European Journal of Operational Research 271 (2018) 594–605

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Decision Support

Model-based capacitated clustering with posterior regularization

Feng Mai [a,*], Michael J. Fry [b], Jeffrey W. Ohlmann [c]

[a] School of Business, Stevens Institute of Technology, Hoboken, NJ 07030, USA
[b] Department of Operations, Business Analytics & Information Systems, University of Cincinnati, Cincinnati, OH 45221, USA
[c] Tippie College of Business, University of Iowa, Iowa City, IA 52242, USA

ARTICLE INFO

Article history:
Received 28 December 2016
Accepted 29 April 2018
Available online 17 May 2018

Keywords:
Heuristics
Gaussian mixture models
Expectation-maximization algorithm
Posterior regularization
Capacitated *p*-median problem

ABSTRACT

We propose a heuristic approach to address the general class of optimization problems involving the capacitated clustering of observations consisting of variable values that are realizations from respective probability distributions. Based on the expectation-maximization algorithm, our approach unifies Gaussian mixture modeling for clustering analysis and cluster capacity constraints using a posterior regularization framework. To test our algorithm, we consider the capacitated *p*-median problem in which the observations consist of geographic locations of customers and the corresponding demand of these customers. Our heuristic has superior performance compared to classic geometrical clustering heuristics, with robust performance over a collection of instance types.

1. Introduction

In a capacitated clustering problem (CCP), a set of n observations must be partitioned into p disjoint clusters so that the total dissimilarity within each cluster is minimized while each cluster's capacity (measured in terms of some characteristic of the observations) is obeyed. The CCP arises both as the primary task and as a sub-problem in many real-world applications. For example, when designing a distribution network, a set of customers must be supplied goods from warehouses subject to the capacity of warehouses (Salema, Barbosa-Povoa, & Novais, 2007). In the topological design of computer communication networks, the network nodes need to be divided into groups, and a concentrator location must be selected for each group so that all the nodes in a group can be assigned to the same concentrator without violating capacity constraints (Pirkul, 1987). Recently, the CCP has also been applied to genetics and population biology to solve the sibling reconstruction problem (Chou, Chaovalitwongse, Berger-Wolf, DasGupta, & Ashley, 2012). In addition, many important applications such as market segmentation, vehicle routing, and location-routing problems involve solving the capacitated clustering as a sub-problem.

In this paper, we propose a CCP heuristic based on a Gaussian mixture modeling approach that incorporates the capacity constraints. Specifically, we modify the well-known expectation-maximization (EM) algorithm that iteratively determines maximum likelihood estimates for the parameters of the latent variable mixture model (Dempster, Laird, & Rubin, 1977). While the EM algorithm has been widely used in statistics, computer science, and marketing research, it has limited usage in solving optimization problems arising in operations research, partially due to the difficulty of incorporating external constraints. Using the posterior regularization (PR) framework of Ganchev, Graça, Gillenwater, and Taskar (2010), we account for the cluster capacity constraints. PR allows prior knowledge to be introduced into models that are traditionally considered as unsupervised learning. Ganchev et al. (2010) show that prior knowledge can be encoded as constraints on posterior probabilities and can be used to guide the outputs on various tasks in natural language processing such as part-of-speech tagging, word alignment, and dependency grammar parsing.

As a testing arena for our approach, we consider the capacitated *p*-median problem (CPMP), a classical location-allocation problem and one of the most-studied variations of the CCP. In the CPMP, the observations consist of geographic locations of customers and the corresponding demand of these customers; each cluster is constrained in the total amount of demand it may be allocated.

This work makes the following contributions. To our knowledge, this is the first CCP study to apply a Gaussian mixture model via an EM-based algorithm with posterior regularization to address the capacity constraints. We demonstrate the promise of this approach in our computational testing as our heuristic outperforms the geometrical-clustering methods proposed by Mulvey and Beck (1984) and Ahmadi and Osman (2004). Further, we investigate the robustness of our approach with respect to the spatial distribution of the observations and consider a case study based on real-world data to demonstrate the applicability of our approach. Moreover,

* Corresponding author.
E-mail address: feng.mai@stevens.edu (F. Mai).

https://doi.org/10.1016/j.ejor.2018.04.048
0377-2217/

we formulate stochastic variations of the CCP and demonstrate that our approach adapts to these variants. As such, our approach is able to consistently achieve high-quality feasible solutions to the CCP without relying on any elaborate search procedures. Due to its ease of implementation and ability to accommodate cluster-level constraints, this procedure can potentially be used an initial construction phase of a local search procedure tailored to a particular application of the CCP. From a broader perspective, we believe that our work provides a novel integration of the fields of statistical machine learning and operations research that could spur future work in this area.

We begin the rest of the paper by first briefly reviewing related literature in Section 2. Then, in Section 3, we formally introduce the CPMP as a mixed-integer linear programming (MILP) model. In Section 4, we review the Gaussian mixture model and the EM algorithm used to maximize the model likelihood. Then we describe how we adapt the PR framework to solve the CPMP in Section 5. In Section 6, we present the computational results on deterministic and stochastic versions of CPMP. In Section 7, we present a case study based on a real world dataset that provides an example of a situation well-suited to our approach. We present our conclusions in Section 8.

2. Related literature

The literature contains both exact and heuristic approaches for the CCP. Exact approaches rely upon integer programming formulations of the CCP. Pirkul (1987) propose exact algorithms using branch-and-bound with Lagrangean relaxation on the partitioning constraints. Baldacci, Hadjiconstantinou, Maniezzo, and Mingozzi (2002) present a set partitioning approach. Lorena and Senne (2004) present column generation approaches and Ceselli (2003) and Ceselli and Righini (2005) develop a branch-and-price algorithm. Boccia, Sforza, Sterle, and Vasilyev (2008) present a cutting-plane algorithm based on a formulation strengthened by Fenchel cuts. However, because the CCP problem is NP-hard (Mulvey & Beck, 1984), exact approaches often cannot guarantee an optimal solution within a practical amount of time for realistically-sized problems.

As an alternative to exact approaches, there have been many studies developing a variety of heuristic methods for the CCP problem. Common meta-heuristics applied to the CCP include genetic algorithms (Alp, Erkut, & Drezner, 2003; Correa, Steiner, Freitas, & Carnieri, 2004; Jánošíková, Herda, & Haviar, 2017; Landa-Torres et al., 2012), simulated annealing (Osman & Christofides, 1994), variable neighborhood search (Fleszar & Hindi, 2008), tabu search (Bozkaya, Erkut, & Laporte, 2003; Osman & Christofides, 1994), bionomic algorithm (Maniezzo, Mingozzi, & Baldacci, 1998), GRASP (Deng & Bard, 2011), and scatter search (Díaz & Fernandez, 2006; Scheuerer & Wendolsky, 2006).

Local-search-based meta-heuristics begin by constructing an initial solution and iteratively improving it. Clustering algorithms, such as *k*-means or hierarchical clustering, are natural candidates for the construction stage of a heuristic, but the key challenge is incorporating problem-specific knowledge such as capacity constraints. A naive algorithm could avoid merging two clusters or stop adding observations to a cluster if such an operation would violate the capacity constraints. The problem with this approach is that points naturally close to each other may be prevented from being grouped together because of the capacity constraint, while points that are far away may be forced into the same cluster (Barreto, Ferreira, Paixao, & Santos, 2007).

Successful heuristics for this problem should take a holistic perspective. To help reduce the need for extensive repairs by a local search phase, Mulvey and Beck (1984) adapt *k*-means clustering to accommodate cluster capacities by defining an assignment *regret* quantity for each observation based on the notion of prioritizing the cluster assignment of observations whose assignment alternatives are the least desirable. Along this line, Ahmadi and Osman (2004) propose an improved definition of regret and introduced the notion of density to the construction heuristic. Instead of initiating the heuristics randomly, the cluster centers will be set on the points that have high-density values.

Instead of incorporating the capacity constraint in an ad hoc manner, we rely upon a principled and robust model-based clustering to conduct capacitated clustering. The flexibility comes from the fact that popular clustering heuristics are approximate methods for a certain model. For example, *k*-means and Ward's method maximize the Gaussian likelihood when the covariance matrix is the same multiple of the identity matrix across mixtures (Fraley & Raftery, 2002). As another example, Dasgupta and Raftery (1998) show that model-based clustering can be extended to detect irregular shapes such as the parallel rectangles and arrow shapes. Although not a focus in this paper, one can imagine that these structures can also be meaningful when taking capacity into consideration.

