Model-Based Learning for Point Pattern Data

Ba-Ngu Vo, Nhan Dam, Dinh Phung, Quang N. Tran, and Ba-Tuong Vo

Abstract—This article proposes a framework for model-based point pattern learning using point process theory. Likelihood functions for point pattern data derived from point process theory enable principled yet conceptually transparent extensions of learning tasks, such as classification, novelty detection and clustering, to point pattern data. Furthermore, tractable point pattern models as well as solutions for learning and decision making from point pattern data are developed.

Index Terms—point pattern, point process, random finite set, multiple instance learning, classification, novelty detection, clustering.

1 Introduction

Point patterns–sets or multi-sets of unordered points–arise in numerous data analysis problems where they are commonly known as 'bags', e.g. in multiple instance learning [1], [2], [3], natural language processing and information retrieval ('bag-of-words') [4], [5], [6], image and scene categorization ('bag-of-visual-words') [7], [8], [9], and in sparse data ('bag-of-features') [10], [11]. A statistical data model, usually specified by the *likelihood* function, plays a fundamental role in model-based data analysis. However, statistical point pattern models have not received much attention in the development of machine learning algorithms for point pattern data.

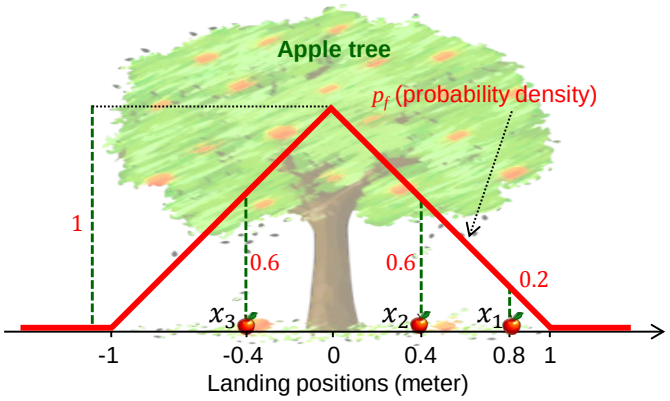


Figure 1. Distribution of landing positions. Position $x_1 = 0.8\,\mathrm{m}$ is 3 times less likely than $x_2 = 0.4\,\mathrm{m}$ and $x_3 = -0.4\,\mathrm{m}$ which are equally likely. Credit: clipartbest.com (apple tree clipart)

To motivate the development of suitable likelihood functions for point patterns, let us consider an example in novelty detection. Suppose that apples from an apple tree land on the ground independently from each other, and that the daily point patterns of landing positions are also independent. Further, the probability density, p_f, of the landing position, learned from 'normal' training data, is shown in Fig. 1. Since the apple landing positions are independent, following common practice (see e.g., [4], [5], [6], [7], [12]) the likelihood that the apples land at positions $x_1, ..., x_m$ is given by the joint (probability) density $p(x_1, ..., x_m)$, which by the independence of the landing positions, is $\prod_{i=1}^{m} p_f(x_i)$.

Suppose we observe one apple landing at x_1 on day 1, and two apples landing at x_2 and x_3 on day 2 (see Fig. 1), which of these daily landing patterns is more likely to be a novelty? The common practice (see e.g., [13]) is to examine the 'normal' likelihoods $p(x_1) = p_f(x_1) = 0.2$ and $p(x_2, x_3) = p_f(x_2)\, p_f(x_3) = 0.36$, from which it is intuitive that the day 1 pattern is novel. However, had we measured distance in centimeters (p_f is scaled by 10^{-2}), then $p(x_1) = 0.002$ is greater than $p(x_2, x_3) = 0.000036$, which contradicts the previous conclusion! This phenomenon arises because $p(x_1)$ is measured in "m^{-1}" or "cm^{-1}" whereas $p(x_2, x_3)$ is measured in "m^{-2}" or "cm^{-2}".

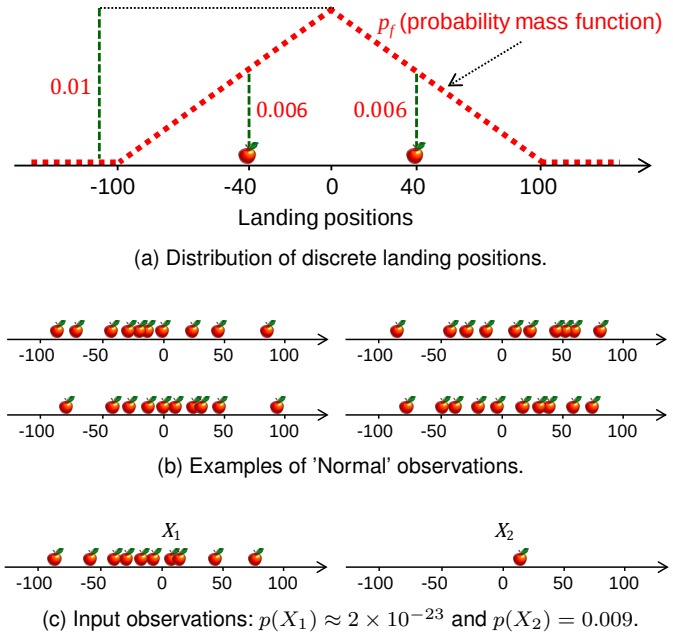


Figure 2. An example with discrete landing positions.

To rule-out the effect of unit incompatibility, we assume only 201 evenly spaced possible landing positions, and a (unit-less) probability mass function on the discrete set $\{-100, \ldots, 100\}$ shown in Fig. 2a (instead of a probability density). Fig. 2b, shows 4 point patterns from the 'normal' training data set, and Fig. 2c shows 2 new observations X_1 and X_2. Since X_2 has only 1 feature, whereas X_1 and the 'normal' observations each has around 10 features, it is intu-

itive that X_2 is novel. However, its likelihood is much higher than that of X_1. This counter intuitive phenomenon arises from the lack of cardinality information in the likelihood.

The above example demonstrates that the joint probability density of the constituent points is not the likelihood of a point pattern. Such likelihood functions could lead to erroneous results in point pattern learning tasks.

This paper proposes a model-based framework for learning from point pattern data using point process theory [14], [15], [16]. Likelihood functions derived from point process theory are probability densities of random point patterns, which incorporate both cardinality and feature information, and avoid the unit of measurement inconsistency. Moreover, they enable the extension of model-based formulations for learning tasks such as classification, novelty detection, and clustering to point pattern data in a conceptually transparent yet principled manner. Such a framework, facilitates the development of tractable point pattern models as well as solutions for learning and decision making. Specifically:

- In classification, we propose solutions based on learning point process models from fully observed training data, and develop an inexpensive classifier using a tractable class of models;
- In novelty detection, where observations are ranked according to their likelihoods, we show that standard point process probability densities are not suitable for point patterns and develop suitable ranking functions;
- In clustering, we introduce point process mixture models, and develop an inexpensive Expectation Maximization clustering algorithm for point pattern using a tractable class of models.

These developments have been partially reported in [17], [18], respectively.

In Section 2 we review basic concepts from point process theory. Subsequent sections present the proposed framework, in progression from: supervised, namely classification, in Section 3; semi-supervised, namely novelty detection, in Section 4; to unsupervised, namely clustering, in Section 5. Numerical studies for these learning tasks are presented in Section 6. Concluding remarks are given in Section 7. We stress that our main contribution is the mathematical framework and tools that facilitates further research into statistical learning for point patterns.

2 Background

This section outlines the elements of point process theory and presents some basic models for point pattern data. For further information, we refer the reader to textbooks such as [14], [15], [16].

2.1 Point Process

A point pattern is a set or multi-set of unordered points. While a multi-set may contain repeated elements, it can be equivalently represented by a set. Specifically, a multi-set with elements $x_1, \ldots, x_m$ of respective multiplicities $N_1, \ldots, N_m$, can be represented as the set $\{(x_1, N_1), \ldots, (x_m, N_m)\}$. A point pattern X can be characterized as a *counting measure* N on the space $\mathcal{X}$ of features, defined, for each (compact) set $A \subseteq \mathcal{X}$ by

$$N(A) = \text{number of points of } X \text{ falling in } A. \quad (1)$$

The values of the counting variables $N(A)$ for all subsets A provide sufficient information to reconstruct the point pattern X [14], [15]. The points of X are the set of x such that $N(\{x\}) > 0$. A point pattern is said to be: *finite* if it has a finite number of points, i.e., $N(\mathcal{X}) < \infty$; and *simple* if it contains no repeated points, i.e., $N(\{x\}) \leq 1$ for all $x \in \mathcal{X}$.

Formally a point process is defined as a *random counting measure*. A random counting measure N may be viewed as a collection of random variables $N(A)$ indexed by $A \subseteq \mathcal{X}$. A point process is *finite* if its realizations are finite almost surely, and *simple* if its realizations are simple almost surely. Likelihoods for point patterns in a countable space is conceptually straightforward, and hereon we only consider point processes on a compact subset $\mathcal{X}$ of $\mathbb{R}^d$.

2.2 Probability Density

In general the probability density of a point process may not exist [19], [20]. To ensure that probability densities are available, we restrict ourselves to simple finite point processes, which are equivalent to *random finite sets* (RFSs) [20], i.e., random variables taking values in $\mathcal{F}(\mathcal{X})$, the space of finite subsets of $\mathcal{X}$.

The probability density $f : \mathcal{F}(\mathcal{X}) \to [0, \infty)$ of a random finite set is usually taken with respect to the dominating measure μ, defined for each (measurable) $\mathcal{T} \subseteq \mathcal{F}(\mathcal{X})$, by (see e.g., [21], [16], [22]):

$$\mu(\mathcal{T}) = \sum_{i=0}^{\infty} \frac{1}{i! U^i} \int \mathbf{1}_{\mathcal{T}}(\{x_1, \ldots, x_i\}) \mathrm{d}(x_1, \ldots, x_i), \quad (2)$$

where U is the unit of hyper-volume in $\mathcal{X}$, $\mathbf{1}_{\mathcal{T}}(\cdot)$ is the indicator function for $\mathcal{T}$, and by convention the integral for $i = 0$ is the integrand evaluated at $\emptyset$. It was shown in [22] that the integral of a function f with respect to μ, given by

$$\int f(X) \mu(\mathrm{d}X) = \sum_{i=0}^{\infty} \frac{1}{i! U^i} \int f(\{x_1, \ldots, x_i\}) \mathrm{d}(x_1, \ldots, x_i), \quad (3)$$

is equivalent to Mahler's set integral [23], [24].

The probability density of a random finite set, with respect to μ, evaluated at $\{x_1, \ldots, x_i\}$ can be written as [19, p. 27] (Eqs. (1.5), (1.6), and (1.7)):

$$f(\{x_1, \ldots, x_i\}) = p_c(i)\, i!\, U^i f_i(x_1, \ldots, x_i), \quad (4)$$

where p_c is the cardinality distribution, and $f_i(x_1, \ldots, x_i)$ is a symmetric function[1] denoting the joint probability density of $x_1, \ldots, x_i$ given cardinality i. Note that by convention $f_0 = 1$ and hence $f(\emptyset) = p_c(0)$. It can be seen from (4) that the probability density f captures the cardinality information as well as the dependence between the features. Also, U^i cancels out the unit of the probability density $f_i(x_1, \ldots, x_i)$ making f unit-less, thereby avoids the unit mismatch.

1. The notations $f_m(x_1, \ldots, x_m)$ and $f_m(\{x_1, \ldots, x_m\})$ can be used interchangeably, since f_m is symmetric.

2.3 Intensity and Conditional Intensity

The *intensity function* λ of a point process is a function on $\mathcal{X}$ such that for any (compact) $A \subset \mathcal{X}$

$$\mathbb{E}\left[N(A)\right] = \int_A \lambda(x)\mathrm{d}x. \tag{5}$$

The intensity value $\lambda(x)$ is interpreted as the instantaneous expected number of points per unit hyper-volume at x.

For a *hereditary* probability density f, i.e., $f(X) > 0$ implies $f(Y) > 0$ for all $Y \subseteq X$, the *conditional intensity* at a point u is given by [20]

$$\lambda\left(u, X\right) = \frac{f\left(X \cup \{u\}\right)}{f(X)}. \tag{6}$$

Loosely speaking, $\lambda(u, X)\mathrm{d}u$ can be interpreted as the conditional probability that the point process has a point in an infinitesimal neighbourhood $\mathrm{d}u$ of u given all points of X outside this neighbourhood. The intensity function is related to the conditional intensity by $\lambda(u) = \mathbb{E}\left[\lambda(u, X)\right]$.

The probability density of a finite point process is completely determined by its conditional intensity [15], [16]. Certain point process models are convenient to formulate in terms of the conditional intensity rather than probability density. Using the conditional intensity also eliminates the normalizing constant needed for the probability density. However, the functional form of the conditional intensity must satisfy certain consistency conditions.

