of features $\mathbf{f}_i$. Our task is to cluster a collection of data samples $\mathcal{D}_v = \{v_j\}_{j=1}^N$. A simple solution is to concatenate all features into a long feature vector $\mathbf{f} = \mathbf{f}_1 \cup \mathbf{f}_2 \cup ... \cup \mathbf{f}_c$. Then we can perform traditional clustering in the new formed feature space $\mathbf{f}$. However, because the depedent information among different types of features are not well utilized, such a simple solution cannot get satisfied results.

We select the multiple features data set from the UCI Machine Learning Repository for evaluation. This multi-class data set consists of handwritten numerals ('0'–'9') extracted from a collection of Dutch utility maps. Each class contains 200 data samples and the data set has 2000 digits in total. Each digit is represented in terms of the 6 features and we select 3 of them for context-aware clustering: (1) 76 Fourier coefficients of the character shapes (*fou*); (2) 64 Karhunen-Loeve coefficients (*kar*) and (3) 240 pixel averages in 2 x 3 windows (*pix*). A data sample v thus generates 3 primitives $\mathbf{f}_{fou}$, $\mathbf{f}_{kar}$ and $\mathbf{f}_{pix}$ in 3 feature spaces respectively. As a result, we obtain in total $N = 2000 \times 3$ primitives and cluster them for word lexicon. The original data sample v now corresponds to a context group $\mathcal{G}$ which characterizes the co-occurrent dependency among different types of features $\mathbf{f}_i$. Each $v \in \mathcal{D}_v$ generates to a transaction $\mathbf{t}$ of length 3. We further cluster these $n = 2000$ transactions into phrases. In the initialization step, we build the word lexicon $\mathbf{\Omega}_i$ ($|\mathbf{\Omega}_i| = 10$, $i = fou, kar, pix$) for three types of features separately and obtain a final lexicon $\mathbf{\Omega} = \mathbf{\Omega}_{fou} \cup \mathbf{\Omega}_{kar} \cup \mathbf{\Omega}_{pix}$ ($|\mathbf{\Omega}| = 30$). The phrase lexicon $\mathbf{\Psi}$ is then constructed based on $\mathbf{\Omega}$. By considering both distortions from 3 individual features and their contextual patterns, the objective function in Eq. 3 now becomes:

$$\begin{aligned} \mathbf{J} &= \sum_{i=1}^{3} tr(\mathbf{R}_i^T \mathbf{D}_i) + \lambda \times tr(\tilde{\mathbf{R}}^T \tilde{\mathbf{D}}) \\ &= tr(\mathbf{R}^T \mathbf{D}) + \lambda \times tr(\tilde{\mathbf{R}}^T \tilde{\mathbf{D}}), \end{aligned}$$

where $\mathbf{R}$ and $\mathbf{D}$ are matrices containing 3 diagonal blocks corresponding to *fou*, *kar* and *pix* features respectively. With a straight forward adjustment of matrices sizes in Table 1, we can still apply the nest-EM algorithm in Alg. 1.

Table 2 compares conventional k-means clustering with our context-aware clustering. Specifically, we try k-means

clustering in 3 features individually, and also in a concatenation of 3 features. In each case, k-means clustering is repeated 100 times and the best result with minimum total distortion is selected for comparison. In context-aware clustering, to balance between the data fidelity in feature space ($\mathbf{J}_1$) and the influence of contextual information $\mathbf{J}_2$, we select $\tau = 1$, which results in $\lambda = \tau \mathbf{J}_1^0 / \mathbf{J}_2^0 = 1061.7$. From Table 2, we can see that although each individual feature has limited ability in clustering, the contextual pattern among them can help to improve the clustering results significantly. Also our context-aware clustering performs better (with error 13.5%) than simply concatenating all features for k-means clustering (with error 17.9%).

Table 2. multiple feature clustering: comparison between traditional k-means clustering and context-aware clustering.

	# feature	# class	error
k-means (fou)	76	k=10	27.5%
k-means (kar)	64	k=10	26.9%
k-means (pix)	240	k=10	26.9%
k-means (all)	76+64+240	k=10	17.9%
context-aware	76+64+240	$\tilde{M} = 30; M = 10$	13.5%

4. Conclusion

We present in this paper a new formulation of self-supervised clustering, called context-aware clustering, and show how contextual information can feed back to improve the clustering results. Two kinds of contextual information (1) spatial contexts of visual primitives and (2) contextual patterns among different types of features are applied to improve the clustering results in (1) image texton discovery and (2) multiple view clustering of hand written digits, respectively. Compared with traditional k-means clustering, our context-aware clustering considers the data (or feature) dependency in a higher level. Thus it not only gets better clustering results, but also can reveal the hidden structures among data samples.

