Chapter 1

Methods based on spatial processes

Lisa Handl, Christian Hirsch and Volker Schmidt

Institute of Stochastics, Ulm University, 89069 Ulm, Germany

Abstract

We present methods of cluster analysis to identify and classify patterns which are formed by stationary point processes and used e.g. in the modeling of human cartilage cells. In particular, we consider a generalization of the single-linkage approach which has been appropriately adapted to deal with inhomogeneous data and describe how the random-forest methodology can be applied to the classification of the previously identified clusters. Furthermore, we show how statistical techniques for spatial point processes can be used to validate the methods of cluster identification and cluster classification considered in this chapter.

1.1 Introduction

In this chapter we present methods of cluster analysis to identify and classify patterns which are formed by stationary point processes and used e.g. in the modeling of human cartilage cells. In particular, we consider a generalization of the single-linkage approach adapted to deal with inhomogeneous data and describe how the random-forest methodology can be applied to the classification of the previously identified clusters. Note that the problem of clustering point patterns occurs not only at microscopic scales in fields like biology, physics, and computational materials science, but also in geographic applications at macroscopic scales. Often point-process models play an important role in these applications and the development of special cluster algorithms adapted to the problem at hand is imperative.

Let us first discuss a macroscopical example which builds on a point-process model developed in [3]. As explained in [4] due to recent technological advances it is now possible to cost-efficiently produce small sensors which can be used to gather information on their environment. Due to their small size such sensors can carry only very limited battery capacities and devising energy-efficient computation schemes is crucial. As the cost of transmitting information over large distances is usually higher than the cost of computation itself, it makes sense to group these sensors into hierarchical clusters so that only one node in each cluster has to transmit the information to the next hierarchy level. In particular it is desirable to have a cluster algorithm that is capable of creating a large number of moderately sized clusters. The classical single-linkage algorithm is inappropriate for this task. Indeed, by percolation theory we can either observe the occurrence of an infinite component or we obtain a large number of clusters consisting of only very few sensors, see e.g. [25, 31].

Next let us discuss an application at a microscopic scale. Recently it was observed [26, 33, 34] that the arrangement of chondrocyte cells in the superficial zone of human cartilage form distinct patterns according to the health status of the cartilage, see Figure 1.1 for an illustration. By being able to analyze these patterns and to understand their connection to the health status of the examined cartilage, it might become possible to see already on the scale of cells, where degenerative diseases such as osteoarthritis might develop soon. In order to create a diagnostic tool based on this observation, one important step is to develop point-process models for such cartilage patterns, see [26] for early results in this direction. Using this type of stochastic model makes diagnoses based on cell patterns amenable to rigorous statistical testing.

Motivated by these applications, the goal of this chapter is to describe how statistical techniques for spatial point processes can be used in the validation of cluster identification

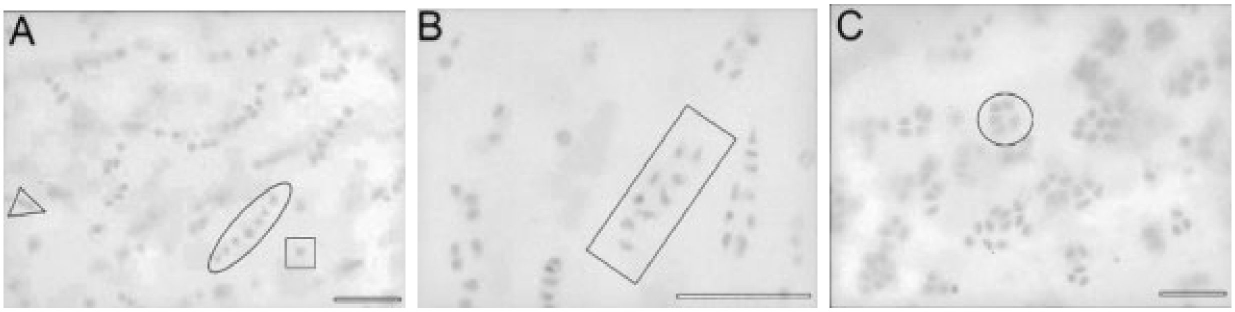

Figure 1.1: Chondrocyte patterns in the superficial zone of the condyle of the human knee joint. Figure taken from [33].

and classification steps, where the chapter is structured as follows. In Section 1.2, we present a generalization of the classical single-linkage algorithm which may be useful in situations where substantial inhomogeneities of cluster densities can be observed. To tackle the problem of cluster classification we recall in Section 1.3 the probabilistic method based on random forests which was introduced in a seminal paper of Breiman [7]. In Section 1.4, we show how the introduced methods can be applied to point patterns formed by cell nuclei in human cartilage tissue. Finally in Section 1.5 we show how statistical techniques for spatial point processes can be used in the validation of the identification and classification steps, where we will first describe the model used as test scenario and then explain our results.

1.2 Generalized single-linkage algorithm

We first review some techniques to reconstruct clusters and correctly determine their shape from a single realization of the model. Naturally this task splits up into two parts. In the current section we discuss the issue of cluster identification while Section 1.3 is devoted to the discussion of the classification step. For generalities on the single-linkage algorithm and other classical hierarchical clustering algorithms we refer the reader to Chapter **??**.

reference to

1.2.1 Description of the algorithm

When considering the identification of clusters generated by a point-process model, the classical method of the single-linkage algorithm seems a reasonable choice. Indeed this algorithm is capable of identifying e.g. long string-like clusters generated inside elongated ellipsoids, see Section 1.5. However the presence of inhomogeneities of cluster densities make it difficult to apply this algorithm directly. Recall from Chapter **??** that the classical single-linkage algorithm has only one parameter corresponding to a global threshold value. Therefore it is difficult to choose a single global threshold value appropriate for both dense as well as sparse clusters. We return to this point in Section 1.5.4 where we show numerical evidence for this observation. We also refer the reader to the survey paper [28] for related generalizations of classical single-linkage (see also [17] and [32]).

Chapter 2.1

In order to describe our generalization of the classical single-linkage algorithm, we recall some basic notions from graph theory. A graph G is called *locally finite* if every vertex has finite degree. Furthermore a *rooted graph* is a graph $G = (\varphi, E)$ together with a distinguished vertex $v_0 \in \varphi$. Also recall that a *minimum spanning tree* on a finite set $\varphi \subset \mathbb{R}^3$ is a tree of minimal total length among all trees whose vertices are given by φ. Recall from Chapter **??** that the single-linkage algorithm defines the clustering corresponding to the connected components of the graph obtained from deleting all edges of length larger than the global threshold value from the minimum spanning tree. To deal with the problem of varying cluster densities described above we propose to consider a generalization of the single-linkage algorithm capable of taking into account not only the length of a specified edge but also the local geometry of the minimum spanning tree close to this edge. This approach can be seen as a mathematical formalization of the graph-theoretical methods introduced in [36] for the detection of clusters of different shapes which also make use of local characteristics of minimum spanning trees.

Chapter 2.1

Definition 1. Write $\mathcal{G}^*$ for the set of all locally finite rooted graphs in $\mathbb{R}^3$. A *splitting rule* is a function $f : \mathcal{G}^* \times \mathcal{G}^* \times [0, \infty) \to \{0, 1\}$ satisfying $f(g_1, g_2, x) = f(g_2, g_1, x)$ for all $g_1, g_2 \in \mathcal{G}^*$ and $x \in [0, \infty)$.

A splitting rule can be used to define a clustering in the following way.

Definition 2. Let $\varphi \subset \mathbb{R}^3$ be locally finite and let $f : \mathcal{G}^* \times \mathcal{G}^* \times [0, \infty) \to \{0, 1\}$ be a splitting rule. Write $T = (\varphi, E_T)$ for the (Euclidean) *minimum spanning forest* on φ, i.e., $\{x, y\} \in E_T$ if and only if there do not exist an integer $n \geq 1$ and a sequence of vertices $x = x_0, x_1, \ldots, x_n = y \in \varphi$ such that $|x_i - x_{i+1}| < |x - y|$ for all $0 \leq i \leq n - 1$. For $\{x_1, x_2\} \in E_T$ write T_{x_1}, T_{x_2} for the connected components of $(\varphi, E_T \setminus \{\{x_1, x_2\}\})$ containing x_1 and x_2, respectively (and rooted at x_1 and x_2, respectively). Then denote by φ_f the family of connected components of the graph $(\varphi, \{\{x_1, x_2\} \in E_T : f(T_{x_1}, T_{x_2}, |x_1 - x_2|) = 1\})$ and call φ_f *the clustering of* φ *induced by* f.

1.2.2 Examples of splitting rules

To illustrate the usage of the generalized single-linkage algorithm introduced in Section 1.2.1, let us consider several examples of splitting rules.

Example 3. For $f(g_1, g_2, \ell) = \mathbb{1}_{\{x : x < \alpha\}}(\ell)$ we recover the usual definition of single-linkage clustering at level $\alpha > 0$, where $\mathbb{1}_A$ is the indicator of the set A. See Figure 1.2 for an illustration of this clustering rule.