3. The capacitated *p*-median problem

We describe the CPMP with the following integer linear optimization model:

$$\min \sum_{i\in\mathcal{I}} \sum_{j\in\mathcal{J}} c_{ij} w_{ij} \tag{1}$$

$$\text{s.t.} \sum_{j\in\mathcal{J}} w_{ij} = 1 \qquad \forall i \in \mathcal{I} \tag{2}$$

$$\sum_{i} d_i w_{ij} \le C y_j \qquad \forall j \in \mathcal{J} \tag{3}$$

$$\sum_{j\in\mathcal{J}} y_j = p \tag{4}$$

$$w_{ij} \in \{0,1\} \qquad \forall i \in \mathcal{I}, \forall j \in \mathcal{J} \tag{5}$$

$$y_j \in \{0,1\} \qquad \forall j \in \mathcal{J} \tag{6}$$

where

- $i = 1 \ldots n$ is the index of points to allocate and also of possible medians, where k medians will be located;
- $j = 1 \ldots n$ is the index of all possible cluster centers or medians;
- d_i is the demand of each point i and C is the capacity of each possible cluster;
- c_{ij} is the distance from point i to median j ;
- y_j are binary variables, with $y_j = 1$ if point y is selected to be a cluster median;
- w_{ij} are binary variables, with $w_{ij} = 1$ if point i is assigned to median j and $w_{ij} = 0$ otherwise;

The objective of the CPMP in (1) is to minimize the sum of distance from points to the cluster medians. Constraint (2) ensures that all points are allocated to exactly one cluster median. Constraint (3) imposes the constraints on cluster capacities, and constraint (4) sets the number of medians to k. Constraints (5) and (6) enforce the binary nature of the decision variables.

4. Model-based clustering with EM algorithm

In this section, we describe Gaussian mixture model-based clustering and our use of the expectation maximization algorithm with posterior regularization (EMPR) to determine the maximum likelihood estimates of the cluster parameters subject to posterior constraints (Ganchev et al., 2010). We then apply this algorithm to incorporate capacity constraints into the model-based clustering. In the EMPR framework, the constraints serve as indirect supervision to the probabilistic learning framework. The posterior probability distributions of latent variables are guided by the constraints toward desired behavior. In the CCP, PR can sway the latent variables (assignment of points) to conform to the capacity constraints while trying to maximize the model likelihood.

4.1. Capacitated Gaussian mixture model via the EMPR algorithm

Let $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(n)}$ be a set of observations in $\mathbb{R}^d$ that arise from a mixture of k groups. The probability that a randomly selected observation comes from the jth mixture is π_j, where $0 \le \pi_j \le 1$ and $\sum_{j=1}^{k} \pi_j = 1$. The mixture density is

$$f(\boldsymbol{x}) = \sum_{j=1}^{k} \pi_j \Phi(\boldsymbol{x}|\mu_j, \Sigma_j), \tag{7}$$

where $\Phi(\boldsymbol{x}|\mu_j, \Sigma_j)$ denotes a Gaussian density with mean μ and variance matrix Σ. We denote the probability that observation i arises from the jth mixture as $p_j^{(i)}$.

The EM algorithm (Dempster et al., 1977) is an iterative method to compute the maximum likelihood estimation for probability models with missing or latent data. In the context of mixture models, the observed data are the $\boldsymbol{x}^{(i)}$, and the latent part of the data is $z_j^{(i)}$ with its value equal to 1 if $\boldsymbol{x}^{(i)}$ belongs to mixture j, and 0 otherwise. The log-likelihood of the complete model is then

$$l(\pi, \mu, \Sigma, z) = \sum_{i=1}^{n} \log \left(\sum_{j=1}^{k} \pi_j \Phi(\boldsymbol{x}|\mu_j, \Sigma_j) \right). \tag{8}$$

To enforce constraints on the clusters, posterior regularization defines the set of valid posterior distributions with respect to expectation of constraints. That is, suppose Q is a set of valid distributions, $q(\boldsymbol{Z})$ is a distribution of latent variables $\boldsymbol{Z}$, and $\boldsymbol{\phi(X, Z)}$ is a function of observed variables $\boldsymbol{X}$ and latent variables $\boldsymbol{Z}$. The desired set of posterior distributions is

$$Q = \{q(\boldsymbol{Z}) : \boldsymbol{E}_q[\phi(\boldsymbol{X}, \boldsymbol{Z})] \le \boldsymbol{b}\}. \tag{9}$$

The objective is to determine parameters (μ, Σ) that maximize (8) subject to (9). To maximize the likelihood of a Gaussian mixture model, the EM algorithm alternates between the E-step and the M-step while raising the lower bound of the model likelihood in each iteration. When the constraint (9) is not present, the E-step computes the conditional expectation of the latent variables given the current value of the parameter estimates:

$$\begin{aligned} p_j^{(i)} &= E(z_j^{(i)}|\boldsymbol{x}^{(i)}; \pi, \mu, \Sigma) = p(z^{(i)} = j|\boldsymbol{x}^{(i)}; \pi, \mu, \Sigma) \\ &= \frac{p(\boldsymbol{x}^{(i)}|z^{(i)} = j; \mu, \Sigma) p(z^{(i)} = j; \pi)}{\sum_{l=1}^{k} p(\boldsymbol{x}^{(i)}|z^{(i)} = l; \mu, \Sigma) p(z^{(i)} = l; \pi)}. \end{aligned} \tag{10}$$

We can find $p(\boldsymbol{x}^{(i)}|z^{(i)} = j; \mu, \Sigma)$ by evaluating a multivariate Gaussian density with mean $\boldsymbol{\mu}_j$ and covariance $\boldsymbol{\Sigma}_j$ at point $\boldsymbol{x}^{(i)}$. The current estimate of mixture probability π_j gives us $p(z^{(i)} = j; \pi)$.

When the constraint is present, in order to find the $q(\boldsymbol{Z}) \in Q$ such that (8) is maximized, Ganchev et al. (2010) show that at each of E-step of the EMPR algorithm, we solve the following optimization problem:

$$\min_q \boldsymbol{KL}(q||p_\theta(\boldsymbol{z}|\boldsymbol{x})) \qquad \text{s.t. } \boldsymbol{E}_q[\phi(\boldsymbol{X}, \boldsymbol{Z})] \le \boldsymbol{b}, \tag{11}$$

where $\boldsymbol{KL}$ is the Kullback–Leibler divergence, which is a measure of the difference between two distributions. It is defined as $\boldsymbol{KL}(q||p) = \sum_q q \log(q/p)$.

The above problem can be solved more efficiently in its dual form

$$\max_{\boldsymbol{\lambda} \ge 0} -\boldsymbol{b} \cdot \boldsymbol{\lambda} - \log Z(\boldsymbol{\lambda}), \tag{12}$$

where $Z(\boldsymbol{\lambda}) = \sum_{\boldsymbol{Z}} p_\theta(\boldsymbol{Z}|\boldsymbol{X}) \exp[-\boldsymbol{\lambda}^* \cdot \phi(\boldsymbol{X}, \boldsymbol{Z})]$, and the solution to the primal is given by

$$q^*(\boldsymbol{Z}) = \frac{p_\theta(\boldsymbol{Z}|\boldsymbol{X}) \exp[-\boldsymbol{\lambda}^* \cdot \phi(\boldsymbol{X}, \boldsymbol{Z})]}{Z(\lambda^*)}. \tag{13}$$

Notice that in the E-step of the standard EM algorithm (without PR), we are solving $\min_q \boldsymbol{KL}(q||p_\theta(\boldsymbol{z}|\boldsymbol{x}))$ (see Neal & Hinton, 1998 for the proof). The optimal solution for $q(\boldsymbol{Z})$ is $p_\theta(\boldsymbol{Z}|\boldsymbol{X})$, which is the posterior probability of latent variables given the current parameters and the observed variables. In the EMPR framework, we instead restrict q to the set Q defined in (9). The restriction trades off a smaller maximum lower bound of likelihood for desired posteriors. Therefore, a simple way of interpreting the EMPR framework is to add a penalty term $\exp[-\boldsymbol{\lambda}^* \cdot \phi(\boldsymbol{X}, \boldsymbol{Z})]$ to tune down the posterior probability in the E-step of the EM algorithm for violations of constraints.

The M-step in the EMPR algorithm is the same as in the standard EM algorithm. That is, the M-step updates the model parameters:

$$\pi_j \leftarrow \frac{1}{n} \sum_{i=1}^{n} p_j^{(i)}, \tag{14}$$

$$\boldsymbol{\mu}_j \leftarrow \sum_{i=1}^{n} p_j^{(i)} \boldsymbol{x}^{(i)} / \sum_{i=1}^{n} p_j^{(i)}, \tag{15}$$

$$\boldsymbol{\Sigma}_j \leftarrow \sum_{i=1}^{n} p_j^{(i)} (\boldsymbol{x}^{(i)} - \boldsymbol{\mu}_j)/(\boldsymbol{x}^{(i)} - \boldsymbol{\mu}_j)^T / \sum_{i=1}^{n} p_j^{(i)}. \tag{16}$$

4.2. Parsimonious models

A practical issue with multivariate Gaussian models is that the number of parameters grows rapidly with the number of clusters. For example, in the facility-location problem with two dimensions, a k-cluster full normal mixture model will have (number of mean parameters) + (number of covariance parameters) + (number of mixture proportion parameters) $= 2k + 3k + (k-1) = 6k - 1$ parameters. Too many parameters, when compared to the number of data points, can result in issues such as degradation of performance and under-specified models (Raftery & Dean, 2006). In particular, the estimation for a full covariance matrix will be singular or near singular.

