Clustering Spatial Data with a Hybrid EM Approach

Tianming Hu *and Sam Yuan Sung
Department of Computer Science, National University of Singapore
Singapore 117543

Abstract: In spatial clustering, in addition to the object similarity in the normal attribute space, similarity in the spatial space needs to be considered and objects assigned to the same cluster should usually be close to one another in the spatial space. The conventional EM algorithm is not suited for spatial clustering because it does not consider spatial information. Although Neighborhood EM (NEM) algorithm incorporates a spatial penalty term to the criterion function, it involves much more iterations in every E-step. In this paper, we propose a Hybrid EM (HEM) approach that combines EM and NEM. Its computational complexity for every pass is between EM and NEM. Experiments also show that its clustering quality is better than EM and comparable to NEM.

Keywords: expectation maximization algorithm, Gaussian mixture, spatial penalty term, penalized likelihood, spatial clustering, spatial autocorrelation

Originality and Contribution: The training of mixture models with EM algorithm is generally slow and does not account for spatial information. Although incorporating a spatial penalty term in the cost function, Neighborhood EM algorithm is more expensive computationally, because it needs multiple iterations in every E-step. To cluster spatial data efficiently using mixture models, we develop a hybrid EM approach, where training is first performed via a selective hard EM we originally propose, then switches to NEM that needs to run only one iteration of E-step in every pass. Thus, we keep maximizing the penalized likelihood criterion, but computational complexity of every pass is kept comparable to EM. The proposed algorithm can be used in various spatial clustering problems, as long as neighborhood information is provided. Such problems include clustering spatial econometric data which are probably irregularly located, spatial contextual classification of remote sensing images, and general image segmentation.

1 Introduction

Spatial data distinguish themselves from conventional data in that associated with each object, the attributes under consideration include not only non-spatial normal attributes, but also spatial attributes that are unique or emphasized and describe the object's spatial information such as location and shape in the 2-D spatial space. Independent and identical distribution (iid), a fundamental assumption often made in pattern recognition, is no longer valid for spatial data. In practice, almost every site is related to its neighbors. For example, houses in nearby neighborhoods tend to have similar prices which are affected by one another. As for identical assumption, often different regions have different distributions, which is referred to as spatial heterogeneity. Spatial data often exhibit positive autocorrelation in that nearby sites tend to have similar characteristics and thus exhibit spatial continuity. In remote sensing images, close pixels usually belong to the same landcover type: soil, forest, etc.

Therefore, in spatial clustering, in addition to the object similarity in the normal attribute space, similarity in the spatial space needs to be considered and objects assigned to the same cluster should also be close to one another in the spatial space. In this paper, using mixture models, we propose a Hybrid Expectation Maximization (HEM) approach to spatial clustering, which combines EM algorithm [5] and Neighborhood EM algorithm (NEM) [1]. Our algorithm provides the following advantages: (1) In terms of clustering quality, HEM is better than EM and comparable to NEM. (2) In terms of computational complexity, HEM is between EM and NEM.

*Corresponding author. Email: tmhu@yahoo.com.

The rest of the paper is organized as follows. In the remainder of this section, we formulate the spatial clustering problem and review related work. Basics of EM are introduced in Section 2, followed by NEM introduced in Section 3. We present our HEM approach in Section 4. Experimental evaluation is reported in Section 5 where several real datasets are used for comparison. Finally Section 6 concludes this paper with a summary and discussion on future work.

1.1 Problem Formulation

The goal of spatial clustering is to partition data into groups or clusters so that pairwise dissimilarity, in both attribute space and spatial space, between those assigned to the same cluster tend to be smaller than those in different clusters. Let S denote the set of locations, e.g., the set of triple (index, latitude, longitude). Spatial clustering can be formulated as an unsupervised classification problem. We are given a spatial framework of n sites,$S = \{s_i\}_{i=1}^n$ with a neighbor relation $N \subseteq S \times S$. Sites s_i and s_j are neighbors iff $(s_i, s_j) \in N, i \neq j$. Let $N(s_i) \equiv \{s_j : (s_i, s_j) \in N\}$ denote the neighborhood of s_i. We assume N is given by a contiguity matrix W whose $W(i,j) = 1$ iff $(s_i, s_j) \in N$ and $W(i,j) = 0$ otherwise. Associated with each s_i, there is a d-D feature vector of normal attributes $\mathbf{x}_i \equiv \mathbf{x}(s_i) \in \Re^d$. We need to find a many-to-one mapping $f : \{\mathbf{x}_i\}_{i=1}^n \to \{1, ..., K\}$. If each object $\mathbf{x}_i$ has a true class label $y_i \in \{1, ..., K\}$, naturally the ultimate goal is to maximize similarity between obtained clustering and true classification. However, since the class information is unavailable during learning, the objective in practice is to optimize some criterion function such as likelihood. Besides, spatial clustering imposes the following constraint of spatial autocorrelation. y_i is not only affected by $\mathbf{x}_i$, but also by $(\mathbf{x}_j, y_j)$ of its neighbors $N(s_i)$. Hence it is more appropriate to model the distribution of y_i with $P(y_i|\mathbf{x}_i, \{(\mathbf{x}_j, y_j) : s_j \in N(s_i)\}$ instead of $P(y_i|\mathbf{x}_i)$.

1.2 Related Work

Most clustering methods in the literature treat each object as a point in the high dimensional space and do not distinguish spatial attributes from normal attributes. Mainly developed in the data mining field for large datasets, they can be divided into the following categories: distance-based [23], density-based [6],hierarchy-based [14], etc.

For spatial clustering, some methods only handle 2-D spatial attributes [7] and deal with problems like obstacles which are unique in spatial clustering [27]. Others incorporate spatial information in the clustering process. They include: (1) Adding spatial information into dataset [13, 11]. (2) Modifying existing algorithms, e.g., allowing an object assigned to a class if and only if this class already contains its neighbor [17]. (3) Selecting a model that encompasses spatial information [1]. This can be achieved by modifying a criterion function that includes spatial constraints [26], which mainly comes from the image analysis where Markov random field is intensively used [9].