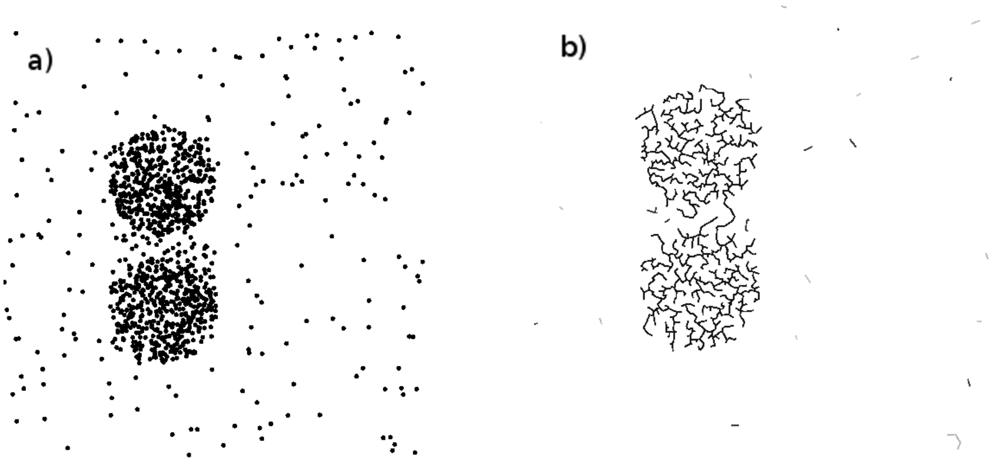

Figure 1.2: Clustering induced by the splitting rule of Example 3

Example 4. For a finite set A denote by $|A|$ the number of elements in A. For $n \geq 1$ we define $f(g_1, g_2, \ell) = 1 - \mathbb{1}_{\{x : x > n\}}(|g_{1,\ell}|)\mathbb{1}_{\{x : x > n\}}(|g_{2,\ell}|)$, where $g_{i,\ell}$ denotes the connected component containing the root of g_i of the subgraph of g_i obtained by retaining only those edges of length not exceeding ℓ. See Figure 1.3 for an illustration of this splitting rule.

Observe that as $n \to \infty$ the clustering induced by f approaches the minimum spanning forest from below. Furthermore this splitting rule has the property of being equivariant under scaling, i.e. for all $c > 0$ we have $(c\varphi)_f = c(\varphi_f)$.

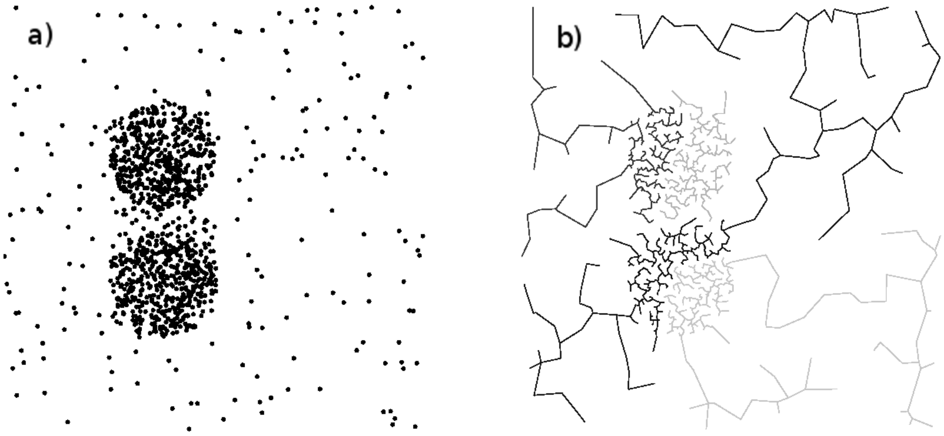

Figure 1.3: Clustering induced by the splitting rule of Example 4

Although the scaling property of these graphs makes them interesting candidates for the inhomogeneous clustering problem, we still need to add further modifications. Indeed it can be shown [19] that on a large class of stationary point processes these graphs almost surely (a.s.) do not percolate (i.e. all connected components are finite). This property is beneficial e.g. for clustering of large-scale sensor networks, where one is faced with the task of separating a diffuse set of points into a large number of connected components of similar and rather moderate sizes, see [3, 4, 14]. In [4] a Voronoi-based hierarchical network is constructed and the energy-efficiency of this network is analyzed in a simulation study. One simplifying assumption in [4] consists in the hypothesis that each sensor can communicate with all other sensors in their radio range using the same amount of energy. However, in reality the energy cost of transmissions increases super-linearly in Euclidean distance [10] which suggests that minimum-spanning-tree based approaches, where short connections are preferred, could lead to a further increase in energy efficiency. By making use of the truncation parameter n in Example 4 it is guaranteed that the determination of the edge set at a given node does not involve the complete graph but can be achieved in a neighborhood of this graph and additionally allows for the introduction of clusters at different hierarchy level.

However in the current setting, where clusters contain a large number of points, this property is rather detrimental since it tends to destroy connectivity inside clusters, even if mutually well-separated.

Example 5. Taking the discussion following Example 4 into account we propose to use the following splitting rule $f : \mathcal{G}^* \times \mathcal{G}^* \times [0, \infty) \to \{0, 1\}$. For $0 < \alpha_1 < \alpha_2$ and $n_0, n_1 \geq 1$ consider

$$\begin{aligned} f(g_1, g_2, \ell) = \mathbb{1}_{\{x : x < \alpha_2\}}(x) \left(1 - \mathbb{1}_{\{x : \alpha_1 < x\}}(\ell) \mathbb{1}_{\{x : x > n_0\}}(|g_{1,\ell}|) \mathbb{1}_{\{x : x > n_0\}}(|g_{2,\ell}|)\right. \\ \left. \times \left(1 - \mathbb{1}_{\{x : x \leq n_1\}}(|g_{1,\ell}|) \mathbb{1}_{\{x : x \leq n_1\}}(|g_{2,\ell}|)\right)\right) . \end{aligned}$$

See Figure 1.4 for an illustration of this clustering rule.

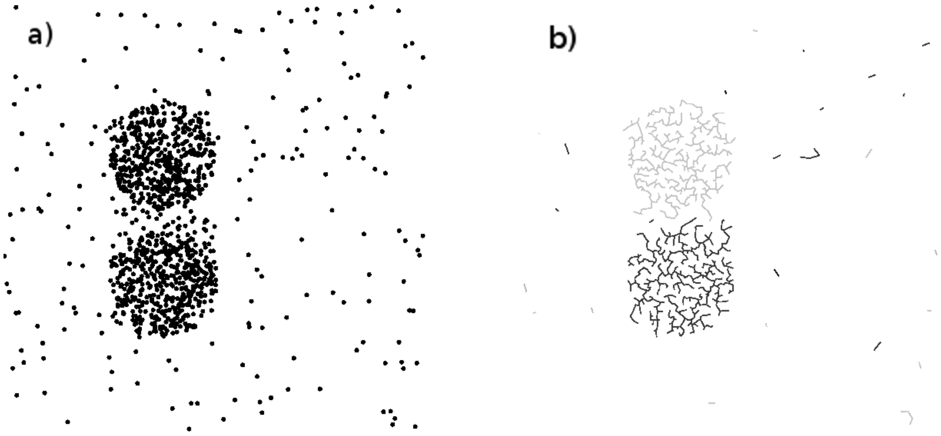

Figure 1.4: Clustering induced by the splitting rule of Example 5

Let us explain the intuition behind the various components of f.

1. The first factor $\mathbb{1}_{\{x : x < \alpha_2\}}(\ell)$ ensures that it is possible to distinguish between clusters having the property that distances between points from one cluster to points from the other cluster are rather large (i.e. having at least distance α_2 from each other).
2. The term $\mathbb{1}_{\{x : \alpha_1 < x\}}(\ell)$ makes sure that close points (i.e. whose distance is at most α_1) always end up in the same cluster.
3. By including the term $\mathbb{1}_{\{x : x > n_0\}}(|g_{1,\ell}|) \mathbb{1}_{\{x : x > n_0\}}(|g_{2,\ell}|)$ it is granted that the algorithm does not create unrealistically small clusters (unless they are already separated due to item 1).

4. Finally, the term $1 - \mathbb{1}_{\{x:x\leq n_1\}}(|g_{1,\ell}|)\mathbb{1}_{\{x:x\leq n_1\}}(|g_{2,\ell}|)$ is included to prevent that the algorithm creates unrealistically large clusters.

1.2.3 Further methods

The generalized single-linkage algorithms proposed in Sections 1.2.1 and 1.2.2 are mainly designed for cluster identification in off-grid data. Further very interesting methods using spatial processes for cluster analysis of pixel-based data are discussed in [29, 30], where a Bayesian approach is considered in conjunction with techniques of Markov random fields to reconstruct clusters of objects from image data arising in meteorology and astrophysics. We also remark that probabilistic methods for cluster analysis of pixel-based data such as the stochastic watershed algorithm play an important role in computational materials science, where a crucial task is the extraction of voxel clusters from (noisy) image data, see [13, 8, 16, 35]. Furthermore, model-based clustering could prove to be an alternative approach for our problem to identify and classify patterns which are formed by stationary point processes (see Section 1.5 for a more detailed discussion).

1.3 Classification methodology

After having discussed the task of cluster identification in Section 1.2, we would like to classify these entities according to their shapes. To present the main ideas of the principal methodology clearly, we consider a stochastic point-process model which is capable of creating two types of cluster shapes, e.g., balls and elongated ellipsoids, see Section 1.5. For this classification step we use the random-forest approach proposed by Breiman [7]. To provide the reader with a gentle introduction to this topic we recall some basic definitions and results from classification theory based on resampling methods, see also Chapter **??**.