2.4 IID-Cluster Model

Imposing the independence assumption among the features, the model in (4) reduces to the *IID-cluster* model [14], [15]:

$$f(X) = p_c(|X|)\,|X|!\,\left[Up_f\right]^X, \tag{7}$$

where $|X|$ denotes the cardinality (number of elements) of X, p_f is a probability density on $\mathcal{X}$, referred to as the *feature density*, and $h^X \triangleq \prod_{x\in X} h(x)$, with $h^{\emptyset} = 1$ by convention.

When the cardinality distribution p_c is Poisson with rate ρ we have the celebrated Poisson point process [14], [15].

$$f(X) = \rho^{|X|}\, e^{-\rho}\left[Up_f\right]^X. \tag{8}$$

The Poisson point process model is completely determined by the intensity function $\lambda = \rho p_f$, which also equals its conditional intensity. Note that the Poisson cardinality distribution is described by a single non-negative number ρ, hence there is only one degree of freedom in the choice of cardinality distribution for the Poisson point process model.

2.5 Finite Gibbs Model

A general model that accommodates dependence between its elements is a finite Gibbs process, which has probability density of the form [15], [16]

$$f(X) = \exp\left(V_0 + \sum_{i=1}^{|X|}\sum_{\{x_1,\ldots,x_i\}\subseteq X} V_i\left(x_1,\ldots,x_i\right)\right), \tag{9}$$

where V_i is called the i^{th} potential, given explicitly by

$$V_i\left(x_1,\ldots,x_i\right) = \sum_{Y\subseteq\{x_1,\ldots,x_i\}} (-1)^{|\{x_1,\ldots,x_i\}|-|Y|}\log f(Y).$$

Note that any hereditary probability density of a finite point process can be expressed in the Gibbs form [20]. The Poisson point process is indeed a first order Gibbs model. Gibbs models arise in statistical physics, where $\log f(X)$ may be interpreted as the potential energy of the point pattern. The term $-V_1(x)$ can be interpreted as the energy required to create a single point at a location x, and the term $-V_2(x_1, x_2)$ can be interpreted as the energy required to overcome the force between the points x_1 and x_2.

The next three sections show how point process models are used in model-based point pattern classification, novelty detection and clustering.

3 Model-Based Classification

Classification is the supervised learning task that uses fully-observed training input-output pairs $\mathcal{D}_{\text{train}} = \{(X_n, y_n)\}_{n=1}^N$ to determine the output class label $y \in \{1,\ldots,K\}$ of each input observation [25], [26], [27], [28]. This fundamental machine learning task is the most widely used form of supervised machine learning, with applications spanning many fields of study.

Model-based classifiers for point pattern data have not been investigated. In multiple instance learning, existing classifiers in the Bag-Space paradigm are based on distances between point patterns, such as Hausdorff [29], [30], Chamfer [31], Earth Mover's [32], [33]. Such classifiers do not require any underlying data models and are simple to use. However, they may perform poorly with high dimensional inputs due to the curse of dimensionality, and are often computationally intractable for large datasets [26], not to mention that the decision procedure is unclear. On the other hand, knowledge of the underlying data model can be used to exploit statistical patterns in the training data, and to devise optimal decision procedures.

Using the notion of probability density for point process from subsection 2.2, the standard model-based classification formulation directly extends to point pattern classification:

- In the *training phase*, we seek likelihoods that 'best' fit the training data. Specifically, for each $k \in \{1,\ldots,K\}$, we seek a likelihood function $f(\cdot\,|\,y = k)$ that best fit the training input point patterns in $\mathcal{D}_{\text{train}}^{(k)} = \{X : (X, k) \in \mathcal{D}_{\text{train}}\}$, according to criteria such as *maximum likelihood* (ML) or Bayes optimal if suitable priors on the likelihoods are available.
- In the *classifying phase*, the likelihoods (learned from training data) are used to classify input observations. When a point pattern X is passed to query its label, the Bayes classifier returns the mode of the class label posterior $p\left(y = k \mid X\right)$ computed from the likelihood and the class prior p via Bayes' rule:

$$p\left(y = k \mid X\right) \propto p(y = k) f(X \mid y = k). \tag{10}$$

The simplest choices for the class priors are the uniform distribution, and the categorical distribution, usually estimated from the training data via

$$p(y = k) = \frac{1}{N_{\text{train}}}\sum_{n=1}^{N_{\text{train}}} \delta_{y_n}[k],$$

where $\delta_i[j]$ is the Kronecker delta, which takes on the value 1 when $i = j$, and zero otherwise. Hence, the main

computational effort in model-based classification lies in the training phase.

3.1 Learning Point Process Models

Learning the likelihood function for class k boils down to finding the value(s) of the parameter θ_k such that the (parameterized) probability density $f(\cdot \mid y = k, \theta_k)$ best explains the observations $X_1, \ldots, X_N$ in $\mathcal{D}^{(k)}_{\text{train}}$. In this subsection, we consider a fixed class label and its corresponding observations $X_1, \ldots, X_N$, and omit the dependence on k.

Methods for learning point process models have been available since the 1970's, see e.g., [16], [20]. We briefly summarize some recognized techniques and presents ML for IID-cluster models as a tractable point pattern classification solution.

3.1.1 Model fitting via summary statistics

The method of moments seeks the parameter θ such that the expectation of a given statistic of the model point process parameterized by θ is equal to the statistic of the observed point patterns [20]. However, this approach is only tractable when the solution is unique and the expectation is a closed form function of θ, which is usually not the case in practice, not to mention that moments are difficult to calculate.

The method of minimum contrast seeks the parameter θ that minimizes some dissimilarity between the expectation of a given summary statistic (e.g., the K-function) of the model point process and that of the observed point patterns [20]. Provided that the dissimilarity functional is convex in the parameter θ, this approach can avoid some of the problems in the method of moments. However, in general the statistical properties of the solution are not well understood, not to mention the numerical behaviour of the algorithm used to determine the minimum.

3.1.2 Maximum likelihood (ML)

In the ML approach, we seek the ML estimate (MLE) of θ:

$$\text{MLE}\left(f(\cdot\,|\,\theta); X_{1:N}\right) \triangleq \underset{\theta}{\text{argmax}} \left(\prod_{n=1}^{N} f(X_n \,|\, \theta) \right). \qquad (11)$$

The MLE has some desirable statistical properties such as asymptotic normality and optimality [20]. However, in general, there are problems with non-unique maxima. Moreover, analytic MLEs are not available because the likelihood (9) of many Gibbs models contains an intractable normalizing constant (which is a function of θ) [16].

To the best of our knowledge, currently there is no general ML technique for learning generic models such as Gibbs from real data. Numerical approximation methods in [34] and Markov Chain Monte Carlo (MCMC) methods in [21] are highly specific to the chosen model, computationally intensive, and require careful tuning to ensure good performance. Nonetheless, simple models such as the IID-cluster model (7) admits an analytic MLE (see subsection 3.1.4).

Remark: The method of estimating equation replaces the ML estimation equation

$$\nabla \left(\sum_{n=1}^{N} \log\left(f(X_n \,|\, \theta)\right) \right) = 0 \qquad (12)$$

by an unbiased sample approximation $\sum_{n=1}^{N} \Psi(\theta, X_n) = 0$ of the general equation $\mathbb{E}_\theta\left[\Psi(\theta, X)\right] = 0$, For example, $\Psi(\theta, X_n) = \nabla \log(f(X_n \mid \theta))$ results in ML since it is well-known that (12) is an unbiased estimating equation. Setting $\Psi(\theta, X_n)$ to the difference between the empirical value and the expectation of the summary statistic results in the method of moments. Takacs-Fiksel is another well-known family of estimating equations [35], [36].

3.1.3 Maximum pseudo-likelihood

Maximum pseudo-likelihood (MPL) estimation is a powerful approach that avoids the intractable normalizing constant present in the likelihood while retaining desirable properties such as consistency and asymptotic normality in a large-sample limit [37], [38]. The key idea is to replace the likelihood of a point process (with parameterized conditional intensity $\lambda_\theta(u; X)$) by the pseudo-likelihood:

$$\text{PL}\left(\theta; X_{1:N}\right) = \prod_{n=1}^{N} e^{-\int \lambda_\theta(u; X_n) \mathrm{d}u} \left[\lambda_\theta\left(\cdot; X_n\right)\right]^{X_n}. \qquad (13)$$

The rationale behind this strategy is discussed in [37]. Up to a constant factor, the pseudo-likelihood is indeed the likelihood if the model is Poisson, and approximately equal to the likelihood if the model is close to Poisson. The pseudo-likelihood may be regarded as an approximation to the likelihood which neglects the inter-point dependence.

An MPL algorithm has been developed by Baddeley and Turner in [39] for point processes with sufficient generality such as Gibbs whose conditional intensity has the form

$$\lambda\left(u, X\right) = \exp\left(\sum_{i=1}^{|X|+1} \sum_{\{x_1,\ldots,x_{i-1}\} \subseteq X} V_i\left(u, x_1, \ldots, x_{i-1}\right) \right).$$

By turning the pseudo-likelihood of a general point process into a classical Poisson point process likelihood, MPL can be implemented with standard generalized linear regression software [39]. Due to its versatility, the Baddeley-Turner algorithm is the preferred model fitting tool for point processes.

The main hurdle in the application of the Baddeley-Turner algorithm to point pattern classification is the computational requirement. While this may not be an issue in spatial statistics applications, the computational cost is still prohibitive with large data sets often encountered in machine learning. On the other hand, disadvantages of MPL (relative to ML) such as small-sample bias and inefficiency [38], [40] become less significant with large data. Efficient algorithms for learning general point process models is an on going area of research.

3.1.4 ML learning for IID-clusters

Computationally efficient algorithms for learning point process models are important because machine learning usually involve large data sets (compared to applications in spatial statistics). Since learning a general point process is computationally prohibitive, the IID-cluster model (7) provides a good trade-off between tractability and versatility by neglecting interactions between the points.

Since an IID-cluster model is uniquely determined by its cardinality and feature distributions, we consider a parameterization of the form:

$$f(X\,|\,\xi,\varphi) = p_\xi(|X|)\;|X|!\;U^{|X|}\;p_\varphi^X, \tag{14}$$

where p_ξ and p_φ, are the cardinality and feature distributions parameterized by ξ and φ, respectively. Learning the underlying parameters of an IID-cluster model amounts to estimating the parameter $\theta = (\xi, \varphi)$ from training data.

The form of the IID-cluster likelihood function allows the MLE to separate into the MLE of the cardinality parameter ξ and MLE of the feature parameter φ. This is stated more concisely in Proposition 1 (the proof is straightforward, but included for completeness).

Proposition 1. *Let $X_1, \ldots, X_N$ be N i.i.d. realizations of an IID-cluster with parameterized cardinality distribution p_ξ and feature density p_φ. Then the MLE of (ξ, φ), is given by*

$$\hat{\xi} = \text{MLE}\left(p_\xi; |X_1|, \ldots, |X_N|\right), \tag{15}$$

$$\hat{\varphi} = \text{MLE}\left(p_\varphi; \uplus_{n=1}^N X_n\right), \tag{16}$$

where $\uplus$ denotes disjoint union.

Proof: Using (14), we have

$$\prod_{n=1}^N f(X_n\,|\,\xi,\varphi) = \prod_{n=1}^N p_\xi(|X_n|)\;|X_n|!\;U^{|X_n|}p_\varphi^{X_n}$$

$$= \prod_{n=1}^N |X_n|!\;U^{|X_n|}\prod_{n=1}^N p_\xi(|X_n|)\prod_{n=1}^N p_\varphi^{X_n}$$

Hence, to maximize the likelihood we simply maximize the second and last products in the above separately. This is achieved with (15) and (16). □

Observe from Proposition 1 that the MLE of the feature density parameter is identical to that used in NB. For example: if the feature density is a Gaussian, then the MLEs of the mean and covariance are

$$\hat{\mu} = \frac{1}{N}\sum_{n=1}^N\sum_{x\in X_n} x, \tag{17}$$

$$\hat{\Sigma} = \frac{1}{N}\sum_{n=1}^N\sum_{x\in X_n}(x-\hat{\mu})(x-\hat{\mu})^{\mathrm{T}}; \tag{18}$$

if the feature density is a Gaussian mixture, then the MLE of the Gaussian mixture parameters can be determined by the EM algorithm. Consequently, learning the IID-cluster model requires only one additional, but relatively inexpensive, task of computing the MLE of the cardinality parameters.

For a categorical cardinality distribution, i.e., $\xi = (\xi_1, \ldots, \xi_M)$ where $\xi_k = \Pr(|X| = k)$ and $\sum_{k=1}^M \xi_k = 1$, the MLE of the cardinality parameter is given by

$$\hat{\xi}_k = \frac{1}{N}\sum_{n=1}^N \delta_k[|X_n|]. \tag{19}$$

Note that to avoid over-fitting, the standard practice of placing a Laplace prior on the cardinality distribution can be applied, i.e. replacing the above equation by $\hat{\xi}_k \propto \epsilon + \sum_{n=1}^N \delta_k[|X_n|]$, where ϵ is a small number.