Clustering using mixture models with EM can be regarded as a soft k-means algorithm [15] in that the output is posterior probability rather than hard classification. It does not account for spatial information and usually cannot give satisfactory performance on spatial data. NEM extends EM by adding a spatial penalty term in the criterion, but this makes it need more iterations in each E-step. If further information about structure is available, the structural EM algorithm may be used to learn Bayesian networks for clustering [8, 24]. In our case, we assume that soft constraints can be derived with locations of sites. A relevant problem is semi-supervised clustering, where some pairs of instances are known belonging to same or different clusters [2]. In their case, the goal is to fit the mixture model to the data while minimizing the violation of hard constraints.

2 Basics of EM

A finite mixture model of K components has the form in Eq. (1), where $f_k(\mathbf{x}|\theta_k)$ is k-th component's probability density function (pdf) with parameters θ_k, π_k is k-th component's prior probability . Φ denotes the set of all parameters and in the case of Gaussian mixture we use in this paper, it includes $\{\pi_k, \mu_k, \Sigma_k\}_{k=1}^K$. Given a set of data $\{\mathbf{x}_i\}_{i=1}^n$, the sample log likelihood function is defined in Eq. 2 where independence among data is implied.

$$f(\mathbf{x}|\Phi) = \sum_{k=1}^{K} \pi_k f_k(\mathbf{x}|\theta_k) \quad (1)$$

$$L(\Phi) = \sum_{i=1}^{n} \ln \left[\sum_{k=1}^{K} \pi_k f_k(\mathbf{x}_i|\theta_k) \right] \quad (2)$$

EM tries to iteratively maximize L in the context of missing data where each $\mathbf{x}$ is now augmented with a missing value $y \in \{1, ..., K\}$ indicating which component it comes from, i.e., $p(\mathbf{x}|y = k) = f_k(\mathbf{x}|\theta_k)$. Essentially, it produces a sequence of estimate $\{\Phi^t\}$, from an initial estimate Φ^0 and consists of two steps:

- E-step: Evaluate Q, the conditional expectation of log likelihood of the complete data $\{\mathbf{x}, y\}$ in Eq. 3, where $E_{\overline{P}}[\cdot]$ denotes the expectation w.r.t. the distribution $\overline{P}$ over y and in this case we set $\overline{P}(y) = P_{\Phi^{t-1}}(y) \equiv P(y|\mathbf{x}, \Phi^{t-1})$.

$$Q(\Phi, \Phi^{t-1}) \equiv E_{\overline{P}}[\ln(P(\{\mathbf{x}, y\}|\Phi))] \quad (3)$$

$$= E_{P_{\Phi^{t-1}}}[\ln(P(\{\mathbf{x}, y\}|\Phi))] \quad (4)$$

- M-step: Set $\Phi^t = \text{argmax}_\Phi Q(\Phi, \Phi^{t-1})$. M-step can be obtained in closed form.

In M-step, EM directly maximizes Q instead of L. We can easily prove $L(\Phi^t) \geq L(\Phi^{t-1})$ from an entropy-based viewpoint, by rewriting Q as

$$Q(\Phi, \Phi^{t-1}) = L(\Phi) - \sum_{i=1}^{n} \sum_{k=1}^{K} P_{\Phi^{t-1}}(y_i = k) \ln \left(\frac{1}{P_{\Phi^{t-1}}(y_i = k)} \frac{P_{\Phi^{t-1}}(y_i = k)}{P_\Phi(y_i = k)} \right) \quad (5)$$

$$= L(\Phi) - \sum_{i=1}^{n} [H(P_{\Phi^{t-1}}(y_i)) + D(P_{\Phi^{t-1}}(y_i) \| P_\Phi(y_i))] \quad (6)$$

In Eq. (6) $H(P_{\Phi^{t-1}}(y_i))$ is the entropy of the distribution $P_{\Phi^{t-1}}(y_i)$ and $D(P_{\Phi^{t-1}}(y_i) \| P_\Phi(y_i))$ is the Kullback-Liebler divergence [16] between two distributions $P_{\Phi^{t-1}}(y_i)$ and $P_\Phi(y_i)$. Thus $L(\Phi^t) \geq L(\Phi^{t-1})$ can be easily proved from Eq. (6) with theorems in information and coding theory [19].

Following [21], other variants of EM such as incremental and sparse ones that partially implement E-step, can be justified in terms of a function F defined in Eq. (7), where $\overline{P}$ denotes a set of distributions $\{\overline{P}(y_i)\}$ and $H(\overline{P})$ denotes $\sum_{i=1}^{n} H(\overline{P}(y_i))$.

$$F(\overline{P}, \Phi) \equiv E_{\overline{P}}[\ln(P(\{\mathbf{x}, y\}|\Phi))] + H(\overline{P}) \quad (7)$$

$$= -D(\overline{P} \| P_\Phi) + L(\Phi) \quad (8)$$

By setting $\overline{P} = P_{\Phi^{t-1}}$ in Eq. (7) and noting that $E_{P_{\Phi^{t-1}}}[\ln(P(\{\mathbf{x}, y\}|\Phi))] = Q(\Phi, \Phi^{t-1})$, we can easily derive Eq. 8 from Eq. 6. Then EM is equivalent to the following two steps that alternately maximize F w.r.t. its two parameters, starting with an initial estimate $(\overline{P}^0, \Phi^0)$.

- E-step: Set $\overline{P}^t = \text{argmax}_{\overline{P}} F(\overline{P}, \Phi^{t-1})$. It can be shown that F is maximized by $\overline{P}^t = P_{\Phi^{t-1}}$. In that case, $F(P_{\Phi^{t-1}}, \Phi^{t-1}) = L(\Phi^{t-1})$, which is obvious from Eq. 8.

- M-step: Set $\Phi^t = \text{argmax}_\Phi F(\overline{P}^t, \Phi)$. It is exactly the same as M-step in EM, because $H(\overline{P})$ does not depend on Φ.