Chapter 6.3

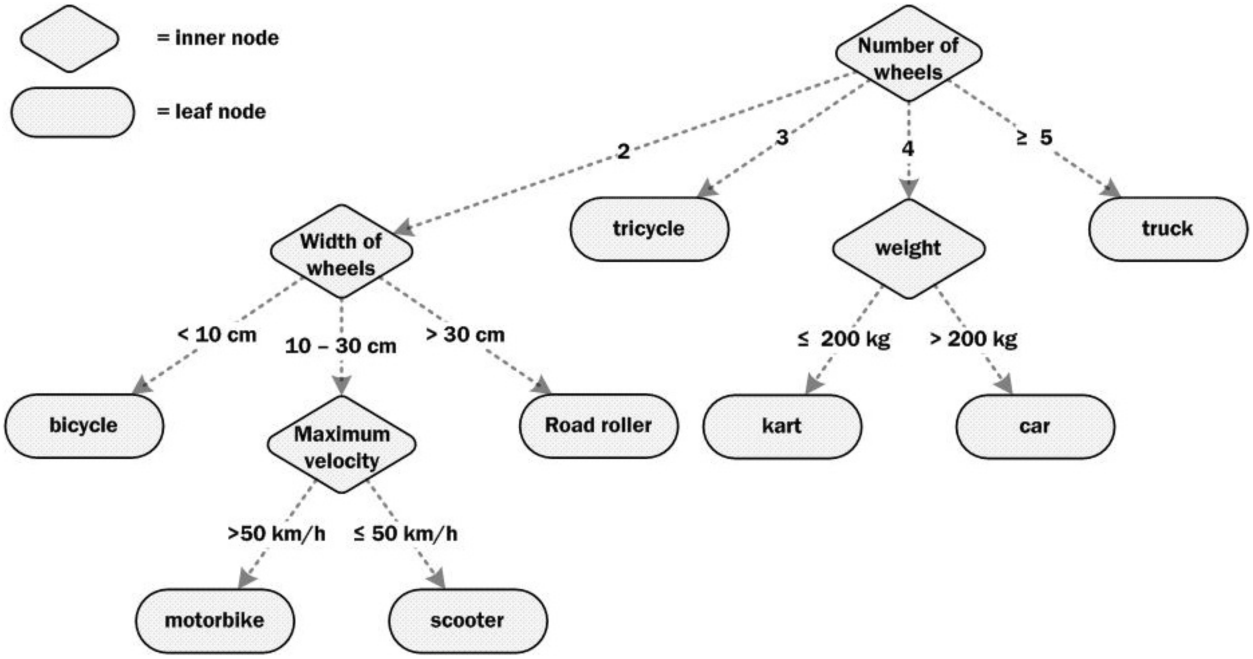

Figure 1.5: Example of a decision tree classifying vehicles.

1.3.1 Preliminaries on decision trees

To understand the methodology of random forests, let us first review the notion of deterministic decision trees, a classical tool in classification theory, see e.g. [12, 18]. Informally speaking, the idea of decision trees is to classify an object by answering a sequence of questions concerning its properties. The question posed at a given point does not need to be independent of the answer to its preceding questions, i.e. each answer can influence further questions. Such a sequence of questions and possible answers can be represented by a directed graph, more specifically by a tree, in which each inner node corresponds to a question and each edge corresponds to a possible answer or a set of possible answers to the question of the node it is emanating from, see Figure 1.5. The decision tree is always rooted, where the root node corresponds to the question from which we start classification. Furthermore, each leaf node, i.e. each node with no further leading edges, needs to be associated with a category label. To classify a given object, one starts with the question at the root node and consecutively follows the edges with the answers applying to the currently classified object until a leaf node is reached. The label of this leaf node is the category our object is assigned to.

It is important to make sure that the sets of answers corresponding to the edges departing from a node should be pairwise disjoint and exhaustive. This means, for each possible answer there should be one and only one departing edge that fits this answer and that can be followed in this case. The category labels at the leaf nodes do not need to be unique, so there can be more than one path through the tree leading to the same category. In our case, we only consider binary decision trees, which means that at each inner node there are exactly two descending nodes. This is reasonable because any decision tree can be represented by a binary one and because it strongly simplifies the construction principle to be explained in Section 1.3.2.

1.3.2 Measures of impurity and construction principle

Next, we need a way to measure the quality of the classification determined by a fixed decision tree. We have a labeled training dataset $\widetilde{M} \subset M \times C$, where M is the space of objects we want to construct a classifier for (in our case the set of possible clusters in $\mathbb{R}^3$, i.e. $M = \{\varphi \in \mathbb{R}^3 : 3 \leq |\varphi| < \infty\}$), and C is the finite set of categories to consider (in our case C contains the two elements *ball* and *elongated ellipsoid*). One can interpret a decision tree as a successive splitting of our training dataset $\widetilde{M}$ into subsets: At each node a subset of the dataset $\widetilde{M}$ is split into one part going to the left and one part going to the right descending node. Building a decision tree can therefore be done by finding a sequence of splits of $\widetilde{M}$ that is optimal in some way. In order to measure the quality of a split quantitatively, the standard approach is to make use of an impurity measure, see e.g. [12, 18]. If V denotes the vertex set of the graph representing the decision tree at hand, an *impurity measure* is a function $i \colon V \to [0, \infty)$ satisfying the following conditions:

1. The impurity $i(v)$ at a node $v \in V$ should be 0 if all patterns reaching v are from the same category.
2. The impurity at a node should be large if all categories are equally present.

There are three popular impurity measures we want to mention here, which all fulfill the conditions given above. If v is any node of a decision tree, we write $\widetilde{M}_v$ for the subset of

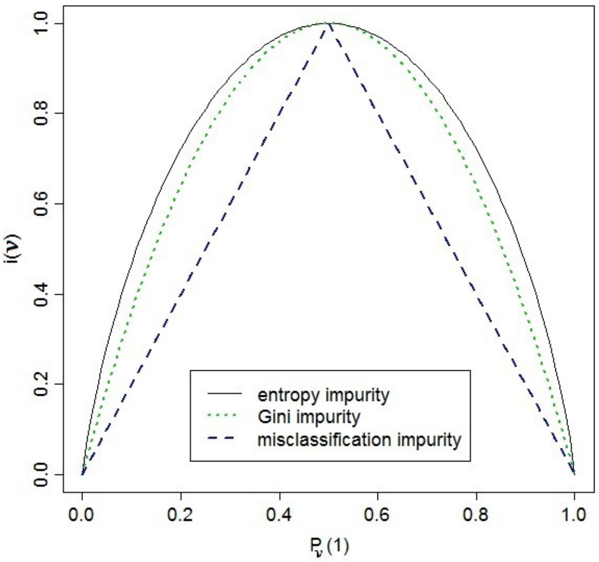

Figure 1.6: Illustration of (appropriately scaled) impurity measures in the two-category case, i.e. $C = \{1, 2\}$.

all patterns in $\widetilde{M}$ that reach v and we denote by $P_v(j)$ the fraction of patterns in $\widetilde{M}_v$, that belong to category $j \in C$. Given this notation, we define

- *Gini impurity* $i_G : V \to [0, \infty)$ by $i_G(v) = \frac{1}{2}\left(1 - \sum_{j \in C} P_v^2(j)\right)$,
- *entropy impurity* $i_e : V \to [0, \infty)$ by $i_e(v) = -\sum_{j \in C} P_v(j) \log_2(P_v(j))$,
- *misclassification impurity* $i_m : V \to [0, \infty)$ by $i_m(v) = 1 - \max_{j \in C} P_v(j)$.

In general, none of these three impurity measures is superior to the others and it depends on the specific application as to which one is to be preferred. An illustration of these functions in the two-category case can be found in Figure 1.6.

To define which split is the best according to a fixed impurity measure $i : V \to [0, \infty)$, we compare the impurity $i(v)$ of the node v to the impurities $i(v_L)$ and $i(v_R)$ of the two descending nodes v_L (to the left) and v_R (to the right) that result from the split. It is desirable to maximize the impurity decrease from v to v_L and v_R. We therefore define the *drop of impurity* as $\Delta i(v) = i(v) - P_v(v_L)i(v_L) - (1 - P_v(v_L))i(v_R))$, where $P_v(v_L)$ is the fraction of patterns at node v that go to the left descendent node v_L and where $i(v_L)$ and $i(v_R)$ are the impurities at the nodes v_L and v_R. The best split at the node v should maximize $\Delta i(v)$.

After having introduced means to measure the quality of a split, it is desirable to construct decision trees that are optimal with respect to the above measure. In our case, we only want to use one property for each decision, because a simple structure of the decision tree is essential for fast computations as they are needed in random forests. This means that if we consider k properties of our objects in M, we have to find k best splits of $\widetilde{M}_v$ at a given node v – one according to each property. This yields k values $\Delta_1 i(v), \ldots, \Delta_k i(v)$ for the respective drops of impurity. The best property to split on can then be chosen as the one corresponding to the maximal value of $\Delta_1 i(v), \ldots, \Delta_k i(v)$.