Banfield and Raftery (1993) show that the covariance matrix of a multi-variate Gaussian distribution can be parameterized in terms of its eigenvalue decomposition in the form

$$\Sigma_k = \lambda_k \boldsymbol{D}_k \boldsymbol{A}_k \boldsymbol{D}_k^T. \tag{17}$$

This approach was later generalized by Celeux and Govaert (1995) into Gaussian parsimonious clustering models. A parsimonious model imposes various restrictions on covariance matrices of the distribution by fixing λ_k, $\boldsymbol{D}_k$, or $\boldsymbol{A}_k$. Imposing these restrictions on the covariance matrices restricts the volume, orientation, or shape of each cluster. We describe two parsimonious models EII and VII (using the notations in Fraley, Raftery et al., 2007). These two models set $\boldsymbol{D}_k$ and $\boldsymbol{A}_k$ to I and restrict the shape of the clusters to be spherical.

1. *EII:* $\Sigma_1 = \cdots = \Sigma_k = \sigma^2 I$. In Model EII, σ^2 is the only unknown covariance parameter.

2. *VII:* $\Sigma_1 = \cdots = \Sigma_k = \sigma_1^2 I, \ldots, \sigma_k^2 I$. In Model VII, the number of covariance parameters is equal to the number of clusters.

In Model EII each cluster has an n-spherical shape and equal volume. In Model VII the clusters are still n-spherical, but the volumes are allowed to vary. When using a more restricted covariance structure, the inference for covariance parameters during the M-step becomes simpler. Instead of iterative optimization procedures which are required for the full model, many parsimonious models have closed-form solutions for Σ in the M-step. In Model EII, the updated σ^2 is

$$\sigma^2 = \frac{\text{tr}(W)}{nd}, \tag{18}$$

where $W = \sum_{j=1}^{k} \sum_{i=1}^{n} p_j^{(i)} (\boldsymbol{x}^{(i)} - \bar{\boldsymbol{x}}_j)(\boldsymbol{x}^{(i)} - \bar{\boldsymbol{x}}_j)'$.

In the M-step of Model VII, σ_j^2 can be calculated as

$$\sigma_j^2 = \frac{\text{tr}(W_j)}{n_j d}, \tag{19}$$

where $W_j = \sum_{i=1}^{n} p_j^{(i)} (\boldsymbol{x}^{(i)} - \bar{\boldsymbol{x}}_j)(\boldsymbol{x}^{(i)} - \bar{\boldsymbol{x}}_j)'$ and $n_j = \sum_{i=1}^{n} p_j^{(i)}$.

Neither the full model nor parsimonious models other than EII and VII are able to give satisfactory performance in the CPMP problems we tested. This is mostly because typical CPMP instances use Euclidean distance as the cost measure, which naturally leads to spherical-shaped clusters. When the costs are assymmetrical across the dimensions, other more complicated variance structures can be considered. For example, if it is more expensive to travel along one axis than another, we may consider model EEI (Celeux & Govaert, 1995), which removes the restriction on $\boldsymbol{A}$ and allows for ellipsoidal-shaped clusters.

5. Model-based heuristic algorithm based on posterior regularization

5.1. Posterior regularization framework for capacitated clustering

For a model-based CCP, we need to write the capacity constraint in terms of the expectation of $\boldsymbol{q}$ as in Eq. (11). If we let $z_j^{(i)} = 1$ represent the event that point i is assigned to cluster j, the capacity constraint we want to include when forming clusters is

$$\sum_i z_j^{(i)} d_i \le C, \quad \forall j. \tag{20}$$

Constraint (20) states that the total capacity in each cluster should not exceed C. Expressing the above constraint as posterior constraints in an EM-algorithm, we have

$$E_q\left[\sum_i z_j^{(i)} d_i\right] \le C, \quad \forall j. \tag{21}$$

Computationally, the optimization problem is solved by maximizing the dual at each E-step:

$$\max_{\lambda_1 \cdots \lambda_k \ge 0} -\sum_{j=1}^{k} C\lambda_j - \log\left(\sum_{j=1}^{k}\left(\prod_{i=1}^{n} p_j^{(i)} \exp\left(-\sum_{j=1}^{k} \lambda_j \sum_{i=1}^{n} d_i p_j^{(i)}\right)\right)\right). \tag{22}$$

The maximization problem can be easily solved using standard nonlinear optimization methods such as the Broyden–Fletcher–Goldfarb–Shanno (BFGS) method with barrier. The primal solution is given by

$$q_j^{(i)} = p_j^{(i)} exp\left(-\lambda_j \sum_{i=1}^{n} d_i p_j^{(i)}\right)/Z, \tag{23}$$

where $Z = \sum_{j=1}^{k} q_j^{(i)}$, which is a normalization factor to ensure that q is a valid probability distribution over each point i.

5.2. An example

To illustrate the effect of posterior regularization, we simulate three 20-point groups; for each group of points, their (x, y) coordinates follow a bivariate normal distribution with $\sigma = 1.8$. The points in the first group centered at (7.5, 10) and the second group centered at (10, 5) have demand of 0.5; the points in the third group centered at (5, 5) have demand of 2.

The left plot in Fig. 1 shows the clustering result of the standard EM algorithm. As expected, the three groups with about an equal number of points are formed according to their (x, y) coordinates. If we set the capacity constraint of each cluster at 25, then the group at the bottom-left corner will violate this capacity constraint. The plot on the right of Fig. 1 shows the capacitated clustering result using posterior regularization. We are able to attain the clusters that respect the capacity constraints while having points in a group naturally close to each other.

Given the adaptation of PR Framework described in Section 5.1, we need several adjustments for practical considerations, which we discuss in the next subsections.

5.3. Penalizing posterior distribution

First of all, the number of dual decision variables in (22), λ_j, equals the number of desired clusters k. Solving a problem with k clusters will require solving a non-linear optimization problem with k decision variables at each iteration of EM. Second, we notice that during the initial few iterations of EM, when there are relatively large capacity violations in some clusters, the optimal λ of the dual problems will be very large as well. Because EM converges to a local maxima of log-likelihood, this may cause many clusters to be empty in the final solution. In other words, adding capacity constraints can result in increasing number of undesired local optima.

To mitigate these two issues, we propose that instead of actually solving the dual problem (22) (or more generally (12)) in the E-step using BFGS, we simply check if (21) is satisfied for the current posterior probability matrix $P(Z|X)$. More specifically, given the parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ obtained from the last M-step, we calculate the posterior probability $p_j^{(i)}$ as in the E-step of regular EM (without PR). Then for each column j of the probability matrix we calculate $\sum_i d_i p_j^{(i)}$, and if the quantity is greater than the capacity constraint C, we apply a penalty to the column by multiplying it by a penalty coefficient r, where $0 < r < 1$, that is

$$p_j^{(i)} \leftarrow r p_j^{(i)} \quad \text{if} \quad \sum_i d_i p_j^{(i)} > C, \quad \forall j. \tag{24}$$

Lastly, to make sure that each row of the probability matrix is a valid marginal distribution, we normalize them so that the sum of each row is 1.

While this adjustment results in much faster computation and more stable results, it nullifies the guaranteed likelihood increase of the EM algorithm with PR at each iteration.[1] Fortunately, we believe this is less of an issue practically. As shown in Fig. 2, the log-likelihood of the model still shows consistently increasing trends across the test instances. Our empirical evaluation shows that the final results are very sensible despite the non-guaranteed likelihood convergence. After a number of iterations, the log-likelihood will vary within a small region. This indicates that eventually the EM algorithm will oscillate between several promising solutions when trying to balance between satisfying the constraints and further increasing the model likelihood. Therefore, we implement a

[1] Because a formal EM algorithm always converges to a local optima or a saddle point, we caution readers that our method is more properly defined as a pseudo-EM algorithm and there is no guarantee of such convergence.

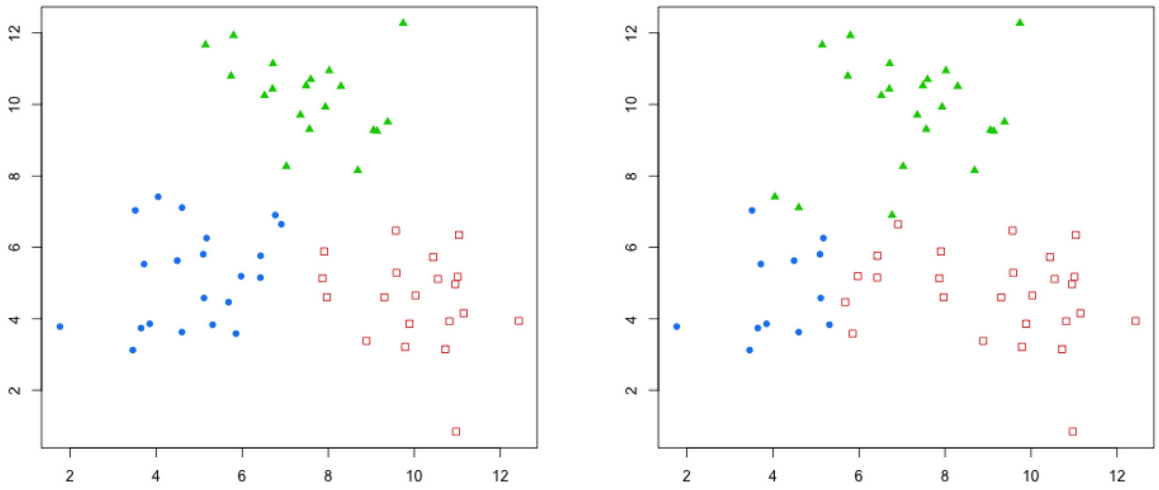

Fig. 1. An illustrative example of capacitated clustering based on PR framework. Points in the bottom-left corner, centered at (5, 5), each have demand of 2. The rest of the points each have demand of 0.5. The clustering results from the standard EM (left) would violate the capacity constraint of 25. EM with PR (right) produces clusters that satisfy the constraint.