3 Neighborhood EM

To incorporate spatial information, we can add a penalty term to F that consists of $\overline{P}(y)$ for all sites. Intuitively, F will be maximized if nearby sites from the same class have similar $\overline{P}(y)$. Proposed in NEM [1], the dot product penalty is defined in Eq. (9), where $\overline{P}_{ik}$ denotes$\overline{P}(y_i = k)$ and $\overline{\mathbf{P}}(y_i)$ in Eq. (10) denotes a column vector $[\overline{P}_{i1}, ..., \overline{P}_{iK}]$. The matrix formed by $[\overline{\mathbf{P}}(y_1), ..., \overline{\mathbf{P}}(y_n)]$ can be regarded as a fuzzy classification matrix [12]. The new criterion U is given in Eq. (11) where $\beta > 0$ is a fixed coefficient. Actually, U can also be directly derived based on the hidden Markov random field, which has been used to provide a principled framework for incorporating supervision into semi-supervised clustering [2]. Under certain assumptions, maximizing U is equivalent to maximizing the maximum a-posteriori probability of the field.

$$G(\overline{P}) \equiv \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{K} W(i,j)\overline{P}_{ik}\overline{P}_{jk} \tag{9}$$

$$= \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} W(i,j)\overline{\mathbf{P}}(y_i)\cdot\overline{\mathbf{P}}(y_j) \tag{10}$$

$$U(\overline{P}, \Phi) \equiv F(\overline{P}, \Phi) + \beta G(\overline{P}) \tag{11}$$

Similar to F, U can be maximized by alternately estimating its two parameters. With $\overline{P}$ fixed, M-step can be solved analytically. In E-step where Φ is fixed, if U is maximized at $\overline{P}^*$, then $\overline{P}_{ik}$ satisfies Eq. (12), which can be organized as $\overline{P}^* = O(\overline{P}^*)$ to include all parameters in $\overline{P}^*$. It is proven in [1] that under certain conditions, the sequence produced by $\overline{P}^m = O(\overline{P}^{m-1})$ will converge to a fixed point to maximize U. Hence $\overline{P}^*_{ik}$ can be regarded as dot product again between the estimation from its own $\mathbf{x}$ and the estimation $P(y_i|N(s_i))$ from its neighbors.

$$\overline{P}^*_{ik} = \frac{\pi_k f_k(\mathbf{x}_i|\theta_k)\exp\left(\beta\sum_{j=1}^{n} W(i,j)\overline{P}^*_{jk}\right)}{\sum_{l=1}^{K}\pi_l f_l(\mathbf{x}_i|\theta_l)\exp\left(\beta\sum_{j=1}^{n} W(i,j)\overline{P}^*_{jl}\right)} \tag{12}$$

Let us analyze in more detail the distribution $P(y_i|N(s_i))$ provided by neighbors, which undergoes two-phase smoothing in Eq. (12). The first smoothing is realized by summing up over neighbors. Then, to make it a legal probability, we apply the softmax function parameterized in β. The default value of β is 1 so that the elements' size relations of the output vector are usually intact after transfer. The authors of NEM also recommend setting $\beta \in [0.5, 1]$. Sometimes, however, a larger β may be needed if we want to magnify the impact of neighbors and strengthen the winner class. More discussion is provided in the experiment part.

4 Hybrid EM

EM is not appropriate for spatial clustering because it does not account for spatial information. In contrast, although NEM incorporates spatial information, it requires more iterations in each E-step where more computation is performed to combine estimates from neighbors.

To avoid additional computation and still achieve satisfactory results on spatial data, we propose HEM, which is based on the following observation. In early passes of EM when L grows rapidly, U also grows and clustering performance increases too. U begins to decrease when the growth of L slows down and EM begins to converge. Such phenomenon seldom happens in NEM where clustering performance generally increases with U. This motivates us to train first using EM and turn to NEM only when U begins to decrease. Furthermore, empirical results show that we need to run E-step only once in NEM. Such hybrid training enables our algorithm to involve much less computation than NEM and still keep U never decreasing.

We define a *kernel site* s_i as one whose largest $\overline{P}(y)$ comes from the same class as all its neighbors' do, i.e., $\exists k, \forall s_j \in \{s_i\}\bigcup N(s_i), \overline{P}_{jk} = \max_l\{\overline{P}_{jl}\}$. For early training we employ a selective hard variant (winner-take-all) of EM that stands midway between k-means and EM. After E-step of EM, we transfer $\overline{P}(y)$ for those

kernel sites into a hard distribution where all values receive zero probability except one value that is the winner (largest) in $\overline{P}(y)$. The motivation is that in our case of spatial clustering, if spatial continuity exists, which is often the case, most sites would be surrounded by sites from the same class. Therefore, if the mixture model fits the data quite well and one site and all its neighbors have been classified into the same class, this classification would probably be correct. Of course, such an EM variant cannot, in general, converge to the unconstrained maximum of F, even after finding Φ that maximizes F in the subsequent M-step. Nevertheless, there are computational advantages to using this variant in early training until convergence and switching to another variant that is able to find the unconstrained maximum [3]. After all, if we know which component data come from, ideally we should use data for that component only.

After such a selective hard EM cannot increase U any longer, we can fix $\overline{P}$ for those kernel sites and need not re-estimate them, for we have more confidence in the classification of the present kernel sites. As demonstrated later, with proper implementation, the computation in every pass in later NEM can be saved even more by $|S_f|/n$, where S_f denotes the set of fixed sites and n is the total data size.

In detail, with pre-specified β and m (the number of iterations of E-step in NEM and set to 1 in our algorithm), HEM is carried out as follows with U as criterion function, starting with initial estimate $(\overline{P}^0, \Phi^0)$.

1. Selective Hard EM
   (a) E-step:
      i. Set $\overline{P}^t = \text{argmax}_{\overline{P}} F(\overline{P}, \Phi^{t-1})$, i.e., $\forall i, k$

$$\overline{P}^t_{ik} = \frac{\pi^{t-1}_k f_k(\mathbf{x}_i|\theta^{t-1}_k)}{\sum_{l=1}^K \pi^{t-1}_l f_l(\mathbf{x}_i|\theta^{t-1}_l)} \tag{13}$$

      ii. Transform $\overline{P}^t$ into a hard distribution for those kernel sites, i.e., for kernel site s_i, set $\overline{P}^t_{ik} = 1$ iff $\overline{P}^t_{ik} = \max_l \overline{P}^t_{il}$ and set $\overline{P}^t_{ik} = 0$ otherwise.
   (b) M-step: Set $\Phi^t = \text{argmax}_\Phi F(\overline{P}^t, \Phi)$.
   (c) Check: If $U^t \leq U^{t-1}$, go to NEM with $(\overline{P}^{t-1}, \Phi^{t-1})$, otherwise go back to E-step in EM.
2. Fix(optional)
   Fix $\overline{P}$ (binary at present) for those kernel sites S_f. We no long update $\overline{P}(y_i), s_i \in S_f$.
3. NEM
   (a) E-step: Set $\overline{P}^t = \text{argmax}_{\overline{P}} U(\overline{P}, \Phi^{t-1})$ by applying Eq. (12) $m = 1$ times. If fixing option is used, then apply Eq. (12) just for those $\overline{P}(y_i)$ whose $s_i \notin S_f$.
   (b) M-step: Set $\Phi^t = \text{argmax}_\Phi U(\overline{P}^t, \Phi)$. This step is exactly the same as the M-step in EM.