For a Poisson cardinality distribution parameterized by the rate $\xi = \rho$, the MLE is given by

$$\hat{\rho} = \frac{1}{N}\sum_{n=1}^N |X_n|. \tag{20}$$

It is also possible to derive MLEs for other families of cardinality distributions such as Panjer, multi-Bernoulli, etc.

Numerical results for point pattern classification, in which ML is used to learn Poisson and IID-cluster models, are given in subsection 6.1. The complexity of IID-cluster MLE is the same as NB, which is $O(NId)$ for training, and $O(KId)$ for classifying, where I is the average number of features per point pattern and d is the dimension of the features.

Remark: Proposition 1 also extends to Bayesian learning for IID-clusters if the prior on (ξ, φ) separates into priors on ξ and φ. Following the arguments in the proof of Proposition 1, the maximum aposteriori (MAP) estimate of (ξ, φ) separates into MAP estimates of ξ and φ. Typically a (symmetric) Dirichlet distribution $\text{Dir}(\cdot\,|\,\eta/K, \ldots, \eta/K)$, with dispersion η on the unit M-simplex, can be used as a prior on the categorical cardinality distribution. The prior for φ depends on the form of the feature density p_φ (see also subsection 5.2 for conjugate priors of the Poisson model).

4 Model-Based Novelty Detection

Novelty detection is the semi-supervised task of identifying observations that are significantly different from the rest of the data [13], [41]. In novelty detection, there is no novel training data, only 'normal' training data is available [41]. Hence it is not a special case of classification nor clustering [42], [43], and is a separate problem in its own right.

Similar to classification, novelty detection involves a training phase and a detection phase. Since novel training data is not available, input observations are ranked according to how well they fit the 'normal' training data and those not well-fitted are deemed novel or anomalous [42], [43]. The preferred measure of goodness of fit is the 'normal' likelihoods of the input data. To the best of our knowledge, there are no novelty detection solutions for point pattern data in the literature.

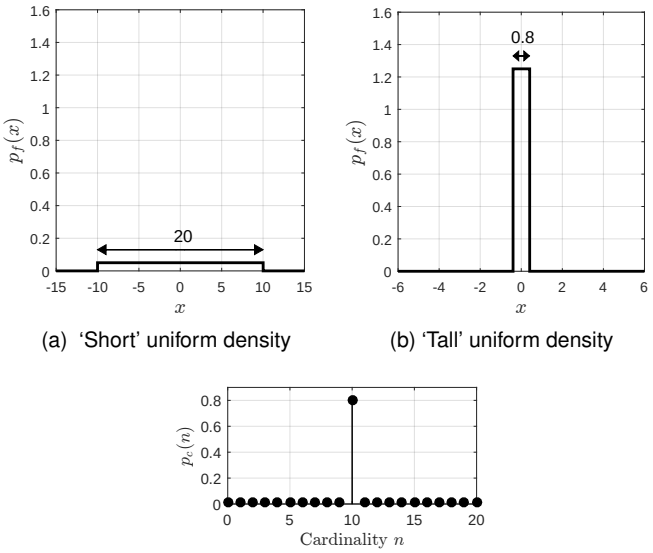
In this section we present a model-based solution to point pattern novelty detection. The training phase in novelty detection is the same as that for classification. However, in the detection phase the ranking of likelihoods is not applicable to point pattern data, even though point process probability density functions are unit-less and incorporates both feature and cardinality information. In subsection 4.1, we discuss why such probability densities are not suitable for ranking input point patterns, while in subsection 4.2 we propose a suitable ranking function for novelty detection.

4.1 Probability Density and Likelihood

This subsection presents an example to illustrate that the probability density of a point pattern does not necessarily indicate how likely it is. For this example, we reserve the term *likelihood* for the measure of how likely or probable a realization is.

Consider two IID-cluster models with different uniform feature densities and a common cardinality distribution as shown in Fig. 3. Due to the uniformity of p_f, it follows from (7) that point patterns from each IID-cluster model with the same cardinality have the same probability density. Note from [14] that to sample from an IID-cluster model,

we first sample the number of points from the cardinality distribution, and then sample the corresponding number of points independently from the feature distribution. For an IID-cluster model with uniform feature density, the joint distribution of the features is completely uninformative (total uncertainty) and so the likelihood of a point pattern should be proportional to the probability of its cardinality.

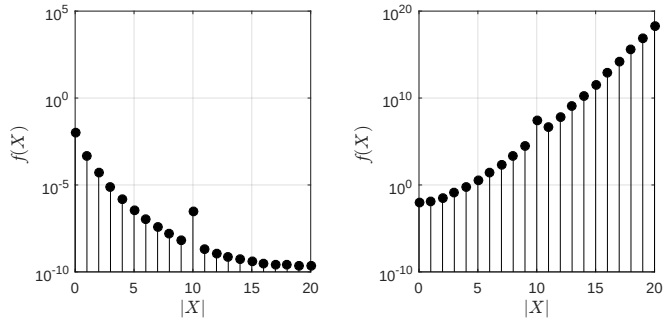


(a) 'Short' uniform density (b) 'Tall' uniform density

(c) Cardinality distribution on {0,...,20} with one mode at 10, the remaining cardinalities are equally likely with total mass 0.2.

Figure 3. Feature and cardinality distributions for 2 IID-clusters.

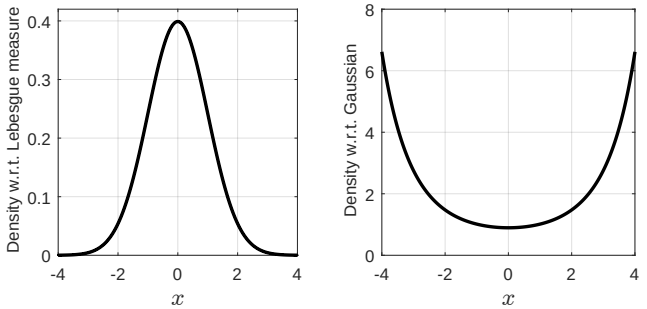
If the probability density were an indication of how likely a point pattern is, then the plot of probability density against cardinality should resemble the cardinality distribution. However, this is not the case. Fig. 4 indicates that for the IID-cluster with 'short' feature density, the probability density tends to decrease with increasing cardinality (Fig. 4a). This phenomenon arises because the feature density given cardinality n is $(1/20)^n$, which vanishes faster than the $n!$ growth (for $n \leq 20$). The converse is true for the IID-cluster with 'tall' feature density (Fig. 4b). Thus, point patterns with highest/least probability density are not necessarily the most/least probable.



(a) 'Short' uniform feature density (b) 'Tall' uniform feature density

Figure 4. Probability density vs cardinality for 2 IID-clusters.

Such problem arises from the non-uniformity of the reference measure. A measure μ is said to be uniform if for any measurable region A with $\mu(A) < \infty$, all points of A (except on set of measure zero) are equi-probable under the probability distribution $\mu/\mu(A)$. One example is the Lebesgue measure vol on $\mathbb{R}^n$: given any bounded measurable region A, all realizations in A are equally likely under the probability distribution $vol(\cdot)/vol(A)$. The probability density $f(X) = P(\mathrm{d}X)/\mu(\mathrm{d}X)$ (as a Radon-Nikodym derivative) at a point X is the ratio of probability measure to reference measure at an infinitesimal neighbourhood of X. Hence, unless the reference measure is uniform, $f(X)$ is not a measure of how likely X is. This is also true even for probability densities on the real line. For example, the probability density of a zero-mean Gaussian distribution with unit variance relative to the (uniform) Lebesgue measure is the usual Gaussian curve shown in Fig. 5a, while its density relative to a zero-mean Gaussian distribution with variance 0.8 is shown in Fig. 5b, where the most probable point has the least probability density value.



(a) (b)

Figure 5. Density of a zero-mean unit-variance Gaussian w.r.t.: (a) Lebesgue measure; (b) zero-mean Gaussian with variance 0.8.

The reference measure μ defined by (2) is not uniform because for a bounded region $\mathcal{T} \subseteq \mathcal{F}(\mathcal{X})$, the probability distribution $\mu/\mu(\mathcal{T})$ is not necessarily uniform (unless all points of $\mathcal{T}$ have the same cardinality). Hence, probability densities of input point patterns relative to μ are not indicative of how well they fit the 'normal' data model.

Remark: In novelty detection we are interested in the likelihood of the input point pattern whereas in Bayesian classification we are interested in its likelihood ratio. Using standard properties of the Radon-Nikodym derivative and relevant assumptions on absolute continuity, the posterior class probability

$$\begin{aligned} p\left(y \mid X\right) &= \frac{p(y)f(X\,|\,y)}{\int p(y)f(X\,|\,y)\mathrm{d}y} \\ &= \frac{p(y)P(\mathrm{d}X\,|\,y)/\mu(\mathrm{d}X)}{\int p(y)(P(\mathrm{d}X\,|\,y)/\mu(\mathrm{d}X))\mathrm{d}y} \\ &= \frac{p(y)P(\mathrm{d}X\,|\,y)}{\int p(y)P(\mathrm{d}X\,|\,y)\mathrm{d}y}, \end{aligned}$$

which is invariant to the choice of reference measure. In essence, the normalizing constant cancels out the influence of the reference measure, and hence, problems with the non-uniformity of the reference measure do not arise.

4.2 Ranking Functions

To the best of our knowledge, it is not known whether there exists a uniform reference measure on $\mathcal{F}(\mathcal{X})$ that dominates the probability distributions of interest (so that they admit

densities). In this subsection, we propose a suitable point pattern ranking function for novelty detection by modifying the probability density.

The probability density (4) is the product of the cardinality distribution $p_c(|X|)$, the cardinality-conditioned feature (probability) density $f_{|X|}(X)$, and a trans-dimensional weight $|X|!\,U^{|X|}$. Note that the cardinality distribution and the conditional joint feature density completely describes the point process. The conditional density $f_{|X|}(X)$ enables the ranking of point patterns of the same cardinality, but cannot be used to rank across different cardinalities because it takes on different units of measurement. The weights $|X|!\,U^{|X|}$ reconcile for the differences in dimensionality and unit of measurement between $f_{|X|}(X)$ of different cardinalities. However, the example in subsection 4.1 demonstrates that weighting by $|X|!\,U^{|X|}$ leads to probability densities that are inconsistent with likelihoods.

In the generalization of the maximum aposteriori (MAP) estimator to point patterns [24], Mahler circumvented such inconsistency by replacing $|X|!\,U^{|X|}$ with $c^{|X|}$, where c is an arbitrary constant. Specifically, instead of maximizing the probability density $f(X)$, Mahler proposed to maximize $f(X)c^{|X|}/\,|X|!$. This generalized MAP estimate depends on the choice of the free parameter c.

Inspired by Mahler's generalized MAP estimator, we replace the weight $|X|!\,U^{|X|}$ in the probability density by a general function of the cardinality $C(|X|)$, resulting in a ranking function of the form

$$r(X) = p_c(|X|)\,C(|X|)\,f_{|X|}(X). \tag{21}$$

The example in subsection 4.1 demonstrated that, as a function of cardinality, the ranking should be proportional to the cardinality distribution, otherwise unlikely samples can assume high ranking values. In general, the ranking function is not solely dependent on the cardinality, but also varies with the features. Nonetheless, the example suggests that the ranking function, on average, should be proportional to the cardinality distribution. Hence, we impose the following consistency requirement: for a given cardinality n, the expected ranking value is proportional to the probability of cardinality n, i.e.,

$$\mathbb{E}_{X||X|=n}\left[r(X)\right] \propto p_c(n). \tag{22}$$

Proposition 2. *For a point process with probability density (4), a ranking function consistent with the cardinality distribution, i.e., satisfies (22), is given by*

$$r(X) \propto \frac{p_c(|X|)}{\|f_{|X|}\|_2^2} f_{|X|}(X) \tag{23}$$

where $\|\cdot\|_2$ denotes the L_2-norm.

Proof: Noting $f(X \mid |X| = n) = n!\,U^n\,f_n(X)\,\delta_n[|X|]$ from (4), we have

$$\begin{aligned}
&\mathbb{E}_{X||X|=n}\left[f_n(X)\right] \\
&= \int f_n(X) f(X\,||X| = n) \mu(\mathrm{d}X) \\
&= \int n!\,U^n (f_n(X))^2\,\delta_n[|X|] \mu(\mathrm{d}X) \\
&= \sum_{i=0}^{\infty} \frac{n!U^n}{i!U^i} \int (f_n(\{x_1,\ldots,x_i\}))^2\,\delta_n[i]\,\mathrm{d}(x_1,\ldots,x_i)
\end{aligned}$$

where the last step follows from definition (3) of the integral with respect to μ. Further due to $\delta_n[i]$, only the nth term in the sum remains, i.e.

$$\begin{aligned}
\mathbb{E}_{X||X|=n}[f_n(X)] &= \frac{n!U^n}{n!U^n}\int (f_n(\{x_1,\ldots,x_n\}))^2 \mathrm{d}(x_1,\ldots,x_n) \\
&= \|f_n\|_2^2.
\end{aligned}$$

Hence taking the expectation of $r(X)$ in (23), we have

$$\begin{aligned}
\mathbb{E}_{X||X|=n}\left[r(X)\right] &\propto \mathbb{E}_{X||X|=n}\left[\frac{p_c(|X|)}{\|f_{|X|}\|_2^2} f_{|X|}(X)\right] \\
&= \frac{p_c(n)}{\|f_n\|_2^2}\mathbb{E}_{X||X|=n}\left[f_n(X)\right] \\
&= p_c(n).
\end{aligned}$$

□

Note that $\|f_{|X|}\|_2^2$ has units of $U^{-|X|}$, which is the same as the unit of $f(X)$, rendering the ranking function r unitless, thereby avoids the unit of measurement inconsistency described in Section 1.