The proposed nested-EM algorithm is an efficient iterative solution for the context-aware clustering and is proved to converge. It provides a general solution to context-aware clustering. Besides spatial contexts and feature contexts proposed in our experiments, other types of contextual information can also be incorporated.

Appendix

We discuss how to update the prototypes of visual phrases ($\tilde{\mathbf{U}}$) in the M-step. Given a cluster of $M \times 1$ transactions $\mathcal{X} = \{\mathbf{t}_i\}_{i=1}^n$, our target is to find their centroid $\tilde{\mathbf{u}} \in \{0,1\}^M$ such that the total quantization distortions are minimized under the Hamming distance criterion in Eq. 5. The optimization problem is formulated as:

$$\min_{\tilde{\mathbf{u}} \in \{0,1\}^M} \sum_{i=1}^{n} \left[M - (\mathbf{t}_i^T \tilde{\mathbf{u}} + (\mathbf{1} - \mathbf{t}_i^T)(\mathbf{1} - \tilde{\mathbf{u}})) \right].$$

Let $\tilde{\mathbf{u}}^k$ denote the k_{th} element of $\tilde{\mathbf{u}}$, we minimize the following objective function by using the Lagrangian and let $\lambda^t \geq 0$ to obtain the unique maximum solution:

$$f(\tilde{\mathbf{u}}, \lambda) = \sum_{i=1}^{n} \sum_{t=1}^{M} \left[2\mathbf{t}_i^k \tilde{\mathbf{u}}^k - \mathbf{t}_i^k - \tilde{\mathbf{u}}^k + 1 + \lambda^k \tilde{\mathbf{u}}^k (1 - \tilde{\mathbf{u}}^k) \right].$$

By applying $\frac{\partial f(\tilde{\mathbf{u}}, \lambda)}{\partial \tilde{\mathbf{u}}^k} = 0, \quad \frac{\partial f(\tilde{\mathbf{u}}, \lambda)}{\partial \lambda^k} = 0, \quad \lambda^k \geq 0, \quad \forall k,$ we obtain

$$\tilde{\mathbf{u}}^k = \frac{1}{2}\left(\frac{2\sum_{i=1}^{n} \mathbf{t}_i^k - n}{\lambda^k} + 1\right), \quad \forall k,$$

and

$$\lambda^k = |2\sum_{i=1}^{n} \mathbf{t}_i^k - n| \geq 0.$$

Finally, we have

$$\tilde{\mathbf{u}}^k = \frac{sgn(2\sum_{i=1}^{n} \mathbf{t}_i^k - n) + 1}{2},$$

where $sgn(a) = 1$ if $a \geq 0$, and $sgn(a) = -1$ if $a < 0$.

Acknowledgment

This work was supported in part by National Science Foundation Grants IIS-0347877 and IIS-0308222.

References

[1] J. Amores, N. Sebe, and P. Radeva. Context-based object-class recognition and retrieval by generalized correlograms. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 29(10):1818–1833, 2007.

[2] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2002.

[3] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proc. of Intl. Conf. on Machine Learning*, 1998.

[4] O. Boiman and M. Irani. Similarity by composition. In *Proc. of Neural Information Processing Systems*, 2006.

[5] A. Frome, Y. Singer, and J. Malik. Image retrieval and classification using local distance functions. In *Proc. of Neural Information Processing Systems*, 2006.

[6] H. Jegou, H. Harzallah, and C. Schmid. A contextual dissimilarity measure for accurate and efficient image search. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2007.

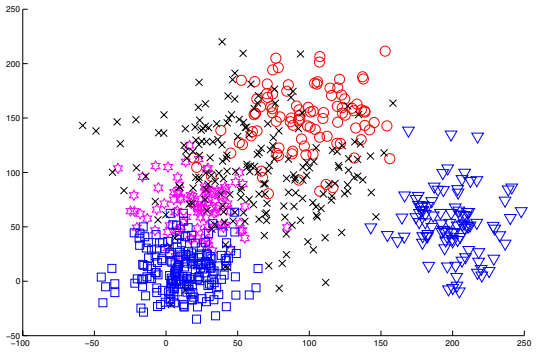
[7] D. Lowe. Distinctive image features from scale-invariant keypoints. *Intl. Journal of Computer Vision*, 2004.

[8] Q. Mei, D. Xin, H. Cheng, J. Han, and C. Zhai. Generating semantic annotations for frequent patterns with context analysis. In *Proc. ACM SIGKDD*, 2006.