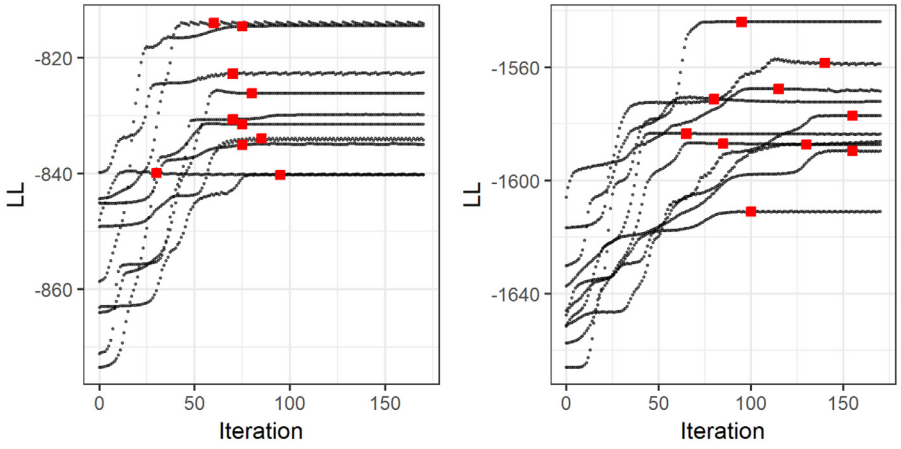

Fig. 2. Log-likelihood in EMPR heuristic. The figure shows the convergence of log-likelihood when using EMPR on the test instances (Osman & Christofides, 1994). Instances $p1$ to $p10$ are on the left, and instances $p11$ to $p20$ are on the right. The squares mark the iterations when the convergence check is passed.

simple convergence check by calculating the log-likelihood after every 5 iterations, and we check if the standard deviation of the last 10 log-likelihoods is smaller than a certain threshold, ϵ. We set $\epsilon = 1$ for our experiments.

5.4. The initialization of EM

Researchers have proposed different initialization strategies for CPMP heuristics. For example, Mulvey and Beck (1984) initialize using random nodes as centers. Osman and Christofides (1994) propose a step to find a set of initial centers that are spread out. Ahmadi and Osman (2004) and Osman and Ahmadi (2007) use a density-based approach to locate the most promising centers.

In our probabilistic model, there is no set of initial centers *per se*. Instead, we can either specify the initial conditional probability matrix (the probabilities of nodes belonging to clusters) or the initial mean vectors and covariance matrices in the context of Gaussian mixture models. We adopt a simple random initialization where the mixing proportions are generated from a symmetric Dirichlet distribution. Besides simplicity of implementation, there are several reasons for us to choose this strategy: (1) it is considered as the standard, and most frequently used initialization strategy, for mixture models (Karlis & Xekalaki, 2003); (2) a comprehensive numerical study by Biernacki, Celeux, and Govaert (2003) demonstrates that under low dimension, other more sophisticated strategies offer no significant improvements; and (3) visual inspection of final solutions produced by our heuristic shows that the quality of solutions depends more on the assignment of nodes, as the clusters are reasonably spread out.

5.5. Assignments of nodes and cluster medians

After the iterations between the regularized E-step and the M-step have converged, we have a posterior distribution of the latent variable $p_\theta(\boldsymbol{Z}|\boldsymbol{X})$. We use the posterior distribution as a guide to assign nodes to clusters.

We investigate several methods to assign the nodes to the clusters with the posterior probability matrix $p_\theta(\boldsymbol{Z}|\boldsymbol{X})$. In the CPMP literature, orders of node assignment include: increasing or decreasing order of demand, increasing order of distances from nodes to medians, etc. Using the decreasing order of regret values is a popular choice; however the definitions of regret differ based on the implementation.

Let $O = o_1, \ldots, o_k$ be the set of medians. For every node x_i, Mulvey and Beck (1984) define regret, $R(x_i)$, as the distance between the closest median, o_{i1}, and the second closest median, o_{i2}:

$$R(x_i) = d(o_{i1}, o_{i2}), \tag{25}$$

Ahmadi and Osman (2004) define the regret as the savings of assigning node i to the closest median compared with assigning it to the second closest median:

$$R(x_i) = d(a_i, o_{i1}) - d(a_i, o_{i2}). \tag{26}$$

We define our regret function as the difference between the posterior probabilities of the two clusters with the highest probabilities of generating x_i:

$$R(x_i) = p_\theta(z_1|x_i) - p_\theta(z_2|x_i). \tag{27}$$

Our regret function based on the posterior probabilities not only considers the location of medians (as in Mulvey & Beck, 1984), and the relative location between nodes and medians (as in Ahmadi & Osman, 2004), but also the entire probabilistic model. In other words, in addition to the geographic locations, the capacity constraint of all clusters is also implicitly considered. This allows the heuristic to take a holistic view of the problem when assigning the nodes.

After assigning nodes to clusters, we determine the cluster medians by simply choosing the node within each cluster that minimizes the within-cluster assignment cost.

5.6. Local search strategies

Once we determine the cluster medians and the associated nodes, we can use local search to improve the solution. Our algorithm explores two different local search neighborhoods – *shift* and *swap*. The shift neighborhood contains the solutions generated from shifting one point assigned to one median to another median. The swap neighborhood contains all pairwise interchanges of non-median nodes between clusters. For a given solution, if a certain shift or swap operation will improve the current solution, and at the same time not violate capacity constraints, the operation can be made. We implement these moves within a best-improvement search strategy; we evaluate all possible moves from a current solution and execute the move resulting in the best improvement (if an improving move exists).

Algorithm 1 formally describes the heuristic based on the EMPR framework that can be used to solve a CPMP.

6. Computational results

6.1. Analyses with test instances

We test our algorithm's performance using the 20 CPMP instances coded as $p1$ to $p20$ from Osman and Christofides (1994). The coordinates of the points are randomly generated from a uniform distribution [1, 100]. The demand values of each point are also generated from a uniform distribution [1, 20]. The first 10 instances have $n = 50$ and $k = 5$, and the other 10 instances have $n = 100$ and $k = 10$. We use the optimality gap of a heuristic algorithm to measure the performance of the algorithms. The optimality gap of a heuristic algorithm is defined as 100 × (heuristic objective value - optimal objective value)/optimal objective value. We code our heuristics in Java and perform all tests on an Intel i5-3570 processor under the Microsoft Windows 10 operating system.

We first study the effects of the posterior penalty parameter r and the choice of parsimonious models on the performance. Because the model choice and posterior penalty is most influential on the solution construction stage, we report the average optimality gap from the solutions without conducting local search. We obtain the optimal values by solving the integer programming model with the Gurobi MILP solver.

Fig. 3 shows that Model VII offers higher quality solutions than Model EII for the test instances. This is not surprising, because Model VII explicitly parameterizes cluster size to be different, and

Algorithm 1 Mixture model clustering via EMPR for the capacitated p-median problem.