For an IID-cluster with feature density p_f the ranking function reduces to

$$r(X) \propto p_c(|X|)\left(\frac{p_f}{\|p_f\|_2^2}\right)^X. \tag{24}$$

The feature density p_f, in the example of subsection 4.1, is uniform and so $p_f/\|p_f\|_2^2 = 1$ on its support. Hence the ranking is equal to the cardinality distribution, as expected. Fig. 6 illustrates the effect of dividing a non-uniform feature density p_f, by its energy $\|p_f\|_2^2$: 'tall' densities become shorter and 'short' densities become taller, providing adjustments for multiplying together many large/small numbers.

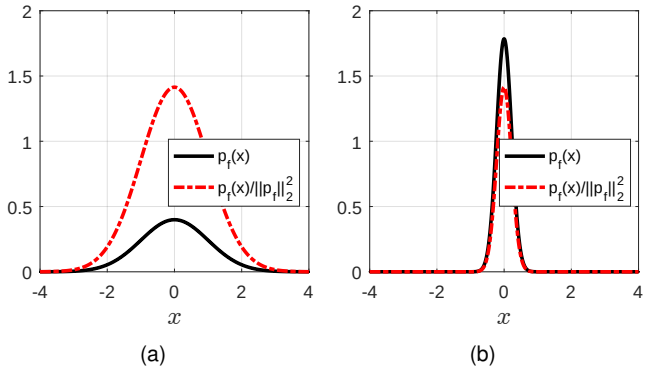


Figure 6. Probability density divided by energy: (a) 'short' Gaussian (mean = 0, variance = 1); (b) 'tall' Gaussian (mean = 0, variance = 0.05).

Numerical results for point pattern novelty detection are given in subsection 6.2, where ML is used to learn a 'normal' Poisson model and input data are ranked via the proposed ranking function. The complexity is the same as NB, which is $O(NId)$ for training, and $O(Id)$ for detection, where I is the average number of features per point pattern.

5 Model-Based Clustering

The aim of clustering is to partition the dataset into groups so that members in a group are similar to each other whilst dissimilar to observations from other groups [44], [45]. A partitioning of a given set of observations $\{X_1,\ldots,X_N\}$ is often represented by the (latent) cluster assignment $y_{1:N}$, where y_n denotes the cluster label for the n^{th} observation.

Clustering is an unsupervised learning problem since the labels are not included in the observations [46], [47]. Indeed it can be regarded as classification without training and is a fundamental problem in data analysis. Comprehensive surveys on clustering can be found in [47], [48].

At present, model-based point pattern clustering have not been investigated. To the best of our knowledge, there are two clustering algorithms for point patterns: the Bag-level MI Clustering (BAMIC) algorithm [49]; and the Maximum Margin MI Clustering (M^3IC) algorithm [50]. BAMIC adapts the k-medoids algorithm with the Hausdorff distance as a measure of dissimilarity between point patterns [49]. On the other hand, in M^3IC, the point pattern clustering problem was posed as a non-convex optimization problem which is then relaxed and solved via a combination of the Constrained Concave-Convex Procedure and Cutting Plane methods [50]. While these algorithms are simple to use, they lack the ability to exploit statistical trends in the data, not to mention computational problems with high dimensional or large datasets [26].

In this section, we propose a model-based approach to the clustering problem for point pattern data. Mixture modeling is the most common probabilistic approach to clustering, where the aim is to estimate the cluster assignment $y_{1:N}$ via likelihood or posterior inference [26]. The point process formalism enables direct extension of mixture models to point pattern data. In particular, the finite mixture point process model for problems with known number of clusters is presented in subsection 5.1, together with an Expectation-Maximization (EM) based solution. The infinite mixture point process model for problems with unknown number of clusters is discussed in subsection 5.2.

5.1 Finite Mixture Model

A finite mixture model assumes K underlying clusters labeled 1 to K, with prior probabilities $\pi_1, \ldots, \pi_K$, and characterized by the parameters $\theta_1, \ldots, \theta_K$ in some space Θ. Let $f(X_n \,|\, \theta_k) \triangleq f(X_n \,|\, y_n = k, \theta_{1:K})$ denote the likelihood of X_n given that cluster k generates an observation. Then

$$f(X_{1:N}, y_{1:N} \,|\, \pi_{1:K}, \theta_{1:K}) = \prod_{n=1}^{N} \pi_{y_n} f(X_n \,|\, \theta_{y_n}), \qquad (25)$$

Marginalizing the joint distribution (25) over the cluster assignment $y_{1:N}$ gives the data likelihood function

$$f(X_{1:N} \,|\, \pi_{1:K}, \theta_{1:K}) = \prod_{n=1}^{N} \sum_{k=1}^{K} \pi_k \, f(X_n \,|\, \theta_k). \qquad (26)$$

Thus, in a finite mixture model, the likelihood of an observation is a mixture of K probability densities. Hence the application of the finite mixture model requires the number of clusters to be known apriori. The posterior probability of cluster label $y_n = k$ (i.e., the probability that, given $\pi_{1:K}, \theta_{1:K}$ and X_n, cluster k generates X_n) is

$$p(y_n = k \,|\, X_n, \pi_{1:K}, \theta_{1:K}) = \frac{\pi_k \, f(X_n \,|\, \theta_k)}{\sum_{\ell=1}^{K} \pi_\ell \, f(X_n \mid \theta_\ell)}. \qquad (27)$$

Under a mixture model formulation, clustering can be treated as an incomplete data problem since only the $X_{1:N}$ of the complete data $\mathcal{D} = \{(X_n, y_n)\}_{n=1}^{N}$ is observed and the cluster assignment $y_{1:N}$ is unknown or missing. We seek $y_{1:N}$, and the mixture model parameter

$$\psi \triangleq (\pi_{1:K}, \theta_{1:K}) \qquad (28)$$

that best explains the observed data $X_{1:N}$ according to a given criterion such as ML or optimal Bayes. ML is intractable in general and often requires the EM algorithm [51], [52] to find approximate solutions. Optimal Bayes requires suitable priors for ψ. Typically the prior for $\pi_{1:K}$ is a Dirichlet distribution $\mathrm{Dir}(\cdot \mid \eta/K, \ldots, \eta/K)$ with dispersion η, while the prior for $\theta_{1:K}$ is model-specific, depending on the form of the likelihood $f(X_n \,|\, \theta_k)$. Computing the cluster label posterior $p(y_{1:N} \,|\, X_{1:N})$ or the joint posterior $p(y_{1:N}, \psi \,|\, X_{1:N})$ are intractable in general and Markov Chain Monte Carlo methods, such as Gibbs sampling are often needed [53], [54].

Next we detail an ML solution to point pattern clustering using EM with an IID-cluster mixture model. Extension to infinite mixture model for unknown number of clusters is discussed in subsection 5.2.

5.1.1 EM clustering via IID-cluster mixture model

The EM algorithm maximizes the data likelihood (26) by generating a sequence of iterates $\{\psi^{(i)}\}_{i=0}^{\infty}$ using the following two steps [51], [52]:

- *E-step:* Compute $Q(\psi \mid \psi^{(i-1)})$, defined as

$$\mathbb{E}_{y_{1:N}|X_{1:N},\psi^{(i-1)}}[\log f(X_{1:N}, y_{1:N} \,|\, \psi)] = \sum_{k=1}^{K} \sum_{n=1}^{N} \log(\pi_k f(X_n|\theta_k)) p(y_n = k \,|\, X_n, \psi^{(i-1)}).$$

- *M-step:* Find $\psi^{(i)} = \operatorname{argmax}_{\psi} Q(\psi \mid \psi^{(i-1)})$.

The expectation $Q(\psi^{(i)} \,|\psi^{(i-1)})$ increases after each EM iteration, and consequently converges to a (local) maximum of (26) [51], [52]. In practice, the iteration is terminated at a user defined number N_{iter} or when increments in $Q(\psi^{(i)} \,|\psi^{(i-1)})$ falls below a given threshold. The optimal cluster label estimate is the mode of the cluster label posterior (27).

Following the arguments from [55], the M-step can be accomplished by separately maximizing $Q(\pi_{1:K}, \theta_{1:K} \,|\psi^{(i-1)})$ over $\theta_1, \ldots, \theta_K$ and $\pi_{1:K}$. Using Lagrange multiplier with constraint $\sum_{k=1}^{K} \pi_k = 1$, yields the optimal weights:

$$\pi_k^{(i)} = \frac{1}{N} \sum_{n=1}^{N} p(y_n = k \,|\, X_n, \psi^{(i-1)}). \qquad (29)$$

Noting that $\log(\pi_k f(X_n \,|\, \theta_k))$ is accompanied by the weight $p(y_n = k \,|X_n, \psi^{(i-1)})$, *maximizing* $Q(\pi_{1:K}, \theta_{1:K} \,|\psi^{(i-1)})$ *over* θ_k *is equivalent to ML estimation of* θ_k *with weighted data*. However, the data-weighted MLE of θ_k depends on the specific form of $f(\cdot|\theta_k)$, and is intractable in general.

Fortunately, for the IID-cluster mixture model, where

$$f(X \,|\, \theta_k) = p_{\xi_k}(|X|) \, |X|! \, U^{|X|} p_{\varphi_k}^{X}, \qquad (30)$$

with $\theta_k = (\xi_k, \varphi_k)$ denoting the parameters of the cardinality and feature distributions, tractable solutions are available. Similar to Proposition 1, *the IID-cluster form allows the data-weighted MLE of* θ_k *to separate into data-weighted MLEs of* ξ_k *and* φ_k. Some examples are:

- For a categorical cardinality distribution with maximum cardinality M, where $\xi_k = (\xi_{k,0}, \ldots, \xi_{k,M})$ lies in the unit M-simplex, the iteration is

$$\xi_{k,m}^{(i)} = \frac{\sum_{n=1}^{N} \delta_m\left[|X_n|\right]\, p(y_n = k \,|\, X_n, \psi^{(i\text{-}1)})}{\sum_{\ell=0}^{M}\sum_{n=1}^{N} \delta_\ell\left[|X_n|\right]\, p\left(y_n = k \,|\, X_n, \psi^{(i\text{-}1)}\right)};$$

- For a Poisson cardinality distribution, where $\xi_k > 0$ is the mean cardinality, the iteration is

$$\xi_{k}^{(i)} = \frac{\sum_{n=1}^{N} |X_n|\, p(y_n = k \,|\, X_n, \psi^{(i\text{-}1)})}{\sum_{n=1}^{N} p\left(y_n = k \,|\, X_n, \psi^{(i\text{-}1)}\right)};$$

- For a Gaussian feature distribution, where $\varphi_k = (\mu_k, \Sigma_k)$ is the mean-covariance pair, the iteration is

$$\mu_k^{(i)} = \frac{\sum_{n=1}^{N} p(y_n = k \,|\, X_n, \psi^{(i\text{-}1)}) \sum_{x\in X_n} x}{\sum_{n=1}^{N} |X_n|\, p\left(y_n = k \,|\, X_n, \psi^{(i\text{-}1)}\right)},$$

$$\Sigma_k^{(i)} = \frac{\sum_{n=1}^{N} p(y_n = k \,|\, X_n, \psi^{(i\text{-}1)}) \sum_{x\in X_n} K_k^{(i)}(x)}{\sum_{n=1}^{N} |X_n|\, p(y_n = k \,|\, X_n, \psi^{(i\text{-}1)})},$$

  where $K_k^{(i)}(x) = (x - \mu_k^{(i)})(x - \mu_k^{(i)})^{\mathrm{T}}$;
- For a Gaussian mixture feature distribution, where φ_k is the Gaussian mixture parameter, $\varphi_k^{(i)}$ can be determined by applying the standard EM algorithm on the weighted data.

For I iterations, the complexity of the EM algorithm is $O(NKI)$. The EM algorithm generally suffers from slow convergence and the final solution depends on both the stopping criterion and the initial values. Thus a judicious choice of initial values is critical to the final cluster configuration. However, this is still an open problem. Numerical results for clustering of point patterns using the proposed EM algorithm are given in subsection 6.3.