In order to find the best split according to a given (numerically valued) property $\ell \in \{1, \ldots, k\}$ for numerical data as in our case, see Section 1.5.3 below, we confine ourselves to splittings of $\widetilde{M}_v$ according to a certain value of threshold. That is, one subset consists of all objects in $\widetilde{M}_v$ for which the value of the considered property is below this threshold, the other one to the objects for with the value is greater or equal. The best value of threshold is then the solution of a one-dimensional optimization problem, which maximizes $\Delta_\ell i(v)$ and can be solved numerically.

In the simplest case, the above explained procedure is started with the full training dataset $\widetilde{M}$ and continued at each descending node, until either the descendent node v is pure, which means that $i(v) = 0$, or until none of the possible splits leads to a further drop of impurity, e.g. if two objects from different categories $i \neq j$ in C have exactly the same properties. Then, the node $v \in V$ is declared a leaf node and associated with the category $j \in C$ such that the majority of objects present at node v belongs to category j. Note that it is not assured that by consecutively choosing the locally optimal split also a globally optimal decision tree is built. There are several approaches for further optimizing this procedure, but for the construction of random forests considered in Section 1.3.3 below, only this simple form of decision trees is needed. We refer the reader to [12] for more detailed information on this topic.

1.3.3 Random forests

One major problem of optimal deterministic decision trees discussed in Section 1.3.2 is that they are optimized with respect to minimizing the *training error* in contrast to the desired minimization of the *classification error*. In other words, these deterministic trees often lead to a considerable overfitting to the special training set at hand, i.e., the training error is much smaller than the classification error. The basic idea of the random-forest approach due to Breiman [7] is to resolve this problem by using a suitable randomization. The random-forest approach is very flexible and yields robust results also for problems with missing data or where the number of considered properties (so-called predictors) is huge when compared to the size of the training set. There are many approaches for randomization, the most popular are *random feature selection*, where only a randomly chosen part of the available properties is considered for each splitting in a decision tree, and *bagging*, where each decision tree is built on the basis of a new training dataset which is drawn with replacement from the original training dataset. Since in our application the number of predictors is small, we concentrate on the description of randomization by bagging, see [7] for a detailed description of these and further techniques.

Let $r \geq 1$ be the size of the training set $\widetilde{M} = \{\widetilde{m}_1, \ldots, \widetilde{m}_r\}$. The basic idea of bagging is to generate new random training sets by drawing r elements of $\widetilde{M}$ uniformly with replacement. Thus, we consider a random vector $\mathbf{\Theta} = (\Theta^{(1)}, \ldots, \Theta^{(r)})$ and the random set $\widetilde{M}_{\mathbf{\Theta}} = \{\widetilde{m}_{\Theta^{(1)}}, \ldots, \widetilde{m}_{\Theta^{(r)}}\}$, where $\Theta^{(1)}, \ldots, \Theta^{(r)}$ are independent random variables with uniform distribution on $\{1, \ldots, r\}$. Typically, drawing with replacement has the effect of leaving out approximately $(1 - 1/r)^r \approx 37\%$ of the elements of the original training set $\widetilde{M}$ which are not involved in the construction of the best-split decision tree and therefore may be used e.g. for cross-validation purposes. These elements are called *out-of-bag samples*.

By repeating this random generation of training sets independently $n \geq 1$ times, i.e. by considering n independent copies $\mathbf{\Theta}_1, \ldots, \mathbf{\Theta}_n$ of $\mathbf{\Theta}$, we obtain a sequence of independent random training sets $\widetilde{M}_{\mathbf{\Theta}_1}, \ldots, \widetilde{M}_{\mathbf{\Theta}_n}$ and may construct decision trees $T_{\mathbf{\Theta}_1}, \ldots, T_{\mathbf{\Theta}_n}$ as described in Section 1.3.2, where each decision tree $T_{\mathbf{\Theta}_i}$ is based on the random training set $\widetilde{M}_{\mathbf{\Theta}_i}$. Hence, the randomized decision trees $T_{\mathbf{\Theta}_i}$ are random objects as well and as the construction of a decision tree $T_{\boldsymbol{\theta}_i}$ is deterministic once we consider a fixed realization $\boldsymbol{\theta}_i$ of $\boldsymbol{\Theta}_i$, the decision trees $T_{\boldsymbol{\Theta}_1}, \ldots, T_{\boldsymbol{\Theta}_n}$ are independent copies of the random classifier $T_{\boldsymbol{\Theta}}$.

The collection $\{T_{\boldsymbol{\Theta}_1}, \ldots, T_{\boldsymbol{\Theta}_n}\}$ of random decision trees is called *random forest*. For realizations $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_n$ of $\boldsymbol{\Theta}_1, \ldots, \boldsymbol{\Theta}_n$, respectively, we call $\{T_{\boldsymbol{\theta}_1}, \ldots, T_{\boldsymbol{\theta}_n}\}$ a realization of a random forest. We can construct a classifier based on $T_{\boldsymbol{\theta}_1}, \ldots, T_{\boldsymbol{\theta}_n}$ as follows. If $x \in M$ is an element to be classified, we first consider the category that was assigned to x by the decision tree $T_{\boldsymbol{\theta}_i}$ for each $i \in \{1, \ldots, n\}$ and afterwards assign x to the most frequent category in this list. We define the *margin function* $mg\colon M \times C \times (\mathbb{R}^r)^n \to [-1, 1]$ of a random forest by

$$mg(x, y, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_n) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}_{\{h(x,\boldsymbol{\theta}_i)=y\}} - \max_{j \neq y} \left(\frac{1}{n} \sum_{i=1}^{n} \mathbb{1}_{\{h(x,\boldsymbol{\theta}_i)=j\}} \right), \tag{1.1}$$

where $\mathbb{1}_A$ is the indicator of the set A and $h(x, \boldsymbol{\theta}_i)$ denotes the category assigned to x by $T_{\boldsymbol{\theta}_i}$. This function measures how far the average number of votes for the right category exceeds the average numbers of votes for all other categories. In particular, $mg(x, y, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_n) < 0$ if and only if the object $x \in M$ with category $y \in C$ is misclassified by the (deterministic) forest $\{T_{\boldsymbol{\theta}_1}, \ldots, T_{\boldsymbol{\theta}_n}\}$.

If in (1.1) we insert the random vectors $\boldsymbol{\Theta}_1, \ldots, \boldsymbol{\Theta}_n$ instead of their realizations $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_n$ and a random object X together with its random category Y instead of (x, y), we may consider the probability of misclassification. Note that (X, Y) and $(\boldsymbol{\Theta}_1, \ldots, \boldsymbol{\Theta}_n)$ are assumed to be independent random vectors on the same probability space. The probability for misclassification conditioned on $\boldsymbol{\Theta}_1, \ldots, \boldsymbol{\Theta}_n$ is given by

$$PE_n^* = P(mg(X, Y, \boldsymbol{\Theta}_1, \ldots, \boldsymbol{\Theta}_n) < 0 \mid \boldsymbol{\Theta}_1, \ldots, \boldsymbol{\Theta}_n)$$

called the *generalization error*, where

$$P(mg(X, Y, \boldsymbol{\Theta}_1, \ldots, \boldsymbol{\Theta}_n) < 0 \mid \boldsymbol{\Theta}_1, \ldots, \boldsymbol{\Theta}_n) = \mathbb{E}(\mathbb{1}_{\{mg(X,Y,\boldsymbol{\Theta}_1,\ldots,\boldsymbol{\Theta}_n)<0\}} \mid \boldsymbol{\Theta}_1, \ldots, \boldsymbol{\Theta}_n),$$

i.e., the generalization error PE_n^* is a random variable. Using the strong law of large numbers, one can show (see [7]) that for unboundedly growing n the generalization error PE_n^* converges to a fixed value, which is given by

$$\lim_{n\to\infty} PE_n^* = P\big(P\left(h(X,\mathbf{\Theta}) = Y \mid X, Y\right) - \max_{j\neq Y}\left(P\left(h(X,\mathbf{\Theta}) = j \mid X, Y\right)\right) < 0\big), \tag{1.2}$$

where $h(X,\mathbf{\Theta})$ is the category which is assigned to X by the random classifier $T_{\mathbf{\Theta}}$. This is the reason why random forests do not overfit.

Apart from (at least partially) resolving the problem of overfitting, the bagging technique has the additional advantage of removing the need to manually leave out a part of the training set for the purpose of cross-validation. For each labeled object (x, y) we can average the votes of only those decision trees $T_{\boldsymbol{\theta}_i}$ that did not receive (x, y) in their training set $\widetilde{M}_{\boldsymbol{\theta}_i}$. We assign x to the category which gets most votes in this modified procedure and call this *out-of-bag classifier*. For a realization $\{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_n\}$ of the random forest $\{\mathbf{\Theta}_1, \ldots, \mathbf{\Theta}_n\}$ we can then define the *out-of-bag classification error* of a specified class $j \in C$ as the fraction of elements in $\widetilde{M}$ from class j that were misclassified by the out-of-bag classifier. By repeatedly generating realizations of the considered random forest and averaging the out-of-bag errors over all iterations, we obtain an estimator for the classification error of the random forest classifier for this problem. In contrast to the naive error rate, this estimator avoids the bias arising if objects are classified which were already included in the training set. However, as each out-of-bag predictor only uses about one third of all decision trees of the forest, it should be assured that a sufficiently high number of decision trees is generated for each random forest.