Input: coordinates of points $\boldsymbol{x}^{(i)}, (i = 1, \ldots, n)$ with demand d_i, number of clusters k, cluster capacity C
Output: a set of k cluster medians; assignments from n points to medians.
Parameters: penalization constant r

Step 1. Initialization of EM
Initialize a $n \times k$ matrix P with entries denoted as $p_j^{(i)}$.
Draw each row from a symmetric Dirichlet distribution ($\alpha = 1$).
Step 2. EM Iterations
while not convergence **do**
  for all cluster j **do** ▷ Regular M-Step
    Update mixture parameters:
    $\pi_j \leftarrow \frac{1}{n}\sum_{i=1}^n p_j^{(i)}$,
    $\boldsymbol{\mu}_j \leftarrow \sum_{i=1}^n p_j^{(i)}\boldsymbol{x}^{(i)} / \sum_{i=1}^n p_j^{(i)}$,
    Update $\boldsymbol{\Sigma}_j$ according to Eq. (18) or (19).
  end for
  for all point i **do** ▷ Regular E-Step
    Update conditional probabilities:
    Compute $p_j^{(i)} \leftarrow p(z^{(i)} = j|\boldsymbol{x}^{(i)}; \phi, \boldsymbol{\mu}_j, \sum_j)$ using Eq. (10).
  end for
  for all cluster j **do** ▷ Posterior Regularization
    if $\sum_i d_i p_j^{(i)} > C$ **then**
      $\boldsymbol{p}_j \leftarrow r\boldsymbol{p}_j$ (multiply column j of P by r).
    end if
  end for
  for all point i **do**
    $p_j^{(i)} \leftarrow p_j^{(i)} / \sum_j p_j^{(i)}$ (normalize each row of P).
  end for
end while
Step 3. Cluster Assignment
for all point i **do**
  Let $R_i \leftarrow p_{(1)}^{(i)} - p_{(2)}^{(i)}$,
  where $p_{(1)}^{(i)}$, $p_{(2)}^{(i)}$ are the largest and second largest entries in row i of P.
end for
for all decreasing $i \in R_i$ **do**
  while point i is not assigned or infeasible **do**
    $j' \leftarrow \arg\max_{j \in 1\ldots k} p_j^{(i)}$
    if $Demand(j') + Demand(i) \le C$ **then**
      Assign point i to cluster $\arg\max_{j \in 1\ldots k} p_j^{(i)}$.
    else
      Set $p_j^{(i)} = 0$
    end if
  end while
end for
Step 4. Determine Cluster Medians
for all cluster j **do**
  Let $i = \arg\min_{i \in 1\ldots n} \sum_{j \ne i} d(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)})$,
  set point i as median for cluster j.
end for
Step 5. Local Search
while improvement **do**
  Best-improvement local search with shift neighborhood.
  Best-improvement local search with swap neighborhood.
end while

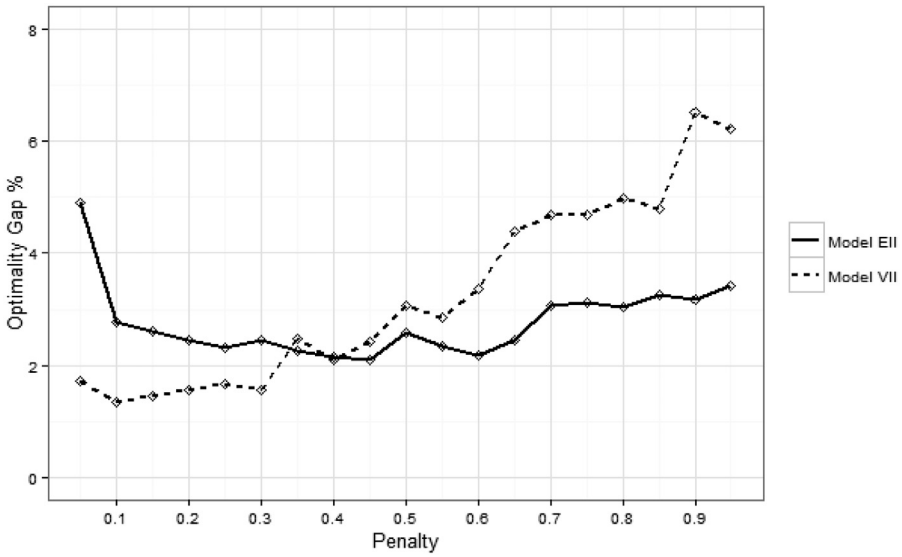

Fig. 3. Effect of different posterior penalty parameters and Gaussian parsimonious models.

therefore has the potential to perform better when node demands are more heterogeneous. Also, a posterior penalty r between 0.1 and 0.3 seems to offer the best results for the Model VII solutions. We choose Model VII with penalty parameter $r = 0.1$ for subsequent studies.

We compare results obtained by EMPR (Algorithm 1) with other geometrical-based construction heuristics. We implement the Mulvey–Beck (MB) heuristic (Mulvey & Beck, 1984) on the same computing platform and with the same local search procedures. For EMPR (Algorithm 1) and the MB heuristic, we report the best results from 40 runs, which is a standard strategy for a randomly initialized clustering method. We also compare the performance with other methods' performances found in literature. The density search constructive method (DSCM) is a method proposed by Ahmadi and Osman (2004), and uses a density function to find cluster centers and then uses a regret function to find assignments. HOC is the naive construction algorithm used in Osman and Christofides (1994).

Table 1 reports the computational results of the standard test instances. In 7 out of 20 instances, EMPR (Algorithm 1) is able to find the optimal solution. Further, the average optimality gap in the construction stage is 1.92% and 0.465% after the local search stage. All these metrics are superior to those of obtained by MB and DSCM heuristics.

6.2. Point pattern and performance of heuristics

After confirming the effectiveness of model-based capacitated clustering with standard test instances, we next investigate the impact of spatial point patterns on the effectiveness of model-based capacitated clustering. According to the definition of Hudson and Fowler (1966), point pattern is the characteristic of a set of points that describes their locations in terms of the relative distances and orientations among them. Point pattern analysis (PPA) has become an important technique in many application areas, particularly in crime analysis, epidemiology, and facility-location planning and management (Boots & Getis, 1985). It is also an essential component of modern geographic information system (GIS) systems (Fotheringham & Rogerson, 2013). However, despite the fact that performance of locational decision making is systematically related of the spatial characteristics (Park, 1989), the analysis of spatial patterns has received surprisingly little attention in the OR community. In the CCP literature, Mulvey and Beck (1984) and Osman and Ahmadi (2007) generated instances where (x, y)'s are either uniformly distributed or drawn from a single normal distribution. To the best of our knowledge, no CCP study has formally investigated the relationship between specific PPA models and heuristic performance.

Intuitively, because we build our method based on Gaussian mixture models, we expect that if the spatial randomness of a CCP shares similar characteristics with a $2d$ Gaussian distribution, then the performance of our model-based approach should be better. Formally, a spatial point pattern consists of the locations of a finite number of points in a region R^d, where the locations are modeled as d-dimensional random variables. A PPA model describes how the points to be clustered arise from a stochastic process. We consider three classes of stochastic process.

- *Homogeneous Poisson Process (HPP):* The HPP, also known as complete spatial randomness (CSR), is defined as: for some $\lambda > 0$, the number Y of events within the region S follows a Poisson distribution with mean $\lambda|S|$, where $|.|$ denotes a two-dimensional area.
- *Modified Thomas Process (MTP):* First described in Diggle, Besag, and Gleaves (1976), the MTP can be used to generate points with natural clustering. It consists of three stages. Firstly, "parent" points are distributed randomly over the plane according to a Poisson process with density λ per unit area. Secondly, each parent independently produces a random number of "offspring" according to a Poisson distribution with mean μ. Lastly, the locations of these offspring are distributed according to the symmetric radial Gaussian with parameter σ.
- *Simple Sequential Inhibition (SSI):* This process can be used to describe points which are regular in pattern. The points are distributed in the area one-by-one. The distribution of each subsequent point is conditional on all previously realized points. More specifically, each new point is generated uniformly in the area, but the new point is rejected if it lies closer than r units from any existing point. The process terminates when desired number of points are generated.

Fig. 4 represents the three basic types of point patterns. HPP generates complete spatial randomness, while the other two show clustering and regularity patterns, respectively. For each of these patterns, we generate 30 test instances with number of nodes

Table 1
Comparison of algorithmic performance on 20 instances from Osman and Christofides (1994).

	EMPR (Algorithm 1)		MB (implemented)		DSCM (reported)		HOC	Optimal
	Const.	LS	Const.	LS	Const.	LS		
1	713	713	713	713	713	713	786	713
2	749	740	740	740	740	740	816	740
3	770	754	779	764	758	753	972	751
4	656	651	651	651	651	651	891	651
5	674	674	696	666	666	666	804	664
6	786	778	820	787	783	778	882	778
7	792	792	811	788	787	787	968	787
8	847	822	846	838	872	839	945	820
9	724	718	718	717	724	724	752	715
10	847	829	841	838	837	837	1017	829
11	1033	1009	1026	1015	1006	1006	1761	1006
12	986	975	976	969	974	970	1567	966
13	1030	1026	1042	1026	1065	1056	1847	1026
14	989	983	1019	988	1009	1009	1635	982
15	1114	1096	1129	1105	1100	1099	1517	1091
16	971	956	973	958	983	979	1780	954
17	1036	1034	1071	1048	1124	1123	1665	1034
18	1089	1058	1088	1053	1073	1062	1345	1043
19	1071	1045	1077	1037	1066	1055	1634	1031
20	1063	1018	1107	1059	1053	1051	1872	1005
Avg. gap (%)	1.920	0.465	2.918	0.933	2.072	1.594	41.575	

Note: Const. considers the solution generated from the construction stage. LS considers the solution after a local search stage.