We have another option on when to turn. Instead of monitoring U, we can check G after E-step in EM and turn to NEM if G decreases, for G depends only on $\overline{P}$ and M-step does not change it. This would make the training turn earlier to NEM, for the increase in F may cancel the decrease in G and thus still keeps U growing. After training, $\mathbf{x}_i$ is assigned to the class k with the maximum posterior $\overline{P}_{ik}$.

4.1 Selective Hardening

Hardening $\overline{P}$ for those kernel sites can be justified if we decompose U as $U = \sum_{i=1}^n U_i(\overline{P}_i, \Phi)$ and $U_i(\overline{P}_i, \Phi)$ has the following form

$$\begin{aligned} U_i(\overline{P}_i, \Phi) &\equiv E_{\overline{P}_i}[\ln(P(\{\mathbf{x}_i, y_i\}|\Phi))] + H(\overline{P}_i) + \beta G(\overline{P}_i) & (14)\\ &= \sum_{k=1}^K \overline{P}_{ik} \ln(\pi_k f_k(\mathbf{x}_i|\theta_k)) + H(\overline{P}_i) + \frac{1}{2}\beta \sum_{s_j \in N(s_i)} \overline{\mathbf{P}}(y_i) \cdot \overline{\mathbf{P}}(y_j) & (15) \end{aligned}$$

Suppose that before hardening, the largest $\overline{P}(y)$ of the kernel site s_i and all its neighbors come from class k, we can derive the change of U_i after hardening as

$$\sum_{l\neq k}\overline{P}_{il}\ln(\frac{\pi_k f_k(\mathbf{x}_i|\theta_k)}{\pi_k f_k(\mathbf{x}_i|\theta_k)}) - H(\overline{P_i}) + \frac{1}{2}\beta\sum_{s_j\in N(s_i)}\sum_{l\neq k}\overline{P}_{il}(\overline{P}_{jk}-\overline{P}_{jl}) \tag{16}$$

If the mixture model fits the data quite well, usually $\overline{P}(y_i)$ would not be far away from $P_\Phi(y_i)$ and this implies that $P_\Phi(y_i = k) = \max_l\{P_\Phi(y_i = l)\}$, so every term in the first summation of Eq. 16 is positive. Apparently, the third summation is also positive. Because hard distribution's entropy is zero, the only negative term is the second term $-H(\overline{P_i})$. Considering s_i is a kernel site, its $\overline{P}(y)$ must be quite stable, which means its $H(\overline{P_i})$ is small. Therefore, after hardening, U_i would probably grow or at least would not decrease much.

4.2 M-step Implementation for Fixing Option

After fixing and switching to NEM, those fixed sites' $\overline{P}(y)$ are no longer updated in E-step of NEM, so the computational complexity of E-step is reduced to $O(n-|S_f|)$. However, if we perform M-step the usual way as in Eqs. (17, 19, 21), every site is still visited once. This problem can be circumvented if we decompose every formula of M-step into two parts, fixed and unfixed, as in Eqs. (18, 20, 22). All terms in the fixed part, such as $\sum_{s_i\in S_f}\overline{P}^t_{ik}\mathbf{x}_i$, are computed only once on the turn and kept fixed in later NEM. We only need to update the terms in the unfixed part and hence the computational complexity of M-step is also reduced to $O(n-|S_f|)$.

$$\begin{aligned}
n_k^t &= \sum_{i=1}^{n}\overline{P}^t_{ik} && (17)\\
&= \sum_{s_i\in S_f}\overline{P}^t_{ik} + \sum_{s_i\notin S_f}\overline{P}^t_{ik} && (18)\\
\mu_k^t &= \frac{\sum_{i=1}^{n}\overline{P}^t_{ik}\mathbf{x}_i}{n_k^t} && (19)\\
&= \frac{\sum_{s_i\in S_f}\overline{P}^t_{ik}\mathbf{x}_i + \sum_{s_i\notin S_f}\overline{P}^t_{ik}\mathbf{x}_i}{n_k^t} && (20)\\
\Sigma_k^t &= \frac{\sum_{i=1}^{n}\overline{P}^t_{ik}\mathbf{x}_i{\mathbf{x}_i}^T}{n_k^t} - \mu_k^t{\mu_k^t}^T && (21)\\
&= \frac{\sum_{s_i\in S_f}\overline{P}^t_{ik}\mathbf{x}_i{\mathbf{x}_i}^T + \sum_{s_i\notin S_f}\overline{P}^t_{ik}\mathbf{x}_i{\mathbf{x}_i}^T}{n_k^t} - \mu_k^t{\mu_k^t}^T && (22)\\
\pi_k^t &= \frac{n_k^t}{n} && (23)
\end{aligned}$$

5 Experimental Evaluation

5.1 Performance Criteria

Let us first take a look at the time complexity of the various EM-style algorithms introduced in this paper. Every pass consists of E-step and M-step. All have the same complexity in M-step, $O(nK)$, except HEM with fixing, whose complexity in later NEM is reduced to $O((n-|S_f|)K)$. As for E-step complexity, EM is $O(nK)$, NEM is $O(mn^2K)$ (m is the number of iterations of E-step in standard NEM), HEM is $O(nK)$ in selective hard EM and $O(n^2K)$ in later NEM. The fastest is EM, closely followed by HEM, and NEM is the worst.