Finally, the randomization approach also provides us with a tool to single out the most relevant predictors. Indeed let $j \in C$ be a category and $\pi \in \{1, \ldots, k\}$ be a predictor. In each iteration we may consider the effect of applying a random permutation on the values of predictor π among all out-of-bag elements of class j. Then we may compare the out-of-bag classification error of class j before and after this permutation and compute its relative increase. This value measures the extent to which disturbing the predictor π decreases the classification accuracy of elements of class j. Averaging these quantities over all iterations we obtain a measure of how essential predictor π is for the classification in category j. This value is called *importance score*. A more formal approach to select the most important variables with the aim of creating a parsimonious model is described in [27]. To achieve this goal a convex penalty for trees is used which is related to the so-called nonnegative Garrote [6].

An implementation of random forests based on the original Fortran code by Breiman can be found in the R package *randomForest*, see [21].

1.4 Application to patterns of cell nuclei in cartilage tissue

After having introduced our methodology, we give a biomedical example for its application. As already mentioned in the introduction of this chapter, the cell nuclei of chondrocytes in human cartilage form distinct patterns according to the type, localization and health state of the cartilage, see [34, 26]. We now apply our methods to the point patterns formed by these cell nuclei in order to detect different types of clusters. First, we describe our data together with some biological background and apply cluster analysis to the originate point patterns. Later in this section we assign the obtained clusters to different types by use of the random-forest approach and explain our results.

1.4.1 Data description

Cartilage tissue contains neither nerves nor veins, it consists of chondrocytes – the only cells that are found in it – and an extracellular matrix that is produced by these chondrocytes and encloses them. The matrix mainly consists of proteoglycans and collagens, where the exact composition varies according to the specific type of cartilage. The chondrocytes are organized in super-imposed layers, a superficial zone at about 0-10%, a middle zone at 10-40% and a deep zone at 40-100% tissue depth. In the superficial zone, the cell density is very high and chondrocytes are organized in horizontal patterns as already illustrated in Figure 1.1, while in the deeper zones the cell density is much lower and chondrocytes form vertical columns.

All our investigated cartilage samples were taken from the human knee joint of patients of the BG Unfallklinik Tübingen who were treated with ACI (autologous chondrocyte implantation), only the superficial zone of the cartilage was examined. ACI is a biomedical treatment method that tries to repair local cartilage damages as they occur, for example, as a consequence of an accident. Therefore, intact cartilage is taken from an area of the joint which is not meant to bear much weight. The cells from this healthy cartilage are isolated, stimulated in vitro to make them proliferate and applied to some carrier material, for example a biomembrane. When there are enough chondrocytes grown, this carrier material containing the cells is reimplanted into the damaged joint. The pieces of cartilage we examined are the ones that are taken out from around the lesion during this surgery in order to clear enough space for the implant.

We distinguished three different health states of these cartilage samples: intact cartilage, cartilage with early osteoarthritis and already severely damaged cartilage. The respective health state was determined by visual inspection according to the following criteria: the sample was considered intact if no or nearly no damage was visually recognizable, i.e. if its surface was smooth and glossy or only slightly roughened. It was considered to be cartilage with early osteoarthritis if fibrillations, fissuring or major surface roughening could be detected and it was considered severely damaged if full defects could be found already.

In total we investigated 20 different cartilage samples, which were taken from five different patients. In each of these samples, the cell nuclei were stained with a fluorescent dye and three-dimensional images were taken by use of a fluorescence microscope and the technique of structured illumination. This technique makes it possible to record two-dimensional sections through the specimen separately without or with only very few noise from above or below the plane in focus. A stack of these two-dimensional sections can later be combined to the final three-dimensional binary image, see also [22]. Note that 10 of the 20 images we obtained like this were taken with ten-fold, 10 with twenty-fold magnification.

After some preprocessing steps we obtained the three-dimensional coordinates of all cell nuclei contained in each binary image, given in μm relative to the observation window.

As the cells found in the type of cartilage we examined form clusters with characteristic shapes according to its health state, our goal was to use the methods introduced in Sections

1.2 and 1.3 to first extract and then classify the clusters contained in our samples.

1.4.2 Results

In order to extract the clusters from the point patterns, we applied the single-linkage algorithm as already discussed in Section 1.2, see also Chapter **??** using the Euclidean distance and a threshold value of $22.5\mu m$. This value led to optically good results and is close to the values that have already been used earlier in [26] for the analysis of similar data. An example of a clustered point pattern can be found in Figure 1.7. Cell nuclei which belong to a given cluster are plotted using the same symbol.

Chapter 2.1

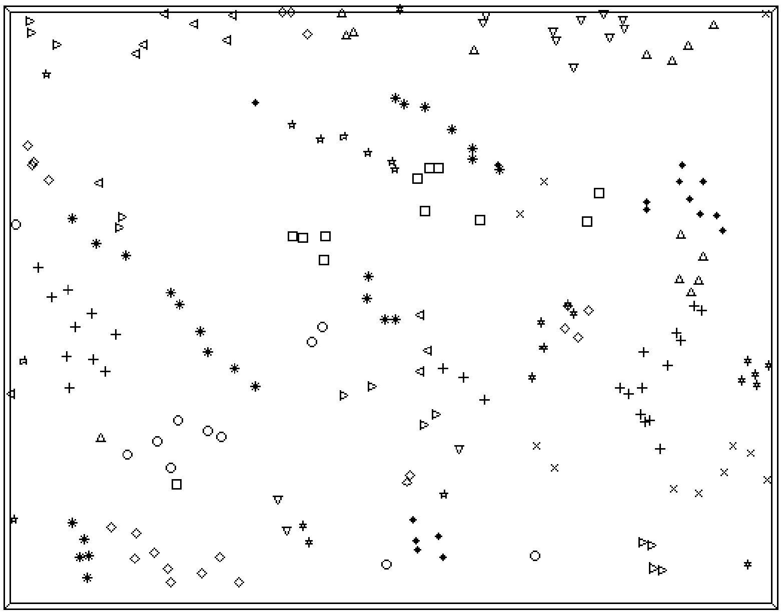

Figure 1.7: 3D pattern of cell nuclei with detected clusters (indicated by different symbols).

Next we classify the resulting clusters according to their shapes. We therefore shortly discuss the choice of properties, which were considered for classification and the choice of categories the clusters shall be assigned to. We excluded clusters containing less than three points from the further analysis, as pairs and singletons do not show any relevant geometric substructure and thus form their own trivial categories. So the space of objects we want to classify consists of the set of all finite subsets of $\mathbb{R}^3$ containing at least three points, i.e. $M = \{\varphi \subset \mathbb{R}^3 : 3 \leq |\varphi| < \infty\}$.

The set of possible categories E the clusters in M should be assigned to by the classifier we are going to construct are the specific patterns that are "typical" for the health states of human cartilage according to earlier studies (see [33]). Three categories will be distinguished in the following.

1. **string:** A group of points that are one after another lying preferably regularly on a line. This pattern is typical for intact cartilage samples from the considered region of the human knee joint, a microscopic picture of a typical string can be found in Figure 1.1, enclosed by an ellipse.
2. **doublestring:** A string in which at least one cell has divided once more, leading to two cells lying next to each other *against* the orientation of the original string. In the extreme case, this results in two close parallel strings. This is typical for early osteoarthritis and the beginnings of degeneration in the cartilage. In Figure 1.1, a typical doublestring is enclosed by a rectangle.
3. **round cluster:** A round or ellipsoid-shaped aggregation of points without relevant substructure. This pattern is typical for already severely damaged cartilage and severe forms of osteoarthritis. A typical round cluster is enclosed by a circle in Figure 1.1.

We labeled the clusters according to the health state of the cartilage sample they had been extracted from, hence clusters from intact cartilage were labeled "strings", clusters from cartilage samples with beginning osteoarthritis were labeled "doublestring" and clusters from severely damaged cartilage were labeled "round cluster". Of course this method leads to some errors because not *only* the specific "typical" pattern is contained in each sample of cartilage but also patterns from the other two types and inarticulate patterns that do not fit any of the three categories do occur. But at least most of the patterns should be assigned to the right category by this procedure and labeling by visual inspection would be very subjective and need a huge amount of time. Using this method, we obtained 332 clusters labeled as strings, 811 clusters labeled as doublestrings and 98 round clusters.

As we want to classify the clusters according to their shapes, we mainly chose geometric features as properties for classification. In detail, we considered:

- the ratio of length and width, where length width and depth of a cluster were defined as the maximum extension in the direction of the coefficient vector of the first, second and third principal component of the cluster,
- the mean distance to the best line fitted through the cluster (by orthogonal distance regression),
- the mean distance to the best plane fitted through the cluster (by orthogonal distance regression),
- the coefficient of determination and the regression coefficient of the following regression: let dk be the distance from an extremal point in the cluster (chosen as one of the two points with the biggest distance) to its k^{th}-nearest neighbor, a line through the origin was fitted to the points (k, dk) by ordinary regression,
- the mean angle between the connection vectors from one extremal point to all other points,
- the mean angle between the connection vectors from each point (the both extremal points excluded) to its first and second nearest neighbor,
- the ratio of the mean distance from the centroid and the number of points in the cluster and
- the ratio of length and number of points.

In [7] evidence is provided that random forests are quite robust with respect to weak inputs. Therefore we put no additional effort in selecting the best among these features but considered all of them simultaneously.