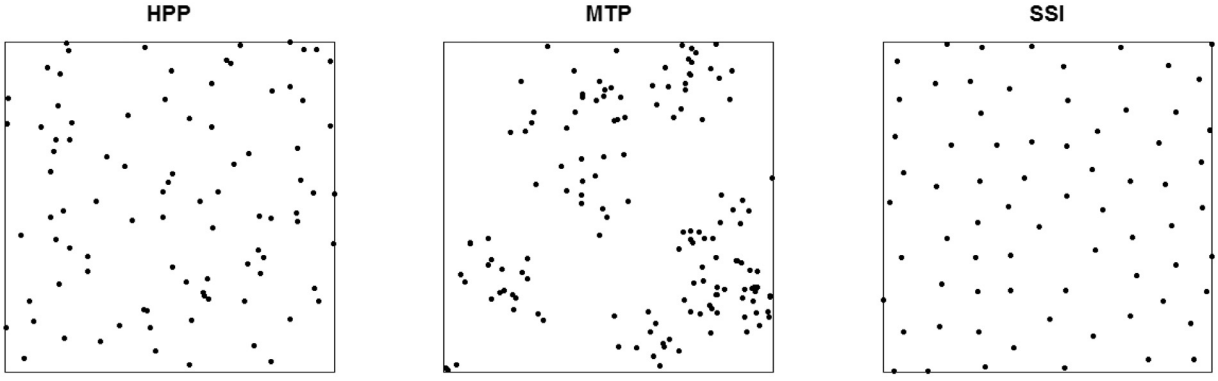

Fig. 4. Examples of different point patterns.

Table 2
Optimality gap (%) under different point patterns.

	HPP (random pattern)	MTP (clustered pattern)	SSI (regular pattern)
EMPR	1.93	1.59	1.88
MB	2.51	4.07	2.96
P-value (paired *t*-test)	< 0.001	0.023	< 0.001

$n = 100$, capacity $C = 120$, and number of clusters $k = 10$. The tightness coefficients (total demand as a percentage of total capacity) are uniformly distributed from 0.6 to 0.8. Given a specified tightness coefficient, the demand for individual nodes are simulated from a symmetric Dirichlet distribution with $\alpha = 2$.

Table 2 summarizes the performance of EMPR (Algorithm 1) and the Mulvey–Beck heuristics on the three different point patterns. We observe significant performance advantages for EMPR across all point patterns. The difference is particularly large when nodes have a natural clustered pattern (as shown in Fig. 4b). The average optimality gap is 4.07% when using Mulvey–Beck, and only 1.59% when using capacitated EM. The results further highlight the need of PPA before applying location-based heuristics, as these heuristics may provide different results for different spatial patterns.

6.3. Stochastic CPMP

We now consider a variation of the CPMP in which the demand of nodes are uncertain. Compared to the deterministic CPMP in Section 3, we make the following modifications to the model.

- Demand at each node is a random variable with a known probability distribution.
- The assignments of cluster medians must be completed before actual demands become known.
- The objective is to minimize the expected total assignment cost.

We formulate the stochastic CPMP using a chance-constrained model. In chance-constrained programing (Charnes & Cooper, 1959), a deterministic linear constraint set $a^T x \leq b$ is replaced by a set of chance-constraints $Pr(a^T x \leq b) \geq 1 - \alpha$. The new constraint

set represents the probability that the deterministic constraint set is satisfied, and α is the allowable probability for the violation.

In the chance-constrained CPMP, we let the d_i's be independent random variables representing node i's demand, and α be the allowed probability that the cluster exceeds its capacity. All other parameters follow the definitions given in Section 3.

$$\text{minimize} \quad \sum_i \sum_j c_{ij} w_{ij} \tag{28}$$

$$\text{subject to} \quad \sum_j w_{ij} = 1 \quad \forall i, \tag{29}$$

$$Pr\left(\sum_i d_i w_{ij} \le y_j C\right) \ge 1-\alpha, \quad \forall j, \tag{30}$$

$$\sum_j y_j = k, \tag{31}$$

$$w_{ij}, y_j \in \{0,1\}, \ \forall i, j. \tag{32}$$

Charnes and Cooper (1959) and researchers in stochastic vehicle routing (for example Gendreau, Laporte, & Séguin, 1996) have shown that chance-constrained models can be transformed into deterministic optimization models. However, the transformed deterministic model is often non-convex and therefore requires significantly more effort to find exact solutions. We adapt Algorithm 1 to the chance-constrained model. We consider two cases: (1) demands follow a Poisson distribution, and (2) demands follow a normal distribution.

6.3.1. Poisson demand

Suppose that the demand in node i follows an independent Poisson distribution with mean μ_i. Because the sum of independent Poisson random variables is also Poisson distributed, the chance constrained capacity constraint can be written in the following deterministic form,

$$\sum_{k=0}^{y_j C} e^{-\sum_i \mu_i w_{ij}} \frac{(\sum_i \mu_i w_{ij})^k}{k!} \ge 1-\alpha, \quad \forall j. \tag{33}$$

In order for the above constraint to be satisfied, we need to first find a Poisson random variable ω with mean $\hat{\mu}$ such that $Pr(\omega \le C) \ge 1-\alpha$, and then let

$$\sum_i \mu_i w_{ij} \le \hat{\mu} y_j, \quad \forall j. \tag{34}$$

For a stochastic CPMP problem, we can use a binary search to find ω because the Poisson CDF is monotonically decreasing in terms of $\hat{\mu}$ (Lin, 2009). Once ω is determined, constraint (34) is equivalent to the capacity constraint in a deterministic CPMP. We omit the discussion of this trivial case.

6.3.2. Normal demand

Suppose that the demands are independently normally distributed with mean μ_i and standard deviation σ_i for node i. In cluster j, the total demand is normally distributed with mean $\Sigma_i \mu_i w_{ij}$ and standard deviation $\sqrt{\sum_i \sigma_i^2 w_{ij}}$. The chance constraint is equivalent to the deterministic constraint,

$$\frac{y_j C - \sum_i \mu_i w_{ij}}{\sqrt{\sum_i \sigma_i^2 w_{ij}}} \ge z_{1-\alpha}, \quad \forall j, \tag{35}$$

which can be rewritten as

$$z_{1-\alpha}\sqrt{\sum_i \sigma_i^2 w_{ij}} + \sum_i \mu_i w_{ij} \le C, \quad \forall j. \tag{36}$$

Table 3
Performance of heuristics on stochastic CPMPs.

α	0.02				0.1			
cv	0.05		0.1		0.05		0.1	
	EMPR	MB	EMPR	MB	EMPR	MB	EMPR	MB
1	721	724	738	762	726	746	726	746
2	740	748	748	748	740	740	755	748
3	784	825	796	856	761	832	784	828
4	657	680	657	679	655	675	655	692
5	683	742	–	768	674	729	683	742
6	782	825	919	1018	792	820	803	952
7	840	811	–	957	820	811	824	815
8	882	960	–	1054	860	962	882	936
9	762	778	–	–	727	729	–	–
10	–	–	–	–	851	1041	–	–
11	1063	1073	1091	1112	1035	1072	1057	1085
12	996	1005	1010	1007	997	1002	994	999
13	1035	1084	1064	1126	1027	1058	1035	1099
14	1026	1047	1040	1141	1012	1074	1020	1065
15	1152	1189	1182	1235	1129	1151	1157	1179
16	996	1022	1032	1124	991	978	990	1031
17	1063	1117	1104	1182	1060	1089	1099	1114
18	1176	1134	1178	1179	1114	1102	1174	1132
19	1085	1123	1159	1118	1111	1088	1096	1135
20	1237	1251	–	–	1122	1224	–	1299
Gap[a]	2.76%		4.22%		3.18%		3.91%	

[a] Gap is defined as the average of (MB-EMPR)/MB.

The equivalent deterministic model is now a nonlinear (non-quadratic) binary program that cannot generally be solved using standard optimization packages. Fortunately, both the Mulvey–Beck heuristic and the EM algorithm can be adapted to solve the chance-constrained CPMP. For the Mulvey–Beck heuristic, we can change how nodes are assigned to current medians. The feasibility check of whether cluster demand is exceeded after an assignment will be replaced by Eq. (36).

For Algorithm 1, we can replace Eq. (24) by

$$p_j^{(i)} \leftarrow r p_j^{(i)} \quad \text{if} \quad z_{1-\alpha}\sqrt{\sum_i \sigma_i^2 p_j^{(i)}} + \sum_i \mu_i p_j^{(i)} > C, \quad \forall j. \tag{37}$$

Table 3 shows the performance comparison between Algorithm 1 (EMPR in Table 3) and Mulvey–Beck heuristic (MB) for the normally-distributed demand case. We assume demand follows a normal distribution with mean equal to the deterministic demand specified in the test instance, and with a known standard deviation generated according to one of two levels of coefficients of variance, $cv = 0.05$ or $cv = 0.1$. We evaluate the performance when $\alpha = 0.02$ and $\alpha = 0.1$. For some instances, the heuristics are not able to generate a feasible solution due to the tightness of capacity constraints.[2] For other test instances, EMPR generally outperforms its geometrical counterpart. On average, EMPR performs better than Mulvey–Beck by a range between 2.76% to 4.22% in our test instances.

7. Case study

We present a case study using a real-world dataset that represents a plausible use case for our modeling framework and solution methodology. Specifically, we use the Sidewalk Café Licenses and Applications dataset from the New York City (NYC) OpenData project (data.cityofnewyork.us). The dataset contains detailed information about sidewalk café license applications filed between 2015 and 2017, including the businesses' names and addresses, latitudes

[2] When the capacity constraint is tight, CCP becomes a bin packing problem in which clustering of the point is less relevant.