If every site has a true class label, although they are unavailable during training, we can use them to evaluate the final clustering quality. Let $C, Y \in \{1, ..., K\}$ denote the true class label and the cluster label, respectively. Clustering quality is measured with conditional entropy $H(C|Y)$ defined in Eq. (24), which can be

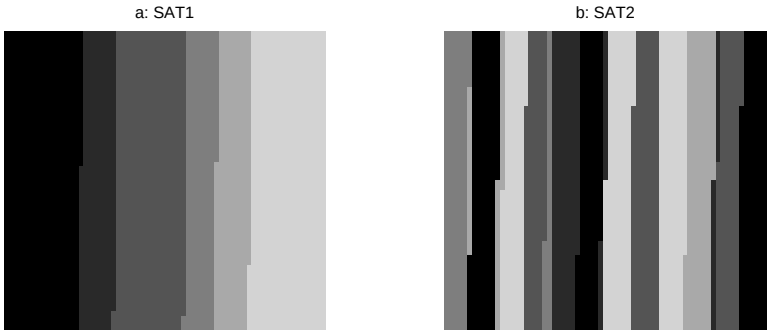

Figure 1: Landcover dataset with site's location synthesized. (a) SAT1's contiguity ratio is 0.9626 and (b) SAT2's contiguity ratio is 0.8858

interpreted as the remaining information in C after knowing Y. Entropy-based criteria have been successfully used in various learning systems, such as node impurity for attribute selection in decision tree [25], and mutual information for discretizing input vector in hybrid speech recognition systems [22]. In the extreme, it equals zero if their distributions are the same. We also use a more intuitive measure, error rate, which is commonly used in classification and can be regarded as a simplified conditional entropy in terms of coding. Using error rate, all data in each cluster that do not belong to the majority class of that cluster are no longer differentiated and we use one bit to encode them. For those belonging to the majority class, we assign zero bit. Therefore, error rate can be written in Eq. (25), where $c(k)$ denotes the majority class label in cluster k and δ denotes the Kronecker function..

$$H(C|Y) = \sum_{k=1}^{K} P(Y=k) \left[\sum_{c=1}^{K} P(C=c|Y=k) \ln \left(\frac{1}{P(C=c|Y=k)} \right) \right] \tag{24}$$

$$E(C|Y) = \sum_{k=1}^{K} P(Y=k) \left[\sum_{c=1}^{K} P(C=c|Y=k) \times (1-\delta(c(k),c)) \right] \tag{25}$$

5.2 Landcover Data

We compare HEM with EM and NEM on a real landcover dataset, Satimage, which is available at the UCI repository [20]. It consists of four multi-spectral values of pixels in 3×3 neighborhoods in a satellite image for an area of agricultural land in Australia. Also provided is the central pixel's class label from a six soil type set { red soil, cotton crop, grey soil, damp grey soil, vegetation stubble, very damp grey soil }. We only use four values for the central pixel. Because the dataset is given in random order and there is no spatial location, we synthesize their spatial coordinates by deleting the first 19 instances from the first class in the training set and allocate the remaining 4416 instances in a 64×69 grid.

4-neighborhood (up, down, left, right) is used in construction of W. The degree of spatial autocorrelation can be measured with Moran's contiguity ratio [4] for continuous attributes. For discrete attributes like soil types, we propose to use Eq. (26), where y denotes the true class label. In the case of regular lattice data like images, it just computes the fraction of edges shared by the pixels from the same class.

$$r = \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} W(i,j) \delta(y_i, y_j)}{\sum_{i=1}^{n} \sum_{j=1}^{n} W(i,j)} \tag{26}$$

To emphasize spatial autocorrelation, we generate two images SAT1 and SAT2 in Fig. 1(a,b) with high contiguity ratio 0.9626 and 0.8858, respectively. In SAT1, all data from the same class are connected within a single block. In SAT2, each class is divided into several blocks. Within the block, data are randomly positioned.

To select β, we test NEM with $\beta = 0.25, 0.5, 1$. The best results are obtained with $\beta = 1$, which needs about 30/10 iterations of E-step for SAT1/SAT2. Table 1 gives the average results of 10 runs by Gaussian

Table 1: Clustering performance on landcover data. $^{+}$SAT1 and *SAT2.

			SAT1			SAT2		
	supervised	EM	NEM	HEM	HEMf	NEM	HEM	HEMf
entropy	0.5121	0.6320	0.5391	0.5176	0.5276	0.5635	0.5530	0.5520
error	0.1508	0.2315	0.2039	0.1919	0.1974	0.2142	0.2057	0.2057
$-U(10^4)$	$^{+}$5.1884 *5.2274	$^{+}$5.1406 *5.1717	5.1029	5.0807	5.0908	5.1416	5.1108	5.1119
$-L(10^4)$	5.8128	5.7711	5.8207	5.7945	5.7974	5.8141	5.7822	5.7823

mixture with random initialization, where HEM/HEMf denotes HEM without/with fixing option. The values are recorded at maximum L for EM, and at maximum U for NEM and HEM. For clarity, we report $-L$ and $-U$ so that all criteria in the tables are to be minimized. For comparison, we also list the results under supervised mode where each component's parameters are estimated with all data from a single class.

We can see that the entropy and error generally decrease as $-U$, rather than $-L$, decreases. Although the lowest $-L$ is achieved by EM, its entropy and error are the worst. This means that for spatial data with high spatial autocorrelation, clustering quality depends not on L, but on U which incorporates the spatial penalty term. As expected, NEM and HEM give better results on SAT1 than on SAT2, for the former's contiguity ratio is higher and hence fits our assumption better.

HEM without fixing slightly beats HEM with fixing on both datasets, probably because (1) we cannot guarantee that all kernel sites in the fixing set receive right classification, and (2) with some fixed sites, NEM cannot perform unconstrained search as it does originally. So the advantage of HEM with fixing in this case seems to be the computational cost it saves, for 48%/37% sites are fixed on the turn to NEM for SAT1/SAT2, which means that in the later NEM part, every pass needs about half computation as its counterpart does in HEM without fixing.

Most trainings converge within 50 passes. For SAT1/SAT2, HEM makes the switch to NEM after about 24/26 passes and slightly outperforms standard NEM in terms of all criteria after convergence. Relatively, the lead is more evident on U than on entropy and error, because of the different form of posterior they use. For many $\overline{P}(y)$, U uses their original soft forms that are different between HEM and NEM. After hardening, however, the binary forms, which are used by entropy and error, become the same. Two typical trainings are depicted in Fig. 2(a-c) for SAT1, and in Fig. 2(d-f) for SAT2. The figures show that NEM initially converges faster than HEM, because NEM directly minimizes $-U$ while HEM minimizes $-F$. However, this faster speed comes with a cost, for NEM needs about 30/10 times computation in every pass for SAT1/SAT2 as HEM does. If fixing option is used in HEM, then after switching, this ratio nearly doubles. After about 30 passes, HEM generally catches up with NEM and converges later to a better or close solution to NEM.