We balanced the dataset by a bootstrapping technique. We took about 50% of the number of clusters from the biggest category, i.e. 408 as we have 811 doublestrings, and randomly

OOB estimates of error rates

	mean value	standard deviation
total	0.1249	0.0090
string	0.1435	0.0156
doublestring	0.2242	0.0200
round cluster	0.0069	0.0039

Test set error rates

	mean value	standard deviation
total	0.3948	0.0187
string	0.4946	0.0551
doublestring	0.3757	0.0255
round cluster	0.1918	0.2263

Table 1.1: Mean values and standard deviations of the error rates of 100 realizations of random forests with balanced training dataset.

drew that many clusters from each category with replacement. Note that this means that any cluster can be contained more than just once in the training dataset.

As test set we only considered clusters which were not selected at all for the training dataset. This was necessary because after resampling the out-of-bag estimators are most likely positively biased. Many clusters occur more than once in the training dataset, and hence, the idea of out-of-bag estimation does not work properly anymore. It may happen that although a cluster is not in the training dataset of a decision tree itself and is therefore considered out-of-bag for this decision tree, one or more identical copies of the same cluster are. A second cause of bias in favor of the classification is that patterns which are in the training dataset more than once are more likely to be classified correctly. Replicating a cluster has a effect similar as weighting it higher than other clusters, concerning both the classification itself and the computation of the error rates. A pattern with many replications in the training dataset is more likely to be classified correctly *and* the classification outcome is taken into account once for each replication. This bias is the reason why we decided earlier to retain a test set additional to the out-of-bag estimation of error rates.

The error rates we obtained seem quite high compared to other classification problems, see Table 1.1. However, one can see by optical inspection that the majority of clusters in each cartilage sample was classified correctly, see Figures 1.7 and 1.8. As the classification

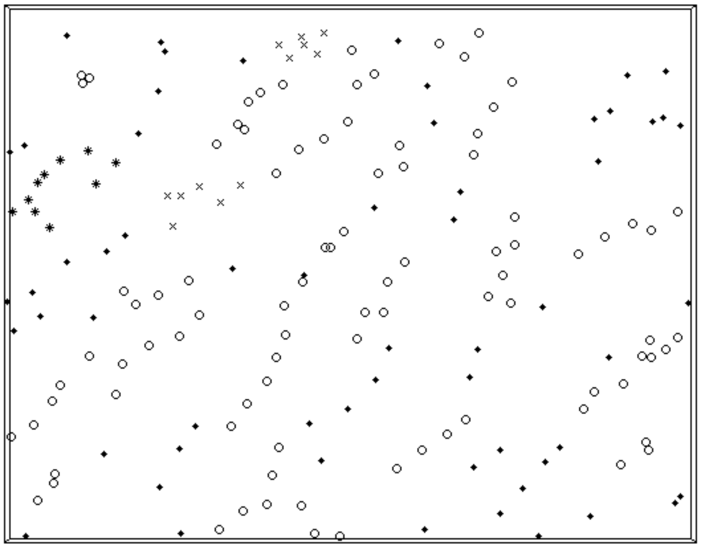

Figure 1.8: 3D pattern of cell nuclei from cartilage with beginning osteoarthritis. Clusters containing at least 3 points are shown according to whether they were classified as strings (symbol o), doublestrings (symbol *) or round clusters (symbol $\times$).

of single clusters can be seen as a preceding step to the classification of cartilage samples themselves, which is even more closely related to the problem of diagnosis, this might be sufficient. Furthermore, it has to be taken into account that the biological data we considered is highly variable, which led to a high variability of cluster shapes. Knowing that there is in fact quite a high amount of noise in our data, we consider the error rates acceptable.

1.5 Validation of clustering methodology

In this section we validate the methodology for cluster identification and cluster classification stated in Sections 1.2 and 1.3, respectively, by performing a Monte-Carlo analysis of a synthetic point-process example. Similar to the biomedical problem discussed in Section 1.4, our goal will be first to identify different clusters in a noisy environment and after this provide a correct classification of the type of the observed cluster shape. In particular, the main goal of our approach is not necessarily to associate every single data point with its true cluster, but to ensure that the numbers of identified clusters of a given shape are in accordance with those corresponding to the true clusters. Note that the point-process model discussed in this section would be a prototypical example for model-based clustering as discussed in [1, 5, 9, 11, 15, 23, 24] and also in Chapter **??**. In particular, since for simplicity we consider a synthetical example where the cluster shapes are given by ellipsoids, it would be particularly well-suited for this classical method. Nevertheless, as explained in Section 1.4, in practice one often has to deal with non-ellipsoidal shapes and it usually takes some effort to accommodate model-based clustering to this framework (see [1, 23, 24] for possible approaches to resolving this problem). Chapter 3.9

1.5.1 Description of the test scenario

As input data for the validation of the clustering and classification algorithms we consider synthetic patterns in $\mathbb{R}^3$ formed by realizations of a simple point-process model, see [20] for a detailed introduction to this class of spatial processes. Note that for simplicity of exposition we consider a three-dimensional example like in Section 1.4, although the methods described in this chapter can be applied in other dimensions as well. Let us first give an informal description of our model. We begin with a primary point process $X^{(1)}$ of possible locations of cluster centers. At each of these centers we place cluster areas with one of two possible random shapes – either a ball or an elongated ellipsoid, see Figure 1.9a. Afterwards we remove some of these cluster areas so that the remaining ones are pairwise disjoint (marked in gray in Figure 1.9a). We then decide for each of the remaining areas independently whether it is supposed to be filled with a dense or sparse cluster and place an appropriate number of points uniformly and independently inside this area, see Figure 1.9b. Finally some background noise is added, see Figure 1.9c.

We now provide a formal mathematical description of this model. For simplicity of exposition we consider a basic Poisson-type scenario exhibiting the properties of the informal description above. It would be interesting to conduct an extensive analysis on far more complicated (and more realistic) point-process models, but this would be beyond the scope of this expository chapter.

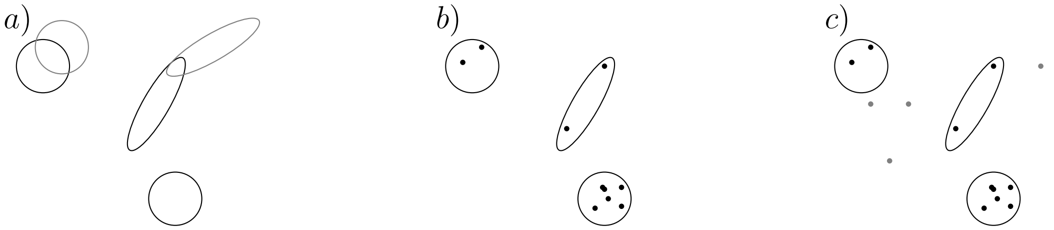

Figure 1.9: Construction principle of the point-process model

Write $\mathbb{N}$ for the family of all locally finite subsets $\varphi \subset \mathbb{R}^3$ and $\mathcal{C}$ for the family of all convex compact bodies of $\mathbb{R}^3$. Furthermore denote by SO_3 the group of orthogonal matrices in $\mathbb{R}^{3\times 3}$ with determinant 1 which can be identified with the group of three-dimensional rotations around the origin. The point process we are interested in is hierarchically built up as follows. Let $\mathbb{Z}_+ = \{1, 2, \ldots\}$ denote the set of positive integers and consider independent random variables $X^{(1)}, X^{(2)}, X^{(3)}, X^{(4)} \in \mathbb{N}$, $Z^{(5)} = \left\{Z_n^{(5)}\right\}_{n\geq 1}$, $Z^{(6)} = \left\{Z_n^{(6)}\right\}_{n\geq 1} \in \{0,1\}^{\mathbb{Z}_+}$, $U^{(7)} = \left\{U_n^{(7)}\right\}_{n\geq 1} \in [0,1]^{\mathbb{Z}_+}$ and $G^{(8)} = \left\{G_n^{(8)}\right\}_{n\geq 1} \in SO_3^{\mathbb{Z}_+}$ with the following properties. Assume that

1. $X^{(1)}$, $X^{(2)}, X^{(3)}, X^{(4)}$ are homogeneous Poisson point processes in $\mathbb{R}^3$ with intensities $\lambda_1, \lambda_2, \lambda_3, \lambda_4$, respectively, where $\lambda_3 < \lambda_4$,
2. $Z^{(5)}$, $Z^{(6)}$ are sequences of independent and identically distributed Bernoulli random variables with $1-\mathbb{P}\left(Z_1^{(5)} = 0\right) = \mathbb{P}\left(Z_1^{(5)} = 1\right) = p_5$ and $1-\mathbb{P}\left(Z_1^{(6)} = 0\right) = \mathbb{P}\left(Z_1^{(6)} = 1\right)$ $= p_6$ for some $p_5, p_6 \in [0,1]$,
3. $U^{(7)}$ is a sequence of independent and identically distributed random variables uniformly distributed on $[0,1]$,
4. $G^{(8)}$ is a sequence of independent and identically distributed random rotations uniformly distributed on SO_3.

Let $a > 0$ and let $K_0, K_1 \in \mathcal{C}$ be two fixed compact bodies centered at the origin, where we assume for simplicity that K_0 is an ellipsoid, and K_1 a sphere. Write $X^{(1)} \cap [-a/2, a/2]^3 = \left\{X_1^{(1)}, \ldots, X_N^{(1)}\right\}$ for the restriction of the Poisson process $X^{(1)}$ to the cube $[-a/2, a/2]^3$, where $N = \left|X^{(1)} \cap [-a/2, a/2]^3\right|$ denotes the total number of points of $X^{(1)}$ in $[-a/2, a/2]^3$.