5.2 Infinite Mixture Model

For an unknown number of clusters, finite mixture models are no longer directly applicable. Bayesian non-parametric modeling (see e.g., [56], [57]) addresses the unknown number of clusters by modeling the set of mixture parameters as a point process. Thus, the observations and the clusters are all modeled as point processes.

In a finite mixture model, the number of components (and clusters) is fixed at K. The mixture parameter $\psi = (\pi_{1:K}, \theta_{1:K})$ is a point in $(\mathbb{R}_+ \times \Theta)^K$, such that $\sum_i^K \pi_i = 1$. Under the Bayesian framework, it is further assumed that $\theta_{1:K}$ follows a given distribution on Θ^K, and that $\pi_{1:K}$ follows a distribution on the unit $(K-1)$-simplex, e.g. a Dirichlet distribution.

An infinite mixture model addresses the unknown number of components by considering the mixture parameter Ψ as a point pattern in $\mathbb{R}_+ \times \Theta$ such that $\sum_{(\pi,\theta)\in\Psi} \pi = 1$. Further, under the Bayesian non-parametric framework, we furnish Ψ with a prior distribution, thereby modeling the mixture parameter as a point process on $\mathbb{R}_+ \times \Theta$. The simplest model would be the Poisson point process, but the resulting component weights do not necessarily sum to one. Nonetheless, these weights can be normalized to yield a tractable point process model for the mixture parameter [58], [59]. More concisely, let Ξ be a Poisson point process on $\mathbb{R}_+ \times \Theta$ with intensity measure $\eta\omega^{-1}e^{-\eta\omega}\mathrm{d}\omega G_0(\mathrm{d}\theta)$, i.e., the product of an improper gamma distribution and the base distribution G_0. Then the prior model for the mixture parameter is given by

$$\Psi = \left\{\left(\nu_\Xi^{-1}\omega, \theta\right) : (\omega, \theta) \in \Xi\right\}, \tag{31}$$

where $\nu_\Xi = \sum_{(\omega,\theta)\in\Xi} \omega$. Note that (31) is no longer a Poisson point process because each constituent element involves the sum ν_Ξ, thereby violating the independence condition.

To specify an explicit form for the prior distribution of the mixture parameter Ψ, note that each point $\left(\nu_\Xi^{-1}\omega, \theta\right)$ can be equivalently represented by atom at θ with weight $\nu_\Xi^{-1}\omega$, and hence the point process (31) can be represented by the random atomic distribution G on Θ, defined by

$$G(A) = \nu_\Xi^{-1} \sum_{(\omega,\theta)\in\Xi} \omega\, \mathbf{1}_A(\theta). \tag{32}$$

It was shown in [58] that G follows a Dirichlet process $\mathrm{DP}(\eta, G_0)$, with parameter η and base distribution G_0. Noting that the cluster parameter θ_n for X_n can be regarded as a sample from G, the data generation process for this model can be summarized as follows

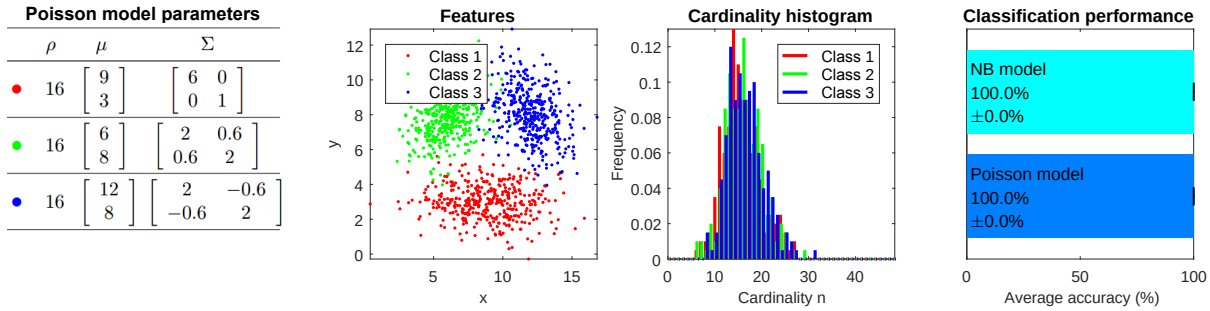
$$\begin{aligned} G &\sim \mathrm{DP}(\eta, G_0) \\ \theta_n &\sim G \\ X_n &\sim f(\cdot \mid \theta_n). \end{aligned}$$

The cluster assignment variables and the mixture parameters, including the number of clusters, can be automatically learned from the data via posterior inference. This Bayesian formulation can be adapted for semi-supervised learning, where only labeled training data for certain clusters are available and the objective is to compute the posterior of the missing labels. This approach can also address the novelty detection problem in Section 4 without having to rank the input observations.

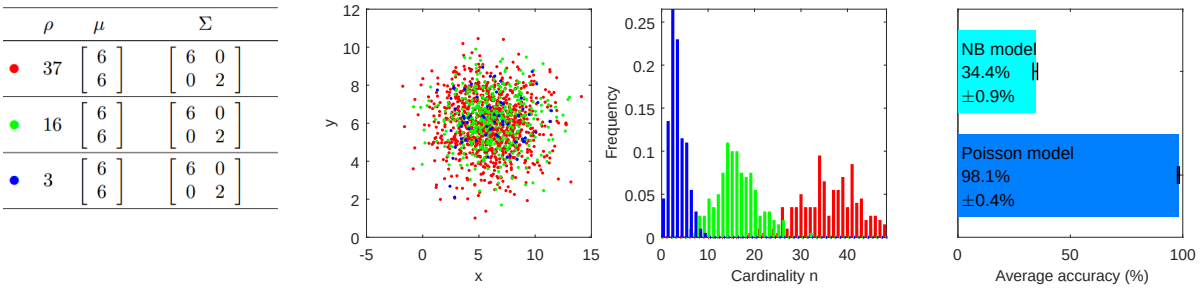
Computing the posterior for infinite point process mixture models is intractable in general, and development of tractable Bayesian inference algorithms is beyond the scope of this paper. Extension of the Gibbs sampler [53], [54] for standard mixture models to point pattern data is not straight forward since the predictive likelihood and consequently the conditionals are intractable in general. Even in the simplest case of Poisson point process mixture with Gaussian mixture intensities, such an extension not only involves determining the number of Poisson components and their weights, but also the number of Gaussians and their mixture parameters in the intensity of each Poisson components. Nonetheless, these are good starting points towards the development of tractable Bayesian clustering algorithms for point pattern data.

6 Experiments

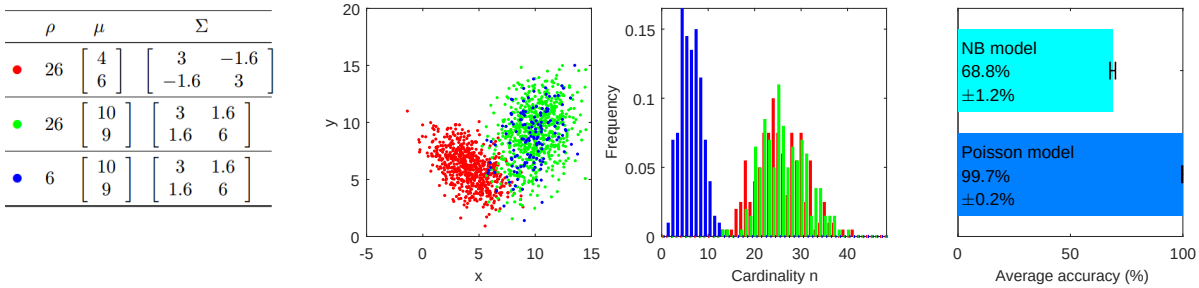
This section demonstrates the viability of the proposed framework using the Poisson model and IID-cluster model (with Categorical cardinality distribution). A Poisson model with Gaussian intensity is specified by the triple (ρ, μ, Σ) where ρ is the rate and μ, Σ are the mean and covariance of the feature density. The NB model is used as a performance benchmark since it has been used for this type of problems (see e.g. [4], [5], [6], [7], [12]) and assumes i.i.d. features.



(a) All 3 classes are well-separated from each other in feature, but overlapping in cardinality.



(b) All 3 classes are overlapping in feature, but well-separated from each other in cardinality.



(c) Classes 2, 3: overlap in feature, well-separated in cardinality. Classes 1, 2: overlap in cardinality, well-separated in feature.

Figure 7. Model, data and classification accuracy [60] (No. correct classifications / No. of observations in the test set) for three scenarios.

6.1 Classification Experiments

This subsection presents three classification experiments on simulated data, the Texture images dataset [61], and the StudentLife dataset [62]. In the training phase, ML is used to learn the parameters of the NB model and the Poisson model (as per subsection 3.1.4) from fully observed training data. Both trained models agree on the feature distribution. For simplicity we use a uniform class prior in the test phase. For the last dataset, we use the IID-cluster model instead of the Poisson model.

6.1.1 Classification on simulated data

We consider three diverse scenarios, each comprising three classes simulated from Poisson point processes with Gaussian intensities shown in Fig. 7. In scenario (a), point patterns from each class are well-separated from other classes in feature, but significantly overlapping in cardinality (see Fig. 7a). In scenario (b), point patterns from each class are well-separated from other classes in cardinality, but significantly overlapping in feature (see Fig. 7b). Scenario (c) is a mix of (a) and (b), where: point patterns from Class 1 are well-separated from other classes in features, but significantly overlapping with Class 2 in cardinality; and the point patterns from Classes 2 and 3 significantly overlap in feature, but well-separated in cardinality (see Fig. 7c).

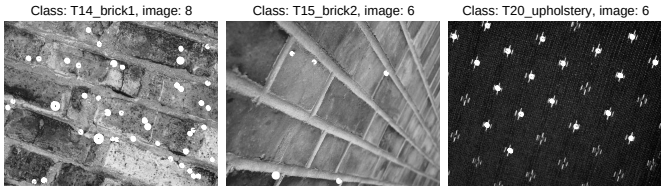
The fully observed training dataset comprises 600 point patterns (200 per class) is used to train the NB/Poisson model in which each class is modeled by a Gaussian density/intensity. In the test phase, 10 different test sets each comprises 300 point patterns (100 per class) are used. The average classification performance is reported in Fig. 7.

In scenario (a) both models achieve perfect classification using only the features of the test point patterns because the classes are so well-separated in the feature space. In scenario (b) on the other hand, neither models are able to differentiate the classes using the features in the test data. Nonetheless, the Poisson model achieved excellent performance by exploiting the separation in cardinality of the classes from the test data. In contrast, the NB model's inability to exploit cardinality information results in very poor performance.

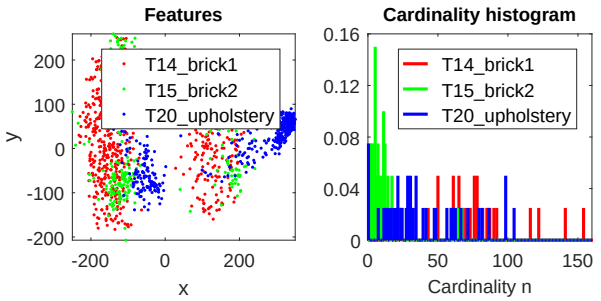
In scenario (c) both models can differentiate Class 1 from 2 and 3 by exploiting the separation in features. However, the Poisson model achieved near perfect performance by further exploiting the well-separated cardinality to differentiate Class 3 from 1 and 2, whereas NB could not.

6.1.2 Classification on the Texture dataset

Three classes "T14 brick1", "T15 brick2", and "T20 upholstery" of the Texture images dataset [61] are considered. Each class comprises 40 images, with some examples shown in Fig. 8a. Each image is processed by the SIFT algorithm (using the VLFeat library [63]) to produce a point pattern of 128-D SIFT features, which is then compressed into a 2-D point pattern by Principal Component Analysis (PCA). Fig. 8b shows the superposition of the 2-D point patterns from the three classes along with their cardinality histograms.



(a) Example images (circles represent detected SIFT keypoints).



(b) Extracted 2-D point patterns.

Figure 8. Three classes of the Texture dataset.

A 4-fold cross validation scheme is used for performance evaluation. In each fold, the fully observed training dataset comprising 30 images per class is used to learn the NB/Poisson model in which each class is parameterized by a 3-component Gaussian mixture density/intensity. The test set comprises the remaining images (10 per class).

Fig. 8b shows that the classes are neither well-separated in feature nor cardinality. Note also that there are possible dependencies between the features of the point patterns not captured by the simple Poisson model. However, the Poisson model still shows good performance (even on a small training set), and outperforms NB, as shown in Fig. 9, by exploiting cardinality information.