To see if one iteration of E-step of NEM is really enough in HEM, we perform a series of experiments by varying the number of iterations of E-step of NEM. The average results of 10 runs are shown in Table 2. Note that 30/10 is the number of iterations of E-step we used in standard NEM. Although the computational cost has been increased by an order of magnitude, we can see that the improvement is not significant, especially in error rate and U.

5.3 House Price Data

We evaluate HEM on a real house price dataset [10] available at [18]. To cluster the dataset, we use 12 explanatory variables, such as nitric oxides concentration, crime rate, index of accessibility to radial highways, average number of rooms per dwelling. The clustering performance is based on the target variable, median values of owner-occupied homes, which is expected to has a small spread in each cluster. Fig. 3(a) shows the true house values of 506 towns in Boston area. Their histogram is plotted in Fig. 3(b), which we can roughly model with a mixture of two components.

Using a Gaussian mixture of two components, we evaluate $\beta = 0.5, 1, 2$, and finally set it to 1. 20 iterations

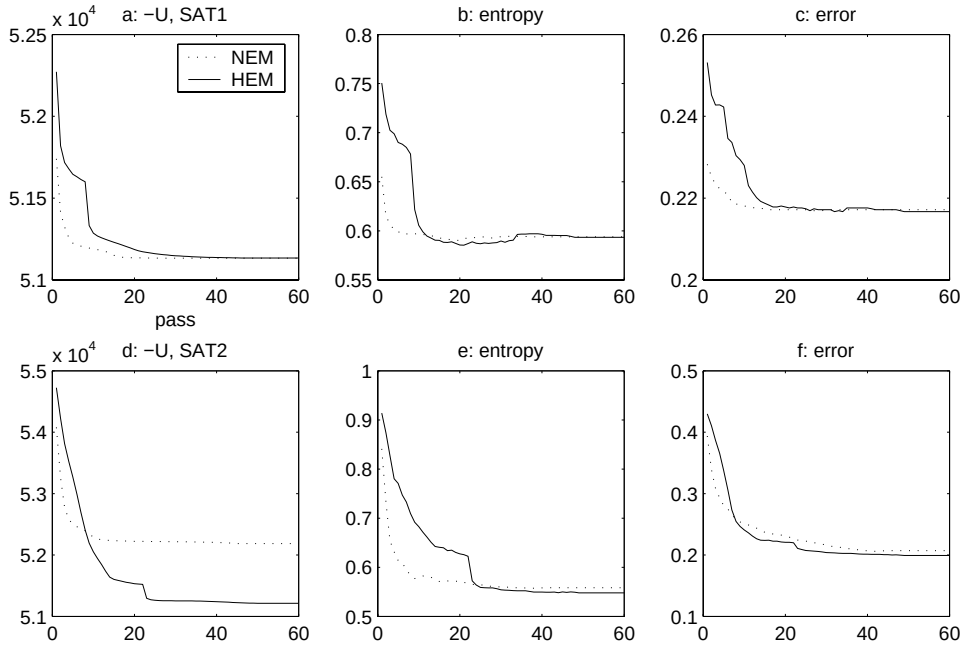

Figure 2: Two samples of training for landcover data. (a-c) for SAT1 and (d-f) for SAT2.

Table 2: Clustering performance on landcover data by HEM with varying number of iterations of E-step.

	SAT1				SAT2		
#E-step	1	10	20	30	1	5	10
entropy	0.5176	0.5095	0.5089	0.5087	0.5530	0.5472	0.5468
error	0.1919	0.1869	0.1868	0.1867	0.2057	0.2032	0.2028
$-U(10^4)$	5.0807	5.0746	5.0730	5.0727	5.1108	5.1091	5.1091
$-L(10^4)$	5.7945	5.7976	5.7990	5.7994	5.7822	5.7830	5.7830

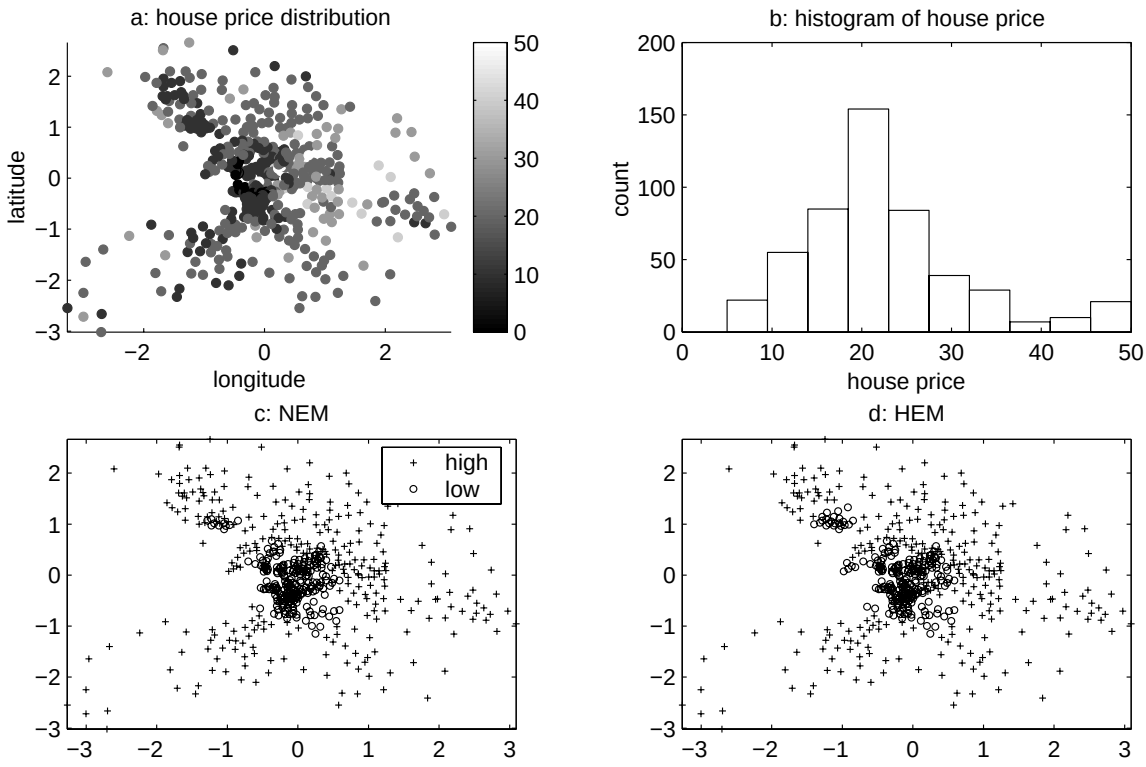

Figure 3: (a) shows house price distribution in 506 towns in Boston area. The corresponding histogram is plotted in (b), which can be roughly modeled with a mixture of two components. Two sample clustering results are shown in (c,d) for NEM and HEM, respectively.