For $1 \leq i \leq N$ we say that $X_i^{(1)}$ survives the construction if there does not exist $1 \leq j \leq N$ with $U_i^{(7)} < U_j^{(7)}$ and $\left(X_i^{(1)} + G_i^{(8)} K_{Z_i^{(5)}}\right) \cap \left(X_j^{(1)} + G_j^{(8)} K_{Z_j^{(5)}}\right) \neq \emptyset$. Let us write $\left\{X_{i_1}^{(1)}, \ldots, X_{i_{N'}}^{(1)}\right\}$ for the set of points of $X^{(1)}$ which survive the construction. The final point configuration X is then obtained as

$$X = \left(X^{(2)} \cap [-a/2, a/2]^3\right) \cup \bigcup_{j=1}^{N'} \left(X^{\left(3+Z_{i_j}^{(6)}\right)} \cap \left(X_{i_j}^{(1)} + G_{i_j}^{(8)} K_{Z_{i_j}^{(5)}}\right)\right), \tag{1.3}$$

where $X^{(2)} \cap [-a/2, a/2]^3$ is some noise.

To summarize, the main features of the point-process model defined in (1.3) are its ability to generate both different cluster shapes as well as different cluster densities. In particular, the Poisson process $X^{(2)}$ generates some background noise, N' denotes the random number of clusters in the cube $[-a/2, a/2]^3$ which survive, and $X^{\left(3+Z_{i_j}^{(6)}\right)}$ either generates dense clusters (if $Z_{i_j}^{(6)} = 1$) or sparse clusters (if $Z_{i_j}^{(6)} = 0$). Finally, $X_{i_j}^{(1)} + G_{i_j}^{(8)} K_{Z_{i_j}^{(5)}}$ generates the area of a spherical (if $Z_{i_j}^{(5)} = 1$) or ellipsoidal (if $Z_{i_j}^{(5)} = 0$) cluster.

Finally, after having introduced our point-process model and tools of cluster analysis, we now validate both the cluster identification as well as the cluster classification steps by extensive Monte-Carlo simulation based on the underlying spatial point process.

Let us first specify the parameters of the point-process model introduced in (1.3) so that the following properties are satisfied. On the one hand the number of cluster centers should be not too large (due to computational restrictions) but also not too small to guarantee that for a considerable fraction of cases the distance between two different clusters can be rather small and cluster separation becomes a non-trivial issue. Furthermore the density in dense clusters should be significantly higher than in sparse clusters so that there is a distinct advantage in using the generalized single-linkage algorithm in contrast to the standard single-linkage algorithm. Finally both cluster shapes should occur with a sufficiently high probability and also the number of background points should not be too small in order to increase the difficulty of cluster identification.

We fix the size of the sampling window $[-a/2, a/2]^3$ by putting $a = 200$ and define suitably chosen values for the intensities $\lambda_1, \dots, \lambda_4$ introduced above, where we put

- $\lambda_1 = 2.125 \cdot 10^{-6}$ (cluster center intensity),
- $\lambda_2 = 1.250 \cdot 10^{-5}$ (intensity of background noise),
- $\lambda_3 = 0.001705$ (intensity of cluster points in sparse clusters),
- $\lambda_4 = 0.01023$ (intensity of cluster points in dense clusters).

Furthermore the lengths of the three half-axes of the elongated ellipsoid K_0 are given by $70, 10$ and 10, respectively, while we let K_1 be a ball whose radius is determined by $\nu_3(K_1) = \nu_3(K_0)$, where $\nu_3(K)$ denotes the volume of the set $K \subset \mathbb{R}^3$. Finally, we put $p_5 = p_6 = 1/2$.

Observe that the values of $\lambda_1, \lambda_2, \lambda_3$ and λ_4 have been chosen so that in the sampling window $[-a/2, a/2]^3$ with side length $a = 200$ the expected number of cluster centers is equal to 17, the expected number of noise points is equal to 100 and such that the expected number of points in dense/sparse clusters is equal to 300 and 50, respectively. A realization of the point-process model for the test scenario described above is shown in Figure 1.10.

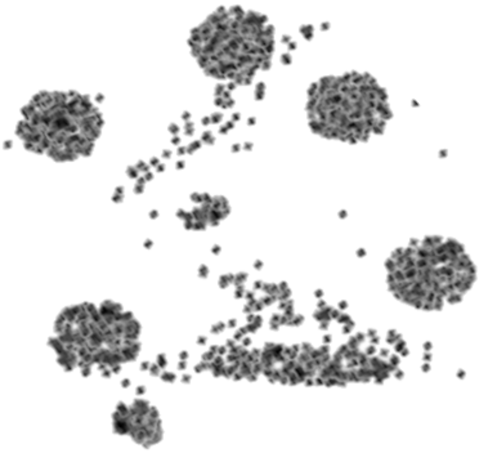

Figure 1.10: Sampled 3D point pattern; cutout in a thin slice of the cube $[-a/2, a/2]^3$

1.5.2 Choice of classification categories

In this section we show how the general framework of random forests described in Section 1.3 can be applied in the point process setting of Section 1.5.1. Ultimately our goal is to classify the clusters created in the identification step either as ball-shaped (we call this class *ball*) or elongated ellipsoidal clusters (called *ellipsoid*). However, even when using the generalized single-linkage algorithm some errors occur already at the stage of cluster identification and we have to take this problem into account in the classification step. In general, we cannot say much on the shape of these erratically identified clusters, but we may at least extract two subclasses that are easy to identify. If we have a set of points that was returned as a cluster by the clustering algorithm but that is neither an element of *ball* nor of *ellipsoid* then we define its category as follows.

1. If it contains more than a prespecified number of elements κ_{large} then we define its category to be *errLarge*.
2. If it contains less than a prespecified number of elements κ_{small} then we define its category to be *errSmall*.
3. If none of the above applies we define its category to be *errRemainder*.

This subdivision of the error categories makes it possible to identify more easily at least those cases where several dense clusters in the model step are merged into only one cluster in the identification step or if one of the sparse clusters in the model step is subdivided into several smaller clusters in the identification step. In our Monte-Carlo example we choose $\kappa_{small} = 30$ and $\kappa_{large} = 350$.

1.5.3 Choice of properties

Finally let us discuss the set of predictors used to assign clusters to the categories defined in Section 1.5.2. To identify at least some of the obvious misidentifications by the single-linkage algorithms considered in this chapter, a useful predictor is the size (in terms of number of elements) of a cluster. We denote this quantity by *nPoints* in the following. In order to distinguish between the categories *ball* and *ellipsoid* we need predictors related to the geometry of a cluster. We may make use of the fact that in elongated clusters the variance of the cluster points is particularly pronounced in one direction. Quantitatively this feature may be captured by principal component analysis on the three spatial coordinates of the points. Indeed for elongated clusters the largest eigenvalue of the empirical covariance matrix tends out to be much larger than the second-largest one. On the contrary, for ball-shaped clusters we would expect the eigenvalues to be of similar size. Therefore we include predictors η_1 and η_2 which are defined as the quotient of the largest and the second-largest eigenvalue and the quotient of the second-largest and the smallest eigenvalue, respectively.

1.5.4 Validation of cluster identification

Naturally it is desirable that the quality of the cluster analysis is robust with respect to moderate changes of the parameters. We comment on this issue in detail in Section 1.5.5 below, but nevertheless let us summarize the main results at already this point. For all parameter constellations of the point-process model we have considered, the generalized single-linkage algorithm with the splitting rule given in Example 5 of Section 1.2 had a better performance than the standard single-linkage algorithm. Since we only considered performance under optimal parameters of the single-linkage algorithm for cluster identification and since the standard single-linkage algorithm can be seen as a special case of the generalized single-linkage algorithm, this is not surprising. When increasing the ratio between the intensities of dense and sparse clusters, the advantage of the generalized single-linkage algorithm on the standard single-linkage algorithm becomes less significant, but both improve with respect to accuracy. Increasing the noise has a greater impact on the performance of the generalized single-linkage algorithm than on the standard single-linkage algorithm. Finally increasing the intensity of cluster centers deteriorates the quality of both the standard as well as the generalized single-linkage algorithm. Both cluster identification algorithms were relatively robust with respect to changes in the cluster shape.