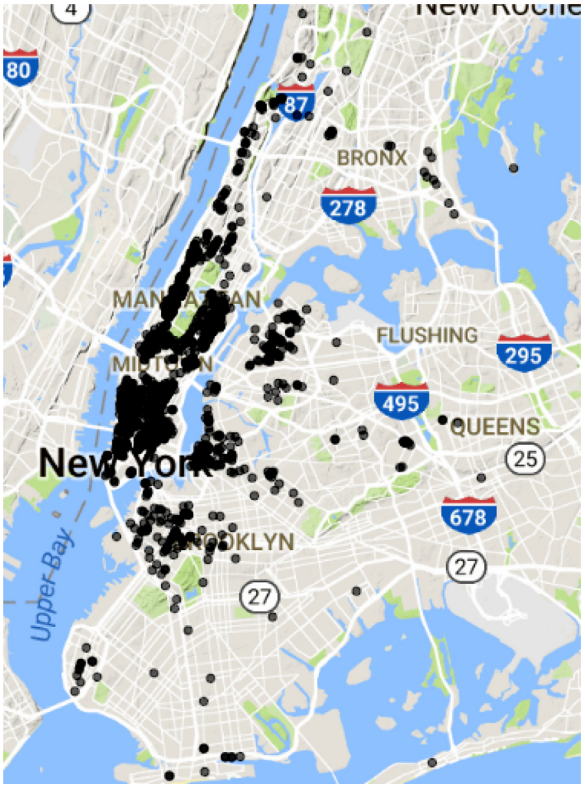

Fig. 5. Distribution of sidewalk cafés in New York City.

and longitudes, number of tables and chairs requested, and application dates. We use the latitudes and longitudes of the cafés as demand locations (Fig. 5). The number of chairs in the application serves as a proxy for the demand at that location.

We consider two different scenarios that require solving CPMP problems. The first scenario is equivalent to the original CPMP formulation in Section 3. That is, we divide the cafés into k clusters, each having a demand capacity C. We create one CPMP instance for each quarter represented in the data, from the fourth quarter of 2015 (2015Q4) to the fourth quarter of 2017 (2017Q4). The number of points in each instance ranges from 142 to 165. We set $k = 5$ for each instance, and we set the demand capacity C based on a tightness coefficient equal to 0.6. A hypothetical example of this scenario could be of a delivery company that needs to set up five mobile depots in the NYC area to deliver fresh produce or other supplies to new cafés. Each of these mobile depots can be reassigned in each quarter. The total delivery amount in a period cannot exceed the available capacity of the depots. Alternatively, we can consider each quarter as a separate problem instance rather than being temporally connected.

The second scenario tests the robustness of the performance of our model-based capacitated clustering. It is also the scenario most likely to be faced by decision makers in practice. Under this scenario, the decision maker solves a CPMP based on the observed point pattern in the first period and builds permanent depots located at the cluster medians. For the subsequent periods, the decision maker can assign new points to a fixed depot, but cannot choose new medians. Previously, via a simulation study, we showed that the model-based approach performs better if the distribution pattern of the points has a natural clustering rather than being completely random or completely regular. If the generation of the new points in the real-world also conforms to such clustering—for example, new cafés are more likely to be established in business districts or near public transportation—then we expect the assignment costs based on the cluster medians from the model-based approach to be well-managed.

Table 4
Comparison of solutions using the NYC-café dataset.

	EMPR (Algorithm 1)		Mulvey–Beck		Optimal
	Obj.	Gap	Obj.	Gap	
2015Q4	34,927	0.43%	35,195	1.20%	34,779
2016Q1	30,708	1.42%	30,803	1.74%	30,277
2016Q2	34,427	3.03%	34,766	4.04%	33,415
2016Q3	24,436	2.03%	25,382	5.98%	23,950
2016Q4	29,476	0.14%	29,837	1.36%	29,436
2017Q1	32,614	0.00%	34,591	6.06%	32,614
2017Q2	38,930	0.69%	39,505	2.18%	38,663
2017Q3	28,183	1.09%	28,349	1.69%	27,879
2017Q4	24,782	0.50%	25,557	3.65%	24,658
Avg. gap		1.04%		3.10%	

We first compare the performance of EMPR (Algorithm 1) and Mulvey–Beck heuristic for the 9 new test instances (NYC-Café). As we have seen previously, the penalty coefficient r in our algorithm is an important tuning parameter. The parameter r that performed well in our previous standard test instances may not perform as well in this particular application. Therefore, we could use a grid search procedure (as in Fig. 3) to find recommendations for r, but each problem requires 20 trials to reach a resolution of 0.05. Here we adopt a more efficient hyper-parameter tuning method called Bayesian optimization (Snoek, Larochelle, & Adams, 2012) to choose an optimal r for each problem. The premise of Bayesian optimization is that similar inputs (r) yield similar outputs (objective). In addition, finding the best hyper-parameter requires both exploration, i.e, trying input values that are different from prior experiments, and exploitation, i.e., trying input values that are close to the best setting so far. Bayesian optimization uses a Gaussian process, a flexible distribution on functions, to fit the relationship between r and the quality of the solution. Bayesian optimization leverages results from prior experiments sequentially to find the most promising input value to try next. When the experiments are time-consuming as in combinatorial optimization problems, Bayesian optimization tends to be more efficient than exhaustive search. We refer the reader to Snoek et al. (2012) for a detailed tutorial of Bayesian optimization. We use the Bayesopt package (Martinez-Cantin, 2014) to search for the optimal r in 20 trials. The optimal r ranges from 0.06 to 0.54 with the average equal to 0.16 in our NYC-Café instances.

Table 4 presents the results for the first scenario. The optimal solutions are obtained from the Gurobi MILP solver. EMPR (Algorithm 1) outperforms the geometrical counterpart for all the test instances. The average optimality gap for EMPR is 1.04% and the gap for Mulvey–Beck is 3.10%. The Wilcoxon Signed-Rank Test confirms that the difference is statistically significant ($W = 0$, critical value $= 5$ at $p = 0.05$) .

Finally, we compare the assignment costs of new points under the second scenario. We solve the CPMP for the test instance generated with the first quarter data (2015Q4) using EMPR (Algorithm 1) and Mulvey–Beck algorithm, respectively. We then retain the cluster medians from the 2015Q4 instance and use them as fixed cluster centers for the eight subsequent quarters. We assign the new points in each of the quarters to the fixed cluster medians using a decreasing order of regret, defined as the difference between the closest and second closest medians. The best-improvement local search procedure is used to improve the assignment of the points. Table 5 compares the assignment costs for the two methods and the percentage improvement obtained using Algorithm 1. As we hypothesized, EMPR out-

Table 5
Comparison of predictive performance of cluster medians.

	EMPR (Algorithm 1)	Mulvey–Beck	Improvement (%)
2016Q1	43,596	48,903	10.85%
2016Q2	46,706	49,968	6.53%
2016Q3	77,234	102,182	24.42%
2016Q4	35,736	36,536	2.19%
2017Q1	45,283	47,260	4.18%
2017Q2	53,025	56,498	6.15%
2017Q3	42,752	46,204	7.47%
2017Q4	41,519	56,861	26.98%
Avg.			11.10%

performs the Mulvey–Beck by an average of 11.10% if the cluster centers were retained from the first period. This provides evidence that the model-based CCP possesses a robust ability to locate the most likely median if both the points and demands arise from a natural clustering process.

8. Conclusion and future research

In this study, we present a new model-based heuristic for the CCP. Our heuristic differs from prior construction heuristics (Ahmadi & Osman, 2004; Mulvey & Beck, 1984) in that a single statistical model is used to describe the objective and the constraint. We use posterior regularization on the EM algorithm to maximize the model likelihood. Our method is competitive on the standard test instances as well as problems generated from a real-world dataset. In addition, we investigate the effect of node-point patterns on the performance of different heuristics. We extend the algorithm to stochastic variants of a CCP, thereby demonstrating the robustness of the framework.

The model-based CCP and the EMPR algorithm provides a promising modeling and solution framework for a wide range of the problems, e.g., social network of actors (Handcock, Raftery, & Tantrum, 2007), genetic data, etc. One benefit of model-based clustering is that it also provides an approach of choosing the number of clusters using model selection techniques in statistics. This may be used to extend our method to solve a more general problem, e.g., simultaneously deciding the location and the number of service depots for customers, while each service depot has a capacity constraint. In terms of methodology extension, Tu, Ball, and Jank (2008) and Jank (2006) propose a stochastic variation of the EM algorithm based on genetic algorithms to search for the global solution; this may provide another promising extension for our approach. Additionally, future researchers could study how to generalize the variance matrices in model-based clustering (e.g. using structures such as Kronecker product) to balance computational complexity and model applicability. Finally, the EM algorithm can be adapted to the Map-Reduce computing paradigm (Chu et al., 2006), and it may be of interest to investigate how the proposed algorithm's speed can be increased in a multi-core machine or computer cluster.