Table 3: Clustering performance on house price data.

	EM	NEM	HEM
$-U(10^4)$	1.2580	1.2675	1.2572
$-L(10^4)$	1.3942	1.4014	1.3946

are needed by E-step of NEM. The average results of 10 runs are given in Table 3. Because the target variable is continuous, we cannot apply Eq. (24, 25) to compute conditional entropy or error rate, so we only report $-U$ and $-L$. One can see that NEM performance is slightly worse than EM in terms of U, but HEM still gives the best result. Two sample clustering results are shown in Fig. 3(c,d) for NEM and HEM, respectively. We can see that HEM yields a clustering with even stronger spatial continuity than that of HEM, which is also confirmed by its average U value. For this data, HEM makes the turn to NEM after about 7 passes. Although 75% sites are fixed in HEM with fixing, it leads to the same result as that without fixing. We also test HEM with different number of iterations of E-step, such as 5,10,15,20. All of them lead to results very close to standard HEM with one iteration of E-step.

5.4 Bacteria Image

Finally, we compare HEM and NEM on an image segmentation problem to extract bacteria from background. In detail, as shown in Fig. 4(a), an extracted bacteria image of 40×40 is to be divided into four regions: dark region of the bacterium itself, bright region immediately surrounding the bacterium, less bright region farther away from the bacterium and grey background. The left and right boundary between the bacterium and its surrounding bright region is really very fuzzy. Due to the conflicting and mixing impact from both sides, the intensity of these border pixels are close to the grey background. Also note that in the right upper corner,

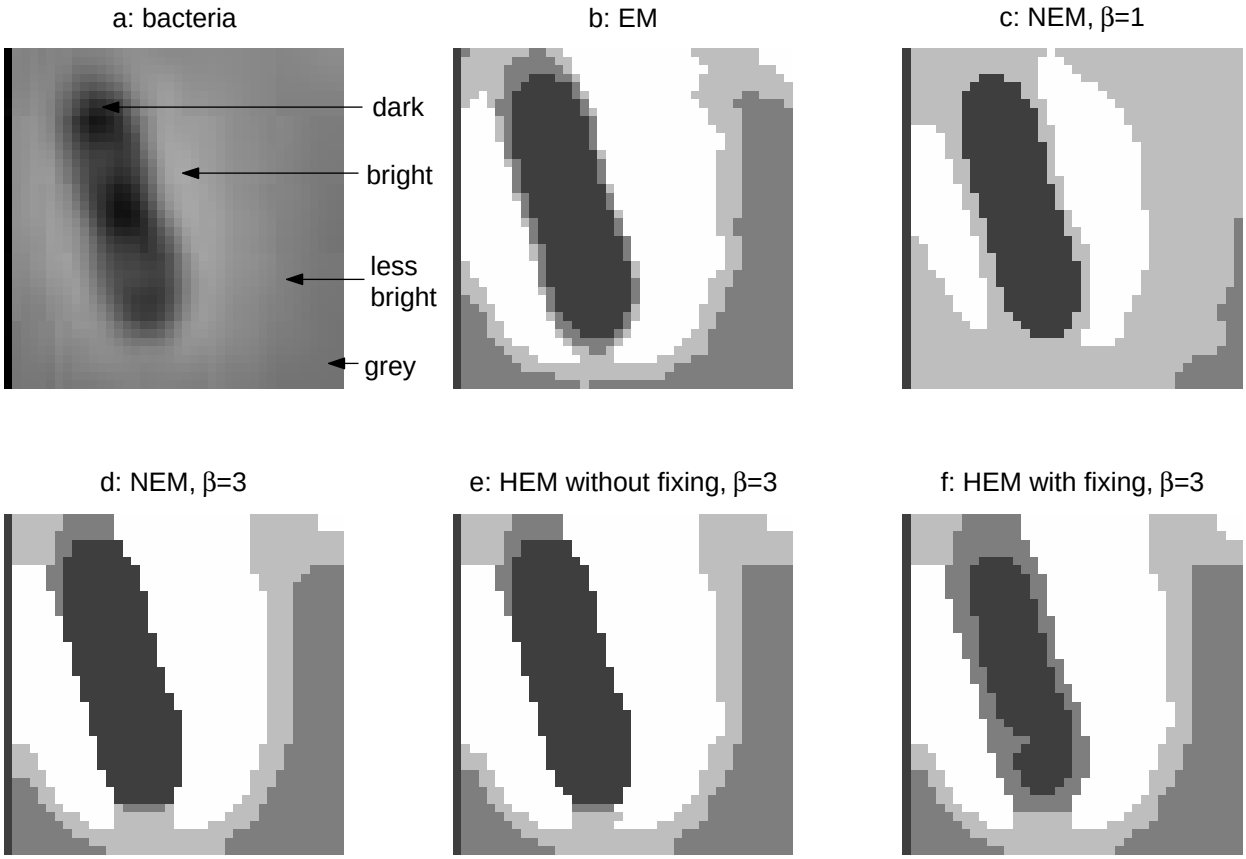

Figure 4: Clustering results for bacteria image. Original image (a) and various clustering results by EM (b), NEM (c-d) and HEM (e-f).

there is a bright area, due to another bacterium in the original image.

Using Gaussian mixture of four components, the best results of 10 runs are illustrated in Fig. 4(b-f). As shown in Fig. 4(b), since EM does not consider spatial information, its output is rather fragmented. In particular, it fails to smooth the bacterium border area, where most pixels are classified as less bright or grey, rather than dark or bright.

For NEM, first we test $\beta = 0.5, 1, 2$. With $\beta = 0.5$, we obtain results similar to EM, which means spatial information has not been emphasized enough. With $\beta = 1, 2$, we obtain results like Fig. 4(c). Although all clusters are connected ones, the bacterium border area is still misclassified as less bright. The reason is that the impact of its neighbors in the dark and bright regions is still very weak and the distribution offered by neighbors is unstable or close to uniform. To make the winners more powerful and hence magnify the neighbors' correct impact, we need a large β. With $\beta = 3$ and 20 iterations of E-step, NEM produces the clustering in Fig. 4(d), where dark and bright regions successfully grow from both side of the border area and finally meet each other by completely occupying the border area.