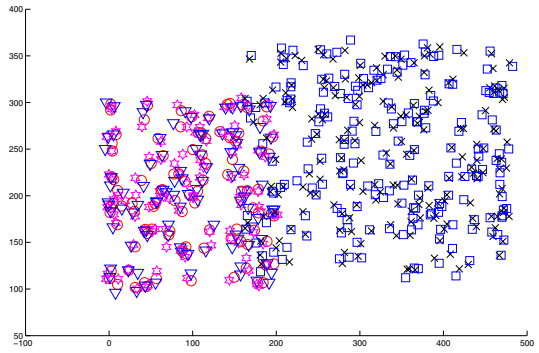
[9] E. Shechtman and M. Irani. Matching local self-similarities across images and videos. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2007.

[10] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell. Distance metric learning, with application to clustering with side-information. In *Proc. of Neural Information Processing Systems*, 2002.

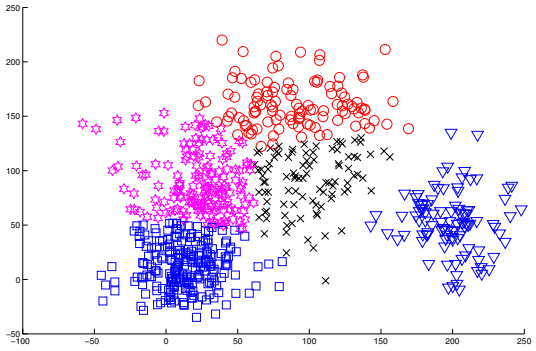
[11] J. Ye, Z. Zhao, and H. Liu. Adaptive distance metric learning for clustering. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2007.



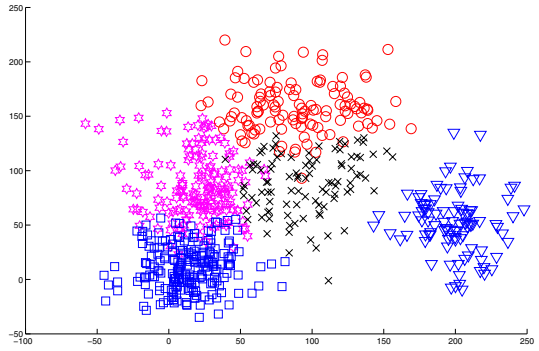
(a) Feature Domain



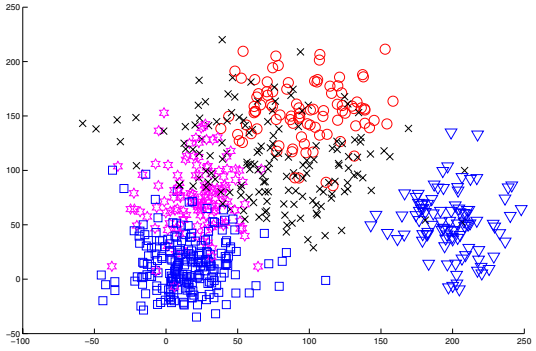
(b) Spatial Domain



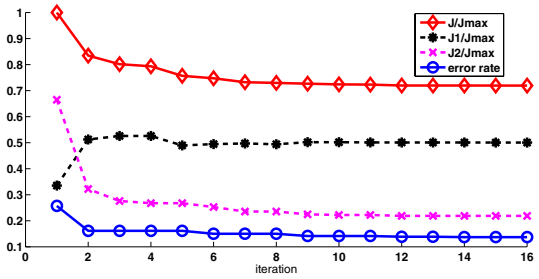
(c) k-means Clustering ($k = 5$)



(d) Context-Aware Clustering ($\lambda = 400$)



(e) Context-Aware Clustering ($\lambda = 2000$)



(f) Performance ($\lambda = 2000$)

Figure 3. Context-aware clustering on the synthesized spatial data and the comparison with the k-means algorithm. Parameter used are $M = 5$, $\tilde{M} = 2$ and $\epsilon = 100$ in searching for ϵ-NN spatial groups. See texts for descriptions. Best seen in color.

[12] S. X. Yu and J. Shi. Segmentation given partial grouping constraints. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2004.

[13] J. Yuan and Y. Wu. Spatial random partition for common visual pattern discovery. In *Proc. IEEE Intl. Conf. on Computer Vision*, 2007.

[14] J. Yuan, Y. Wu, and M. Yang. Discovery of collocation patterns: from visual words to visual phrases. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2007.

[15] D. Zhou and C. J. Burges. Spectral clustering and transductive learning with multiple views. In *Proc. Intl. Conf. on Machine Learning*, 2007.

[16] S.-C. Zhu, C. en Guo, Y. Wang, and Z. Xu. What are textons? *Intl. Journal of Computer Vision*, 2005.