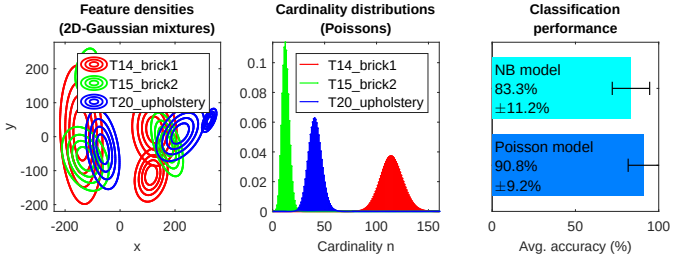


Figure 9. MLE of model parameters and classification performance on the Texture dataset (standard deviations are indicated by the error-bars). The feature densities are the same for both Poisson and NB models.

6.1.3 Classification on the StudentLife dataset

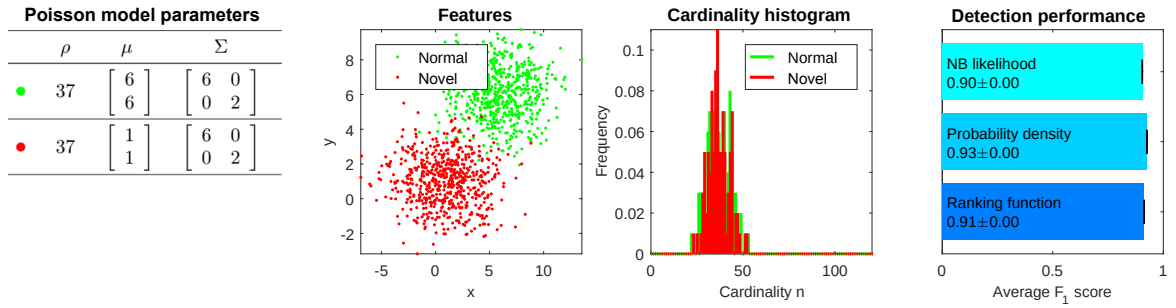
To demonstrate the scalability of the proposed solutions, we choose the StudentLife dataset [62] that is widely-used in pervasive computing research. This dataset contains various data types (e.g. Wi-Fi signals, Bluetooth scan) collected from the smartphones of 49 voluntary students at Dartmouth College over a 10-week term in 2013. Pre-processing of the data is described in [64]. For the purpose of our experiments, we only use the Wi-Fi signal strength readings, which are grouped into 10-minute time frames, based on their time-stamps. If there are multiple readings of a Wi-Fi ID within a 10-minute frame, we use the mean signal strength as its (sole) observation. Only10-minute frames with at least 1 observation of any Wi-Fi ID after aggregation are retained. The resulting dataset is a collection of records, each of which is a 1271-D vector corresponding to readings of the 1271 Wi-Fi IDs in 10-minute intervals, and is compatible with the benchmark NB-based classifier and K-means clustering algorithm. Each point pattern observation is obtained by retaining only the non-zero entries of each 1271-D vector (hence, the cardinality of this point pattern is the number of non-zero entries of the vector). An element of the converted point pattern is an ordered pair of Wi-Fi ID and its signal strength. For the StudentLife dataset classification and clustering experiments, we use an IID-cluster model with Categorical cardinality distribution and feature density consisting of a Categorical distribution for the Wi-Fi ID and a 3-component 1-D Gaussian mixture for the corresponding signal strength.

No. days	No. PPs	No. feat.	Min, Mode, Max	NB (%)	IC (%)
1	2,132	10,002	1, 2, 39	**77.0±3.5**	76.4±2.4
2	4,449	24,141	1, 2, 56	73.3±2.0	**78.4±2.4**
3	6,796	36,297	1, 2, 49	66.3±1.9	**77.5±1.7**
4	8,833	46,514	1, 2, 56	67.4±1.0	**79.4±2.0**
5	10,396	53,804	1, 2, 56	65.7±1.0	**79.7±1.4**
6	12,718	64,334	1, 2, 56	66.2±1.0	**79.3±1.0**
7	15,034	76,060	1, 2, 56	67.9±1.1	**81.0±0.7**
14	30,231	152,203	1, 2, 56	63.3±0.9	**80.8±0.3**
21	46,219	221,995	1, 2, 57	63.0±0.9	**79.8±0.7**
28	61,945	295,863	1, 2, 57	61.8±0.9	**78.7±0.6**

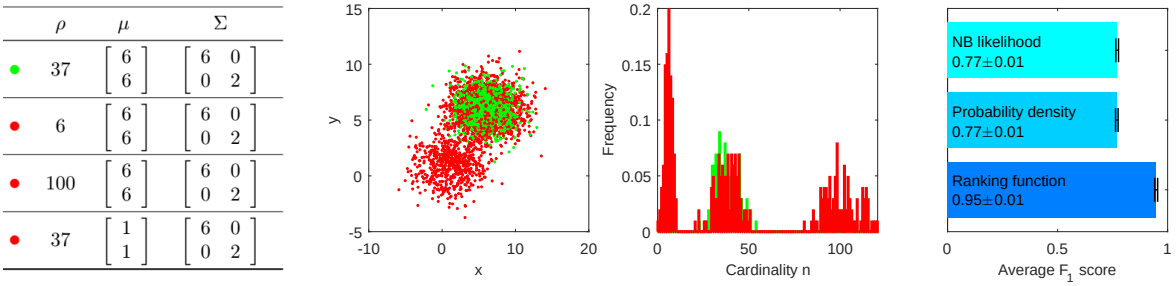
Table 1
Statistics for the10 subdatasets constructed from StudentLife dataset, and classification accuracies for NB and IID-cluster (IC) models.

In our classification experiment, we construct (from the full StudentLife dataset) 10 subdatasets, with respective total observation periods of 1 day, 2 days, ... , 7 days, 2 weeks, 3 weeks, and 4 weeks. Further, for each subdataset, we select only the top 20 users with the most number of non-empty observations. The total numbers of point patterns and features, the minimum, mode, and maximum of the point pattern cardinalities for each subdataset are shown in Table 1. The user IDs are used as ground-truth classification labels, hence we have 20 classes in each classification task. In each task, we employ a 10-fold cross validation scheme.

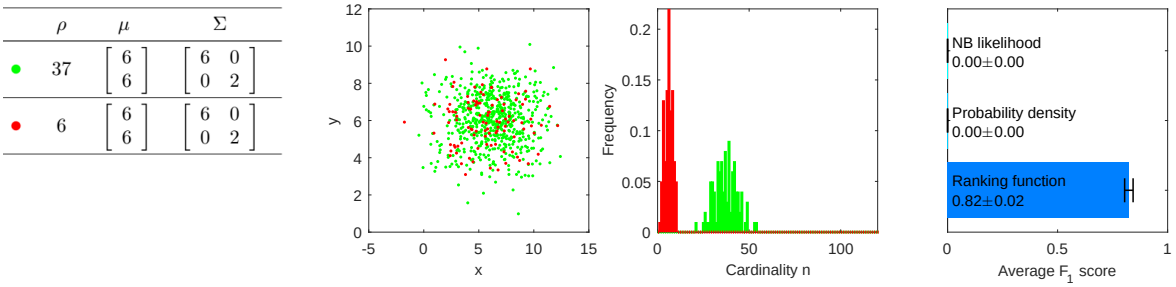
The average accuracies of the NB and IID-cluster models are reported respectively in the last 2 columns of Table 1. Except for the first subdataset, the proposed classifier outperforms its benchmark by a large margin. Observe the overall trend that as we have more observations, the accuracy of IID-cluster model tends to increase whilst the accuracy of NB model tends to decrease.



(a) Novelty well-separated from 'normal' data in feature, but overlap in cardinality.



(b) Novelty and 'normal' data partially overlap in feature and cardinality.



(c) Low-cardinality novelty that overlap with 'normal' data in feature.

Figure 10. Model, data and novelty detection performance for three scenarios.

6.2 Novelty Detection Experiments

This subsection presents two novelty detection experiments on simulated and real data using the Poisson model to illustrate the effectiveness of the proposed ranking function against the NB likelihood and standard probability density (classification on the StudentLife dataset is sufficient to demonstrate scalability since the proposed novelty detection and classification use the same ML algorithm). Like the classification experiments, ML is used to learn the parameters of the 'normal' NB and Poisson models in the training phase. The novelty threshold is set at the 2nd 10-quantile of the ranking values of the 'normal' training data. The detection performance measure is the F_1 *score* [60]:

$$F_1(\text{precision, recall}) \triangleq 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}},$$

where *precision* is the proportion of correct decisions in the output of the detector, and *recall* is the proportion of correctly identified novelties in the test set. To ensure functional continuity of F_1, we define $F_1(0,0) \triangleq 0$.

6.2.1 Novelty detection on simulated data

We consider three simulated scenarios comprising 'normal' and novel point patterns generated from Poisson point processes with 2-D Gaussian intensities as shown in Fig. 10. All scenarios have the same 'normal' point patterns, with cardinalities between 20 and 60. In scenario (a) novelties are well-separated from 'normal' data in feature, but overlapping in cardinality (see Fig. 10a). In scenario (b) novelties are overlapping with 'normal' data in feature, but only partially overlapping in cardinality (see Fig. 10b). In scenario (c) we remove the high cardinality novelties from (b) (see Fig. 10c).

In the training phase, the same 300 'normal' point patterns for each scenario are used to learn the 'normal' NB/Poisson model that consists of a Gaussian density/intensity. In the testing phase, 10 tests are ran per scenario with each test set comprising 100 'normal' point patterns and 100 novelties generated according to their respective models.

Observe from Fig. 10a that in scenario (a) the NB likelihood, probability density, and ranking function all perform well. Even though the NB likelihood and probability density are not consistent in ranking, the good separation in features of 'novel' from 'normal' test data is sufficient to differentiate them. The box plots in Fig. 11a shows that the range of ranking values for 'normal' data (for all three functions) are well-separated from 'novel' data, and hence the good detection performance.

Fig. 10b shows that the proposed ranking function out performs the others in scenario (b). The performance of the NB likelihood and probability density are actually inflated by erroneously ranking all high cardinality point patterns lower than they should be due to the multiplication of many small numbers, which inadvertently include some novelties. The box plots in Fig. 11b show that the ranges of NB likelihood and probability density values for 'normal' data fall within those for 'novel' data, making them difficult to differentiate. On the other hand the range of ranking function values for 'normal' data sits above that for 'novel' data, which allows them to be differentiated.

In scenario (c), where the high cardinality novelties are removed from the training and test sets, Fig. 10c shows that only the ranking function performed well while the others completely failed. The reason for such failure (apart from failing to detect low cardinality novelties) is that there are no high cardinality novelties for NB likelihood and probability density to inadvertently detect this time. The boxplots in Fig. 11c shows that the NB likelihood and the probability density even rank novelties much higher than 'normal' data. Only the proposed ranking function is consistent in all three scenarios.

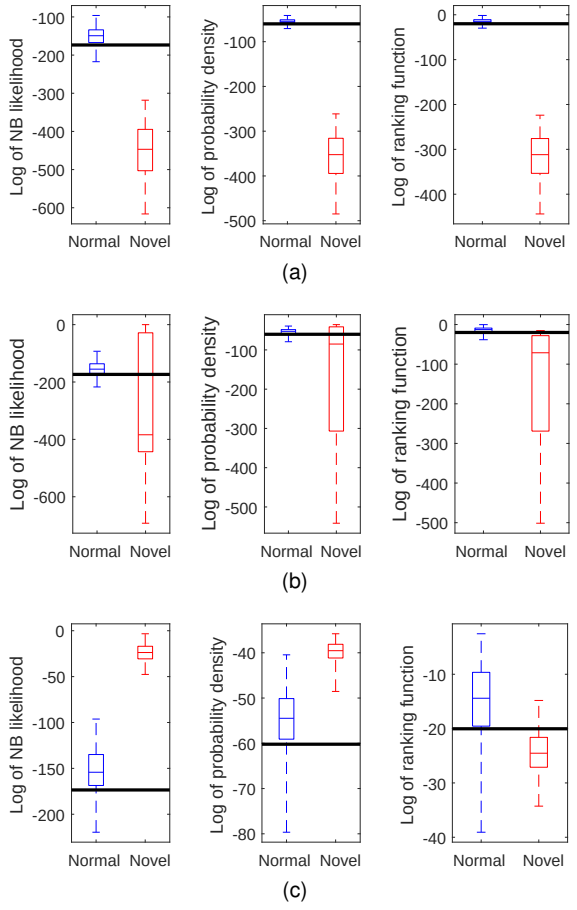


Figure 11. Boxplots of: NB likelihood, probability density, and ranking function for the test data in the three simulated scenarios in Fig. 10 (solid line through each graph indicates the novelty threshold chosen from training data).

6.2.2 *Novelty detection on the Texture dataset*

For this experiment, data from class "T14 brick1" of the Texture dataset from subsection 6.1.2, are considered 'normal' while novel data are taken from class "T20 upholstery". A 4-fold cross validation scheme is used for performance evaluation. In each fold, training data comprising 30 'normal' images is used to learn the 'normal' NB/Poisson model that consists of a 3-component Gaussian mixture density. The test set comprises the remaining 10 'normal' and 10 novel images. The learned models are similar to those of class "T14 brick1" in Fig. 9.