With $\beta = 3$ and no fixing, HEM generates the clustering in Fig. 4(e), which is very similar to NEM. Once fixing option is employed, however, HEM results in the clustering in Fig. 4(e) where the grey class dominates the bacterium border area, though about 60% pixels are fixed on the turn and thus 60% computation is saved in later NEM. Compared to HEM with fixing, we can see that although those border pixels are misclassified as grey on the turn in HEM without fixing, due to a large β, they are converted to dark or bright in later NEM. Detailed $-U$ and $-L$ are reported in Table 4, which indicates that HEM(without fixing) leads to a much lower $-U$ than HEMf (with fixing) does. It suggest that we should not use fixing option when the mixture model does not fit the data very well or the border area is very fuzzy.

Table 4: Clustering performance comparison for bacteria image.

	EM	NEM	HEM	HEMf
$-L(10^3)$	7.325	7.351	7.353	7.438
$-U(10^3)$	1.238	-0.712	-0.705	-0.471

6 Conclusion

Spatial clustering usually requires consideration of spatial information. In EM style algorithms, this makes likelihood alone inappropriate and a spatial penalty term must be incorporated. Although NEM incorporates a spatial penalty term, it needs more iterations in every E-step. To incorporate spatial information while avoid too much additional computation, we proposed HEM that combines EM and NEM. In HEM, we first perform a selective hard variant of EM till the penalized likelihood stops increasing. Then training is turned to NEM, which runs one iteration of E-step and plays a role of finer tuning. Thus the computational complexity of every pass is similar to EM and much lower than NEM. Experiments show the final clustering performance is comparable to NEM.

There are several research directions for improving HEM. First, as in most EM style algorithms, the final result of HEM depends on initialization. We can try some incremental variants of EM. Second, it is worth trying other penalty terms, such as the derivative of likelihood. The general requirement is that it should embody spatial information without entailing much trouble in optimizing the penalized new criterion. Finally, as in NEM, choosing penalty term coefficient β remains a main difficulty and it is highly desirable if we can automatically determine its optimal value. This value may be chosen independently for each site by automatically weighting its relative importance. All these issues need further research.

References

[1] C. Ambroise and G. Govaert. Convergence of an EM-type algorithm for spatial clustering. *Pattern Recognition Letters*, 19(10):919 – 927, 1998.

[2] S. Basu, M. Bilenko, and R. J. Mooney. A probabilistic framework for semi-supervised clustering. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 59–68, 2004.

[3] G. Celeux and G. Govaert. A classification EM algorithm for clustering and two stochastic versions. *Computational Statistics and Data Analysis*, 14(3):315–332, 1992.

[4] N. A. Cressie. *Statistics for Spatial Data*. John Wiley & Sons, revised edition, 1993.

[5] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of Royal Statistical Society*, B(39):1–38, 1977.

[6] M. Ester, H. P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining*, pages 226–231, 1996.

[7] V. Estivill-Castro and I. Lee. Fast spatial clustering with different metrics and in the presence of obstacles. In *Proceedings of the 9th ACM International Symposium on Advances in Geographic Information Systems*, pages 142 – 147, 2001.

[8] N. Friedman. The Bayesian structural EM algorithm. In *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*, pages 129–138, 1998.

[9] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741, 1984.

[10] O. W. Gilley and R. K. Pace. On the harrison and rubinfeld data. *Journal of Environmental Economics and Management*, 31:403–405, 1996.

[11] D. Guo, D. Peuquet, and M. Gahegan. Opening the black box: Interactive hierarchical clustering for multivariate spatial patterns. In *Proceedings of the 10th ACM International Symposium on Advances in Geographic Information Systems*, pages 131 – 136, 2002.

[12] R. J. Hathaway. Another interpretation of the EM algorithm for mixture distributions. *Statistics and Probability Letters*, 4:53–56, 1986.

[13] A. K. Jain and F. Farrokhnia. Unsupervised texture segmentation using Gabor filters. *Pattern Recognition*, 24(12):1167–1186, 1991.

[14] G. Karypis, E. H. Han, and V. Kumar. CHAMELEON: A hierarchical clustering algorithm using dynamic modeling. *Computer*, 32(8):68–75, 1999.

[15] L. Kaufman and P. J. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley & Sons, 1990.

[16] S. Kullback and R. A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22:79–86, 1951.

[17] P. Legendre. Constrained clustering. In P. Legendre and L. Legendre, editors, *Developments in Numerical Ecology*, pages 289–307, 1987. NATO ASI Series G 14.

[18] J. P. LeSage. *MATLAB Toolbox for Spatial Econometrics*. http://www.spatial-econometrics.com, 1999.

[19] R. Mceliece. *Theory of Information and Coding*. Addison-Wesley, 1977.

[20] P. M. Murphy and D. W. Aha. *UCI Repository of Machine Learning Databases*. Department of Information and Computer Science, University of California at Irvine, 1994. http://www.ics.uci.edu/ ∼ mlearn/MLRepository.html.

[21] R. Neal and G. Hinton. A view of the EM algorithm that justifies incremental, sparse, and other variants. In M. Jordan, editor, *Learning in Graphical Models*, pages 355–368. Kluwer Academic Publishers, 1998.

[22] C. Neukirchen, J. Rottland, D. Willett, and G. Rigoll. A continuous density interpretation of discrete HMM systems and MMI-neural networks. *IEEE Transactions on Speech and Audio Processing*, 9(4):367–377, 2001.

[23] R. Ng and J. Han. CLARANS: A method for clustering objects for spatial data mining. *IEEE Transactions on Knowledge and Data Engineering*, 14(5):1003–1016, 2002.

[24] J. M. Pena, J. A. Lozano, and P. Larranaga. An improved Bayesian structural EM algorithm for learning Bayesian networks for clustering. *Pattern Recognition Letters*, 21(8):779–786, 2000.

[25] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, CA, 1993.

[26] J. P. Rasson and V. Granville. Multivariate discriminant analysis and maximum penalized likelihood density estimation. *Journal of the Royal Statistical Society*, B(57):501–517, 1995.

[27] A. K. H. Tung, J. Hou, and J. Han. Spatial clustering in the presence of obstacles. In *Proceedings of 17th International Conference on Data Engineering*, pages 359–367, 2001.