Acknowledgment

The authors thank the editor, Immanuel Bomze, and three anonymous reviewers for many insightful comments. We are grateful to Ibrahim Osman and Samad Ahmadi for providing the test datasets. Our special thanks to Yichen Qin for his guidance on expectation-maximization.

References

Ahmadi, S., & Osman, I. H. (2004). Density based problem space search for the capacitated clustering p-median problem. *Annals of Operations Research, 131*(1–4), 21–43.
Alp, O., Erkut, E., & Drezner, Z. (2003). An efficient genetic algorithm for the p-median problem. *Annals of Operations Research, 122*(1–4), 21–42.
Baldacci, R., Hadjiconstantinou, E., Maniezzo, V., & Mingozzi, A. (2002). A new method for solving capacitated location problems based on a set partitioning approach. *Computers & Operations Research, 29*(4), 365–386.
Banfield, J. D, & Raftery, A. E. (1993). Model-based Gaussian and non-Gaussian clustering. *Biometrics, 49*(3), 803–821.
Barreto, S., Ferreira, C., Paixao, J., & Santos, B. S. (2007). Using clustering analysis in a capacitated location-routing problem. *European Journal of Operational Research, 179*(3), 968–977.
Biernacki, C., Celeux, G., & Govaert, G. (2003). Choosing starting values for the EM algorithm for getting the highest likelihood in multivariate Gaussian mixture models. *Computational Statistics & Data Analysis, 41*(3), 561–575.
Boccia, M., Sforza, A., Sterle, C., & Vasilyev, I. (2008). A cut and branch approach for the capacitated p-median problem based on Fenchel cutting planes. *Journal of Mathematical Modelling and Algorithms, 7*(1), 43–58.
Boots, B. N., & Getis, A. (1985). Point pattern analysis. In G. I. Thrall (Ed.). In *Scientific geography series*. Regional Research Institute, West Virginia University. http://EconPapers.repec.org/RePEc:rri:bkchap:13.
Bozkaya, B., Erkut, E., & Laporte, G. (2003). A tabu search heuristic and adaptive memory procedure for political districting. *European Journal of Operational Research, 144*(1), 12–26.
Celeux, G., & Govaert, G. (1995). Gaussian parsimonious clustering models. *Pattern Recognition, 28*(5), 781–793.
Ceselli, A. (2003). Two exact algorithms for the capacitated p-median problem. *Quarterly Journal of the Belgian, French and Italian Operations Research Societies, 1*(4), 319–340.
Ceselli, A., & Righini, G. (2005). A branch-and-price algorithm for the capacitated p-median problem. *Networks, 45*(3), 125–142.
Charnes, A., & Cooper, W. W. (1959). Chance-constrained programming. *Management Science, 6*(1), 73–79.
Chou, C.-A., Chaovalitwongse, W. A., Berger-Wolf, T. Y., DasGupta, B., & Ashley, M. V. (2012). Capacitated clustering problem in computational biology: Combinatorial and statistical approach for sibling reconstruction. *Computers & Operations Research, 39*(3), 609–619.
Chu, C.-T., Kim, S. K., Lin, Y.-A., Yu, Y., Bradski, G., Ng, A. Y., & Olukotun, K. (2006). Map-reduce for machine learning on multicore. In B. Schölkopf, J. C. Platt, & T. Hoffman (Eds.), *Proceedings of the 19th International Conference on Neural Information Processing Systems (NIPS'06): 19* (pp. 281–288). Cambridge, MA, USA: MIT Press.
Correa, E. S., Steiner, M. T. A., Freitas, A. A., & Carnieri, C. (2004). A genetic algorithm for solving a capacitated p-median problem. *Numerical Algorithms, 35*(2–4), 373–388.
Dasgupta, A., & Raftery, A. E. (1998). Detecting features in spatial point processes with clutter via model-based clustering. *Journal of the American Statistical Association, 93*(441), 294–302.
Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological), 39*(1), 1–38.
Deng, Y., & Bard, J. F. (2011). A reactive GRASP with path relinking for capacitated clustering. *Journal of Heuristics, 17*(2), 119–152.
Díaz, J. A., & Fernandez, E. (2006). Hybrid scatter search and path relinking for the capacitated p-median problem. *European Journal of Operational Research, 169*(2), 570–585.
Diggle, P. J., Besag, J., & Gleaves, J. T. (1976). Statistical analysis of spatial point patterns by means of distance methods. *Biometrics, 32*(3), 659–667.
Fleszar, K., & Hindi, K. S. (2008). An effective VNS for the capacitated p-median problem. *European Journal of Operational Research, 191*(3), 612–622.
Fotheringham, S., & Rogerson, P. (2013). *Spatial analysis and GIS*. CRC Press.
Fraley, C., & Raftery, A. E. (2002). Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association, 97*(458), 611–631.
Fraley, C., & Raftery, A. (2007). Model-based methods of classification: Using the mclust software in chemometrics. *Journal of Statistical Software, 18*(6), 1–13.
Ganchev, K., Graça, J., Gillenwater, J., & Taskar, B. (2010). Posterior regularization for structured latent variable models. *The Journal of Machine Learning Research, 99*, 2001–2049.
Gendreau, M., Laporte, G., & Séguin, R. (1996). Stochastic vehicle routing. *European Journal of Operational Research, 88*(1), 3–12.
Handcock, M. S., Raftery, A. E., & Tantrum, J. M. (2007). Model-based clustering for social networks. *Journal of the Royal Statistical Society: Series A, 170*(2), 301–354.
Hudson, J. C., & Fowler, P. M. (1966). *The concept of pattern in geography*. University of Iowa, Department of Geography.
Jank, W. (2006). The EM algorithm, its randomized implementation and global optimization: Some challenges and opportunities for operations research. In *Perspectives in operations research* (pp. 367–392). Springer.
Jánošíková, L., Herda, M., & Haviar, M. (2017). Hybrid genetic algorithms with selective crossover for the capacitated p-median problem. *Central European Journal of Operations Research, 25*(3), 651–664.
Karlis, D., & Xekalaki, E. (2003). Choosing initial values for the EM algorithm for finite mixtures. *Computational Statistics & Data Analysis, 41*(3), 577–590.

Landa-Torres, I., Del Ser, J., Salcedo-Sanz, S., Gil-Lopez, S., Portilla-Figueras, J. A., & Alonso-Garrido, O. (2012). A comparative study of two hybrid grouping evolutionary techniques for the capacitated p-median problem. *Computers & Operations Research, 39*(9), 2214–2222.

Lin, C. (2009). Stochastic single-source capacitated facility location model with service level requirements. *International Journal of Production Economics, 117*(2), 439–451.

Lorena, L. A., & Senne, E. L. (2004). A column generation approach to capacitated p-median problems. *Computers & Operations Research, 31*(6), 863–876.

Maniezzo, V., Mingozzi, A., & Baldacci, R. (1998). A bionomic approach to the capacitated p-median problem. *Journal of Heuristics, 4*(3), 263–280.

Martinez-Cantin, R. (2014). BayesOpt: A Bayesian optimization library for nonlinear optimization, experimental design and bandits. *The Journal of Machine Learning Research, 15*(1), 3735–3739.

Mulvey, J. M., & Beck, M. P. (1984). Solving capacitated clustering problems. *European Journal of Operational Research, 18*(3), 339–348.

Neal, R. M., & Hinton, G. E. (1998). A view of the em algorithm that justifies incremental, sparse, and other variants. In *Learning in graphical models* (pp. 355–368). Springer.

Osman, I., & Ahmadi, S. (2007). Guided construction search metaheuristics for the capacitated p-median problem with single source constraint. *Journal of the Operational Research Society, 58*(1), 100–114.

Osman, I. H., & Christofides, N. (1994). Capacitated clustering problems by hybrid simulated annealing and tabu search. *International Transactions in Operational Research, 1*(3), 317–336.

Park, S. B. (1989). Performance of successively complex rules for locational decision-making. *Annals of Operations Research, 18*(1), 323–343.

Pirkul, H. (1987). Efficient algorithms for the capacitated concentrator location problem. *Computers & Operations Research, 14*(3), 197–208.

Raftery, A. E., & Dean, N. (2006). Variable selection for model-based clustering. *Journal of the American Statistical Association, 101*(473), 168–178.

Salema, M. I. G., Barbosa-Povoa, A. P., & Novais, A. Q. (2007). An optimization model for the design of a capacitated multi-product reverse logistics network with uncertainty. *European Journal of Operational Research, 179*(3), 1063–1077.

Scheuerer, S., & Wendolsky, R. (2006). A scatter search heuristic for the capacitated clustering problem. *European Journal of Operational Research, 169*(2), 533–547.

Snoek, J., Larochelle, H., & Adams, R. P. (2012). Practical Bayesian optimization of machine learning algorithms. In F. Pereira, C. J. C. Burges, L. Bottou, & K. Q. Weinberger (Eds.), *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 2 (NIPS'12): Vol. 2* (pp. 2951–2959). USA: Curran Associates Inc..

Tu, Y., Ball, M. O., & Jank, W. S. (2008). Estimating flight departure delay distributions – A statistical approach with long-term trend and short-term pattern. *Journal of the American Statistical Association, 103*(481), 112–125.