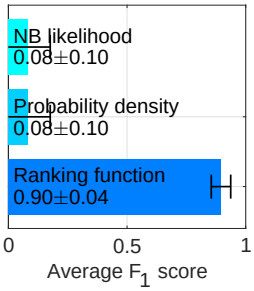


Figure 12. Averaged novelty detection performance on the Texture dataset for: NB likelihood, probability density, and proposed ranking function.

Fig. 12 showed that ranking the data using the NB likelihood or the probability density failed to detect most novelties, whereas the proposed ranking function achieved a high F_1 score. The poor performance can be attributed to the fact (established in subsection 4.1) that the NB likelihood and probability density do not indicate how probable or likely a point pattern is. This inconsistency is illustrated by the box plots in Fig. 13. Note that even with hindsight it is not possible to separate 'novel' from 'normal' data using their NB likelihood and probability values. On the other hand, the box plots verified that the proposed ranking function provides a consistent ranking.

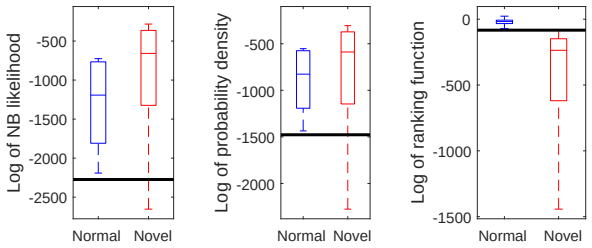
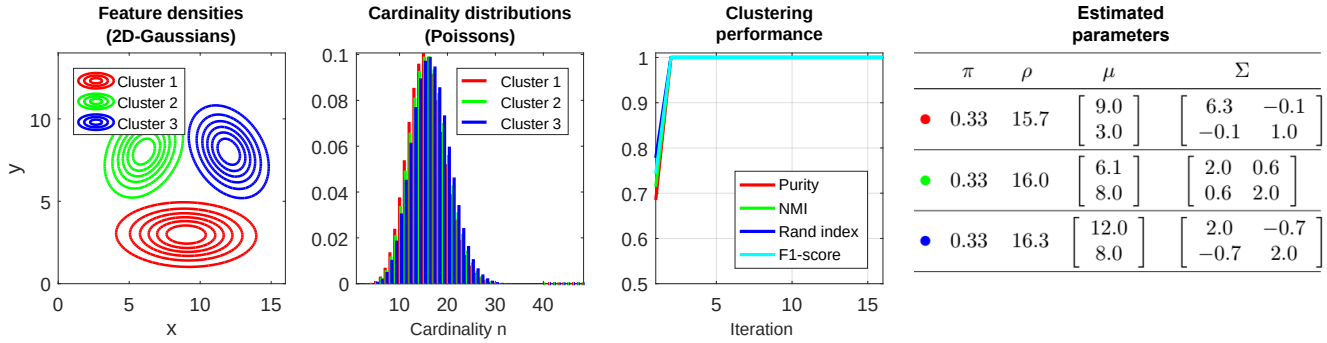


Figure 13. Boxplots of: NB likelihood; probability density, and ranking function; for 'normal' and novel test data in one fold of the Texture dataset.

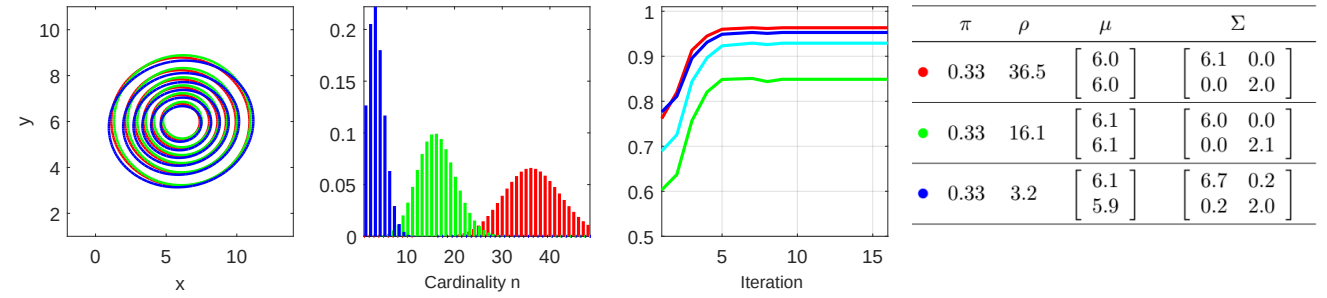
6.3 Clustering Experiments

To investigate the characteristics of the Poisson mixture model, subsections 6.3.1 and 6.3.2 present two clustering experiments on simulated data and the Texture images dataset [61], respectively, with known number of clusters using the EM clustering algorithm (outlined in subsection 5.1.1). Further, for a larger scale demonstration we benchmark the IID-cluster mixture model against the K-Means algorithm on the StudentLife dataset [62] in subsection 6.3.3. For clustering performance measure, we use purity, normalized mutual information (NMI), Rand index (RI) and F_1 score.



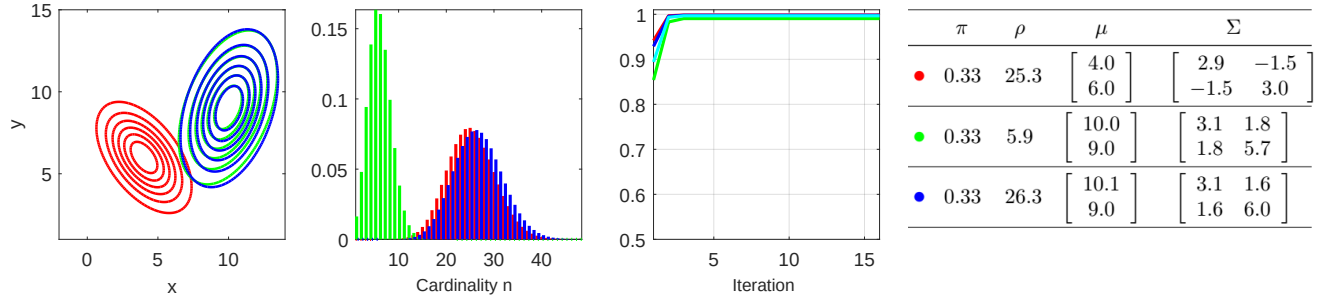
	π	ρ	μ	Σ
●	0.33	15.7	$\begin{bmatrix} 9.0 \\ 3.0 \end{bmatrix}$	$\begin{bmatrix} 6.3 & -0.1 \\ -0.1 & 1.0 \end{bmatrix}$
●	0.33	16.0	$\begin{bmatrix} 6.1 \\ 8.0 \end{bmatrix}$	$\begin{bmatrix} 2.0 & 0.6 \\ 0.6 & 2.0 \end{bmatrix}$
●	0.33	16.3	$\begin{bmatrix} 12.0 \\ 8.0 \end{bmatrix}$	$\begin{bmatrix} 2.0 & -0.7 \\ -0.7 & 2.0 \end{bmatrix}$

(a) All 3 clusters are well-separated from each other in feature, but overlapping in cardinality.



	π	ρ	μ	Σ
●	0.33	36.5	$\begin{bmatrix} 6.0 \\ 6.0 \end{bmatrix}$	$\begin{bmatrix} 6.1 & 0.0 \\ 0.0 & 2.0 \end{bmatrix}$
●	0.33	16.1	$\begin{bmatrix} 6.1 \\ 6.1 \end{bmatrix}$	$\begin{bmatrix} 6.0 & 0.0 \\ 0.0 & 2.1 \end{bmatrix}$
●	0.33	3.2	$\begin{bmatrix} 6.1 \\ 5.9 \end{bmatrix}$	$\begin{bmatrix} 6.7 & 0.2 \\ 0.2 & 2.0 \end{bmatrix}$

(b) All 3 clusters are overlapping in feature, but well-separated from each other in cardinality.



	π	ρ	μ	Σ
●	0.33	25.3	$\begin{bmatrix} 4.0 \\ 6.0 \end{bmatrix}$	$\begin{bmatrix} 2.9 & -1.5 \\ -1.5 & 3.0 \end{bmatrix}$
●	0.33	5.9	$\begin{bmatrix} 10.0 \\ 9.0 \end{bmatrix}$	$\begin{bmatrix} 3.1 & 1.8 \\ 1.8 & 5.7 \end{bmatrix}$
●	0.33	26.3	$\begin{bmatrix} 10.1 \\ 9.0 \end{bmatrix}$	$\begin{bmatrix} 3.1 & 1.6 \\ 1.6 & 6.0 \end{bmatrix}$

(c) Clusters 2 and 3: overlap in feature, well-separated in cardinality. Clusters 1 and 2: overlap in cardinality, well-separated in feature.

Figure 14. EM clustering performance on the three simulated data scenarios in Fig. 7.

6.3.1 *EM clustering on simulated data*

This experiment uses the same simulated dataset described in subsection 6.1.1 but without labels. Since there are three clusters, we use a 3-component Poisson mixture model, where each constituent Poisson point process is parameterized by a Gaussian intensity.

Fig. 14 show that the proposed point pattern clustering algorithm performs well on all three scenarios. In scenario (a) the estimated mixture model parameters are very close to the ground truth and perfect clustering is achieved because the clusters are so well-separated in the feature space.

In scenario (b) clustering performance is very good, using only the separation of the clusters in cardinality. While the estimated mixture model parameters are very close to the ground truth, the clustering performance is not perfect because the constituent Poisson cardinality distributions significantly overlap with each other. On the other hand, in scenario (a) the constituent Gaussians have negligible overlap, resulting in perfect clustering performance.

In scenario (c) the estimated mixture model parameters are very close to the ground truth. Despite partial overlaps in features and cardinality, the proposed algorithm was able to exploit the separation in features to differentiate Cluster 1 from 2 and 3, as well as the separation in cardinality to differentiate Cluster 3 from 1 and 2, to achieve near perfect clustering performance.

6.3.2 *EM clustering on the Texture dataset*

This experiment uses the Texture dataset of subsection 6.1.2, but without labels. Since there are three clusters, we use a 3-component Poisson mixture model, where each constituent Poisson point process is parameterized by a 3-component Gaussian mixture intensity (similar to subsection 6.1.2). The M-step of the proposed EM algorithm is accomplished by applying standard EM to find the data-weighted MLE of the Gaussian mixture parameter.

Fig. 15 shows that the proposed EM algorithm achieved good clustering performance. While the Gaussian mixture intensity function captures the multi-modality of the features, the Poisson model itself cannot capture cardinalities other than Poisson, nor interactions between the features (which is present in the Texture data, as different texture patterns are characterized by different spacing between

the points). Nonetheless, despite very limited degrees of freedom, the Poisson model still shows encouraging clustering performance. The limitations of the Poisson model can be alleviated by the more sophisticated Gibbs model (subsection 2.5). However, clustering techniques for the Gibbs model are yet to be developed, not to mention the higher computational cost. The next experiment uses the IID-cluster model to provide more degrees of freedom.

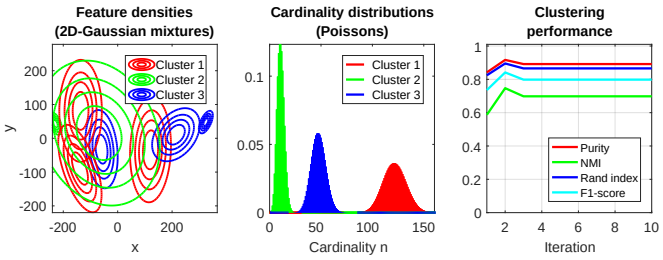


Figure 15. Clustering performance for Poisson MM on the Texture dataset.

6.3.3 EM clustering on the StudentLife dataset

This experiment uses the dataset and the vector-to-point-pattern conversion process described in subsection 6.1.3. From the full StudentLife dataset, we construct a subdataset with a total observation period of 72 hours. We further truncate this subdataset by selecting only the top 20 users with the most number of non-empty observations. In total, this yields 6,796 point patterns with 36,297 features. In terms of cardinality, the largest point pattern has 49 elements whilst the smallest ones have 1 element. The user IDs are used as the ground-truth clustering labels, giving a total of 20 clusters.

The point pattern observations are fed to the proposed EM algorithm with an IID-cluster mixture model (IC MM) of 20 components, each of which is an IID-cluster (described in subsection 6.1.3). The Categorical cardinality distribution in this model offers additional degrees of freedom over the Poisson point process model, which allows it to capture better cardinality information in the data. We choose K-Means, arguably one of the best clustering algorithms available, as the benchmark method and feed it with the pre-converted (vector-valued) observations. Using a 10-fold validation scheme, each algorithm was run 10 times and the average performance indices are reported in Table 2. This table shows that the IID-cluster mixture model yields comparable results to K-Means in purity and NMI. However, the IID-cluster mixture model outperforms K-Means by a large margin in RI, adjusted RI, and F_1 score.

	Purity	NMI	RI	Adj. RI	$\mathbf{F_1}$
K-Means	0.37±0.05	0.45±0.04	0.73±0.05	0.10±0.02	0.17±0.02
IC MM	0.37±0.03	**0.47±0.02**	**0.89±0.02**	**0.26±0.03**	**0.32±0.03**

Table 2
Clustering performance for K-Means and IID-cluster MM (IC MM) on the StudentLife dataset.

7 Conclusions

This article outlined a model-based learning framework for point pattern data using point process theory, with a view to facilitate application to multiple instance learning. We demonstrated how the proposed framework enables the extensions of various model-based learning tasks to accommodate point pattern data in a conceptually transparent yet principled manner. A salient and consequential observation (in a statistical learning context) is that, contrary to common interpretation, the probability density of a point pattern does not necessarily indicate how likely or probable it is. We also developed algorithms, based on simple point process models, for classification, novelty detection, and clustering, which demonstrated impressive performance on a series of experiments. For tractability, these solutions assumed independence between the points of the point patterns, and thus unable to capture the intra-point-pattern correlations observed in most applications.

While our study only touches on some aspects of multiple instance learning, we hope it paves the way for exciting new research. Efficient techniques for learning models that can capture intra-point-pattern interactions such as Gibbs will provide a universal toolset for analysing most types of point pattern data. This is also an active research area in stochastic geometry, which will open the doors to many applications in data science. Another exciting venue for further research is Bayesian point pattern clustering, starting with our discussion on the computation of the posterior infinite point process mixture model. The proposed framework is flexible enough to accommodate other learning tasks. Model-based treatments of regression, density estimation, and dimensionality reduction for point pattern data are important areas of investigation, with novel results and algorithms waiting to be discovered.

References

[1] J. Amores, "Multiple instance classification: Review, taxonomy and comparative study," *Artificial Intelligence*, vol. 201, pp. 81–105, 2013.
[2] V. Cheplygina, D. M. Tax, and M. Loog, "On classification with bags, groups and sets," *Pattern Recognition Letters*, vol. 59, pp. 11–17, 2015.
[3] V. Cheplygina, D. M. Tax, and M. Loog, "Multiple instance learning with bag dissimilarities," *Pattern Recognition*, vol. 48, no. 1, pp. 264–275, 2015.
[4] M. E. Maron, "Automatic indexing: an experimental inquiry," *JACM*, vol. 8, no. 3, pp. 404–417, 1961.
[5] T. Joachims, "A probabilistic analysis of the rocchio algorithm with tfidf for text categorization.," tech. rep., DTIC Document, 1996.
[6] A. McCallum and K. Nigam, "A comparison of event models for naive Bayes text classification," in *AAAI-98 Workshop learning for text categorization*, vol. 752, pp. 41–48, 1998.
[7] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *Workshop statistical learning in computer vision, ECCV*, 2004.
[8] B. Ramesh, C. Xiang, and T. H. Lee, "Shape classification using invariant features and contextual information in the bag-of-words model," *Pattern Recognition*, vol. 48, no. 3, pp. 894–906, 2015.
[9] K. Zagoris, I. Pratikakis, A. Antonacopoulos, B. Gatos, and N. Papamarkos, "Distinction between handwritten and machine-printed text based on the bag of visual words model," *Pattern Recognition*, vol. 47, no. 3, pp. 1051–1062, 2014.
[10] D. M. Chickering and D. Heckerman, "Fast learning from sparse data," in *Proc. 15th Conf. Uncertainty in artificial intelligence*, pp. 109–115, Morgan Kaufmann Publishers Inc., 1999.

[11] L. Jing, M. K. Ng, and J. Z. Huang, "An entropy weighting k-means algorithm for subspace clustering of high-dimensional sparse data," *IEEE Trans. Knowl. Data Eng.*, vol. 19, no. 8, pp. 1026–1041, 2007.

[12] I. V. Cadez, S. Gaffney, and P. Smyth, "A general probabilistic framework for clustering individuals and objects," in *Proc. 6th ACM SIGKDD Int. Conf. knowledge discovery and data mining*, pp. 140–149, 2000.

[13] M. Markou and S. Singh, "Novelty detection: a review – part 1: statistical approaches," *Signal Process.*, vol. 83, no. 12, pp. 2481–2497, 2003.

[14] D. J. Daley and D. Vere-Jones, *An introduction to the theory of point processes*, vol. 2. Springer, 1988.

[15] D. Stoyan, W. S. Kendall, and J. Mecke, *Stochastic geometry and its applications*. John Wiley & Sons, 1995.

[16] J. Moller and R. P. Waagepetersen, *Statistical inference and simulation for spatial point processes*. CRC Press, 2003.

[17] B.-N. Vo, Q. N. Tran, D. Phung, and B.-T. Vo, "Model-based classification and novelty detection for point pattern data," in *23rd Intl. Conf. Pattern Recognition (ICPR)*, Dec. 2016.

[18] Q. N. Tran, B.-N. Vo, D. Phung, and B.-T. Vo, "Clustering for point pattern data," in *23rd Int. Conf. Pattern Recognition (ICPR)*, Dec 2016.

[19] M. van Lieshout, *Markov Point Processes and their Applications*. Imperial College Press, 2000.

[20] A. Baddeley, I. Bárány, and R. Schneider, "Spatial point processes and their applications," *Stochastic Geometry: Lectures given at the CIME Summer School held in Martina Franca, Italy, September 13–18, 2004*, pp. 1–75, 2007.

[21] C. J. Geyer *et al.*, "Likelihood inference for spatial point processes," *Stochastic geometry: likelihood and computation*, vol. 80, pp. 79–140, 1999.

[22] B.-N. Vo, S. Singh, and A. Doucet, "Sequential monte carlo methods for multitarget filtering with random finite sets," *Aerosp. Electron. Syst., IEEE Trans.*, vol. 41, no. 4, pp. 1224–1245, 2005.

[23] R. Mahler, "Multi-target Bayes filtering via first-order multi-target moments," *IEEE Trans. Aerospace & Electronic Systems*, vol. 39, no. 4, pp. 1152–1178, 2003.

[24] R. P. Mahler, *Statistical multisource-multitarget information fusion*. Artech House, Inc., 2007.

[25] C. M. Bishop, *Pattern recognition and machine learning*. Springer, 2006.

[26] K. Murphy, *Machine learning: a probabilistic perspective*. MIT press, 2012.

[27] J. Hamidzadeh, R. Monsefi, and H. S. Yazdi, "Irahc: instance reduction algorithm using hyperrectangle clustering," *Pattern Recognition*, vol. 48, no. 5, pp. 1878–1889, 2015.

[28] V. H. Moghaddam and J. Hamidzadeh, "New hermite orthogonal polynomial kernel and combined kernels in support vector machine classifier," *Pattern Recognition*, vol. 60, pp. 921–935, 2016.

[29] D. P. Huttenlocher, J. J. Noh, and W. J. Rucklidge, "Tracking non-rigid objects in complex scenes," in *Proc. 4th Int. Conf. Comput. Vision, 1993*, pp. 93–101, IEEE, 1993.

[30] R. T. Rockafellar and R. J.-B. Wets, *Variational analysis*. Springer Sci. & Business Media, 2009.

[31] D. M. Gavrila and V. Philomin, "Real-time object detection for "smart" vehicles," in *Proc. 7th Int. Conf. Comput. Vision, 1999*, vol. 1, pp. 87–93, 1999.

[32] J. Zhang, M. Marszałek, S. Lazebnik, and C. Schmid, "Local features and kernels for classification of texture and object categories: A comprehensive study," *Int. J. Comput. Vision*, vol. 73, no. 2, pp. 213–238, 2007.

[33] Y. Rubner, C. Tomasi, and L. J. Guibas, "A metric for distributions with applications to image databases," in *6th IEEE Int. Conf. Comput. Vision*, pp. 59–66, 1998.

[34] Y. Ogata and M. Tanemura, "Likelihood analysis of spatial point patterns," *J. Royal Statistical Society. Series B (Methodological)*, pp. 496–518, 1984.

[35] R. Takacs, "Estimator for the pair–potential of a gibbsian point process," *Statistics: A J. Theoretical and Applied Statistics*, vol. 17, no. 3, pp. 429–433, 1986.

[36] T. Fiksel, "Estimation of interaction potentials of gibbsian point processes," *Statistics*, vol. 19, no. 1, pp. 77–86, 1988.

[37] J. Besag, "Statistical analysis of non-lattice data," *The statistician*, pp. 179–195, 1975.

[38] J. Besag, "Some methods of statistical analysis for spatial data," *Bulletin of the Int. Statistical Institute*, vol. 47, no. 2, pp. 77–92, 1977.

[39] A. Baddeley and R. Turner, "Practical maximum pseudolikelihood for spatial point patterns," *Australian & New Zealand J. Stats.*, vol. 42, no. 3, pp. 283–322, 2000.

[40] J. L. Jensen and J. Møller, "Pseudolikelihood for exponential family models of spatial point processes," *Annals of Applied Probability*, pp. 445–461, 1991.

[41] M. Filippone and G. Sanguinetti, "Information theoretic novelty detection," *Pattern Recognition*, vol. 43, no. 3, pp. 805–814, 2010.

[42] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surveys (CSUR)*, vol. 41, no. 3, p. 15, 2009.

[43] V. J. Hodge and J. Austin, "A survey of outlier detection methodologies," *Artificial Intelligence Review*, vol. 22, no. 2, pp. 85–126, 2004.

[44] G. Peters, "Is there any need for rough clustering?," *Pattern Recognition Letters*, vol. 53, pp. 31–37, 2015.

[45] M. Filippone, F. Camastra, F. Masulli, and S. Rovetta, "A survey of kernel and spectral methods for clustering," *Pattern recognition*, vol. 41, no. 1, pp. 176–190, 2008.

[46] S. Russell and P. Norvig, *Artificial Intelligence: A modern approach*. Prentice Hall, 2003.

[47] A. K. Jain, "Data clustering: 50 years beyond k-means," *Pattern recognition letters*, vol. 31, no. 8, pp. 651–666, 2010.

[48] R. Xu and D. Wunsch, "Survey of clustering algorithms," *IEEE Trans. Neural Networks*, vol. 16, no. 3, pp. 645–678, 2005.

[49] M.-L. Zhang and Z.-H. Zhou, "Multi-instance clustering with applications to multi-instance prediction," *Appl. Intell.*, vol. 31, no. 1, pp. 47–68, 2009.

[50] D. Zhang, F. Wang, L. Si, and T. Li, "M3ic: Maximum margin multiple instance clustering," in *IJCAI*, vol. 9, pp. 1339–1344, 2009.

[51] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *J. Royal Statistical Soc. Series B (Methodological)*, pp. 1–38, 1977.

[52] R. J. Little and D. B. Rubin, *Statistical analysis with missing data*. John Wiley & Sons, 2002.

[53] J. M. Bernardo and A. F. Smith, *Bayesian theory*, vol. 405. John Wiley & Sons, 2009.

[54] A. Gelman, J. Carlin, H. Stern, and D. Rubin, *Bayesian Data Analysis*. Chapman & Hall/CRC, 2003.

[55] J. A. Bilmes, "A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models," *Int. Comput. Sci. Institute*, vol. 4, no. 510, 1998.

[56] J. Ghosh and R. Ramamoorthi, *Bayesian Nonparametrics*. Springer Verlag, 2003.

[57] N. Hjort, C. Holmes, P. Müller, and S. Walker, *Bayesian nonparametrics*. Cambridge Univ. Press, 2010.

[58] D. Lin, E. Grimson, and J. Fisher, "Construction of dependent dirichlet processes based on poisson processes," *Advances in Neural Information Processing Systems*, 2010.

[59] M. Jordan, "Hierarchical models, nested models and completely random measures," in *Frontiers of Statistical Decision Making and Bayesian Analysis: In Honor of James O. Berger*, Springer-Verlag, New York, NY, 2010.

[60] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to information retrieval*, vol. 1. Cambridge univ. press Cambridge, 2008.

[61] S. Lazebnik, C. Schmid, and J. Ponce, "A sparse texture representation using local affine regions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1265–1278, 2005.

[62] R. Wang, F. Chen, Z. Chen, T. Li, G. Harari, S. Tignor, X. Zhou, D. Ben-Zeev, and A. T. Campbell, "Studentlife: assessing mental health, academic performance and behavioral trends of college students using smartphones," in *Proc. Intl. J. Conf. Pervasive & Ubiq. Comp.*, pp. 3–14, ACM, 2014.

[63] A. Vedaldi and B. Fulkerson, "Vlfeat: An open and portable library of comput. vision algorithms." http://www.vlfeat.org/, 2008.

[64] T. Nguyen, V. Nguyen, S. Venkatesh, and D. Q. Phung, "MCNC: multi-channel nonparametric clustering from heterogeneous data," in *23rd Intl. Conf. Pattern Recognition, ICPR 2016, Cancún, Mexico, December 4-8, 2016*, pp. 3633–3638, 2016.