The first validation step is to compare the quality of cluster identification of the generalized single-linkage algorithm to that of the standard single-linkage algorithm. This is achieved by first generating $n = 1000$ realizations of the point-process model and computing the fraction of clusters generated by the point-process model which are detected by each of the algorithms. For both the standard and the modified single-linkage algorithm the parameters of the algorithm were optimized using the Nelder-Mead method, see [2]. Note that the Nelder-Mead method is a popular simplex-based method for heuristically finding maxima/minima of multivariate functions. An advantage of the Nelder-Mead method in contrast to other classical optimization techniques such as gradient descent is due the fact that no knowledge on the derivatives of the objective function is needed. Let us elaborate on the optimization procedure in our specific setting. Denote by $D_\ell = \{D_{\ell,1}, \ldots, D_{\ell,m_\ell}\}$ and $S_\ell = \{S_{\ell,1}, \ldots, S_{\ell,m'_\ell}\}$ the family of dense respectively sparse clusters that appear in the ℓ-th realization of the point process. We say that the algorithm correctly identified a dense cluster $D_{\ell,j}$ if there exists a purported cluster Ξ with $\max(|\Xi \setminus D_{\ell,j}|, |D_{\ell,j} \setminus \Xi|) \leq 30$. Similarly, we say that the algorithm correctly detects a sparse cluster $S_{\ell,j}$ if there is a set of points Ξ which is deemed to be a cluster by the identification algorithm and such that $\max(|\Xi \setminus S_{\ell,j}|, |S_{\ell,j} \setminus \Xi|) \leq 5$ holds. The bounds 30 and 5 correspond to 10% of the expected number of elements in a dense respectively sparse cluster. Observe that we chose different bounds for dense and sparse clusters, since in our application it is appropriate to use a threshold relative to the cluster size. Indeed, our final goal is to determine the shape of various clusters and not to determine the membership of a single point to one of these clusters.

In Figure 1.11 the identification of clusters in a sampled 3D point pattern is visualized using the splitting rule of Example 5 in Section 1.2, where we also show an example of a misspecified cluster.

Let ρ_d denote the ratio of the number of correctly identified dense clusters (accumulated over the n iterations) by the overall number of simulated dense clusters. Similarly we define ρ_s. We want to compare the quality of the two clustering methods under optimal choice of their parameters α and $\alpha_1, \alpha_2, n_0, n_1$, respectively, where we use the Nelder-Mead search for the parameters maximizing the sum $\rho_s + \rho_d$. These parameters were introduced in Example 5 of Section 1.2. For the ordinary single-linkage algorithm we obtain an optimal threshold of

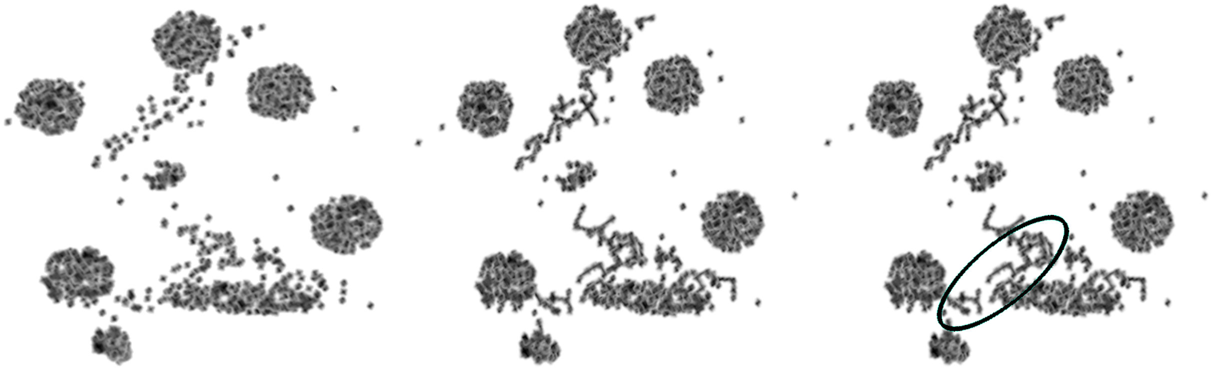

Figure 1.11: Identification of clusters in a sampled 3D point pattern; cutout in a thin slice of the cube $[-a/2, a/2]^3$

$\alpha = 10.66$ while for the generalized single-linkage algorithm the optimizer yields the values $\alpha_1 = 6.06$, $\alpha_2 = 16.14$, $n_0 = 10$ and $n_1 = 50$. Table 1.2 shows the fractions of clusters detected in the identification step using either the standard or generalized single-linkage algorithm.

Observe that the α_2-value for the generalized single-linkage algorithm is much larger than the α-value of the standard single-linkage algorithm. This is reasonable, as decreasing the α-value is the only possibility to distinguish between a larger number of clusters, while in the generalized single-linkage algorithm α_2 can be chosen much higher as we have the additional possibility to delete an edge between two clusters if one of the subgraphs $g_{1,\ell}$ or $g_{2,\ell}$ is sufficiently large. We also note that the generalized single-linkage algorithm considerably improves the proportion both of detected dense and sparse clusters when compared to the standard single-linkage algorithm.

Table 1.2: Proportion of clusters detected in the identification step

algorithm	dense clusters	sparse clusters	total
standard single-linkage	75.7%	52.4%	63.9%
generalized single-linkage	91.5%	79.5%	85.6%

1.5.5 Robustness of identification quality

Let us conclude by discussing the robustness of the quality concerning the presented clustering algorithms when changing the parameters of the point-process model. For all configurations of model parameters, the parameters of the single-linkage algorithms were optimized by the Nelder-Mead method. After that $n = 1000$ realizations of the point-process model were generated and the numbers of correctly identified clusters were computed. As a first step, one may vary the ratio between the intensities of dense and sparse clusters. If we put $p_6 = 0.99$, so that only 1% of all clusters are sparse, then the standard single-linkage algorithm correctly identifies 84.9% of all clusters while the generalized single-linkage algorithm correctly identifies 86.1% of all clusters. On the other hand if we put $p_6 = 0.01$, so that only 1% of all clusters are dense, then the standard single-linkage algorithm correctly identifies only 68.4% of all clusters while the generalized single-linkage algorithm manages to identify correctly a fraction of 77.4% of all clusters. In a second step, one may change the half-axes of the ellipsoid K_0. If we shrink the size of the largest half-axis by a factor of 2 while increasing the lengths of the other half-axes by a factor of $\sqrt{2}$, then the standard single-linkage algorithm correctly identifies 79.7% of all clusters while the generalized single-linkage algorithm correctly identifies a fraction of 92.1% of all clusters. In a third step, one may change the intensity of the noise, e.g. we may choose $\lambda_2 = 2.5 \cdot 10^{-5}$ (i.e. double the original value). Then the standard single-linkage algorithm correctly identifies 64.0% of all clusters while the generalized single-linkage algorithm correctly identifies a fraction of 82.5% of all clusters. Finally we can increase the intensity of cluster centers, e.g. let us double the value of λ_1 so that $\lambda_1 = 4.5 \cdot 10^{-6}$. Then the standard single-linkage algorithm correctly identifies 54.3% of all clusters while the generalized single-linkage algorithm correctly identifies a fraction of 76.5% of all clusters.

1.5.6 Validation of cluster classification

We now describe the numerical results of the cluster classification based on random forests. As discussed in Section 1.5.2 each cluster is assigned to precisely one of the five categories *ball, ellipsoid, errLarge, errSmall, errRemainder.* Furthermore we use the three predictors *nPoints,* η_1 and η_2 introduced in Section 1.5.3. We consider the classification of clusters obtained in 1000 realizations of the point-process model and use a random forest consisting of 500 decision trees. All clusters consisting of less than four points were discarded, since due to the choice of our intensities the probability that these points constitute the realization of a cluster from the original point process practically vanishes. In this way, we can greatly reduce the number of clusters to be classified. Table 1.3 shows the results obtained for the out-of-bag estimators for the classification error for the five categories under consideration. These estimators were introduced in Section 1.3. In particular, we see that the classification error for the four classes *ball*, *ellipsoid*, *errLarge*, *errSmall* is very low, while for the category *errRemainder* it is rather large. This is intuitively reasonable. Indeed the predictor *nPoints* is very well suited to discriminate between the sets of categories $\{errSmall\}$, $\{errLarge\}$ and $\{ball, ellipsoid, errRemainder\}$. Similarly, due to the simple geometry of the shapes considered in our point-process model the predictor η_1 is appropriate to distinguish between *ball* and *ellipsoid*. However, as the class *errRemainder* may contain a variety of shapes, it is also not surprising that our simple set of predictors has problems to discriminate between the classes *ball* or *ellipsoid* on the one hand and *errRemainder* on the other hand. These observations are reflected in the importance scores obtained from the out-of-bag sample, see Table 1.4. Furthermore, we see that the importance score for the predictor η_2 is rather low when compared to η_1 or *nPoints* (which is also in accordance with intuition). The notion of the importance score was introduced in Section 1.3.3.

Table 1.3: Out-of-bag estimators for the classification error

	ball	*ellipsoid*	*errSmall*	*errLarge*	*errRemainder*	*total*
oob-estimate	0.54%	2.07%	0.00%	1.36%	28.47%	3.12%

Table 1.4: Importance scores

predictor	*ball*	*ellipsoid*	*errSmall*	*errLarge*	*errRemainder*	*total*
nPoints	0.11	0.096	0.87	0.85	0.19	0.20
η_1	0.49	0.58	0.10	0.18	0.32	0.46
η_2	0.088	0.054	0.021	0.19	0.027	0.065

References

[1] D. Allard and C. Fraley. Nonparametric maximum likelihood estimation of features in spatial point processes using Voronoi tessellation. *Journal of the American Statistical Association*, 92:1485–1493, 1997.

[2] M. Avriel. *Nonlinear programming: Analysis and methods.* Dover Publishing, New York, 2003.

[3] F. Baccelli and S. Zuyev. Poisson-Voronoi spanning trees with applications to the optimization of communication networks. *Operations Research*, 47:619–631, 1999.

[4] S. Bandyopadhyay and E.J. Coyle. Minimizing communication costs in hierarchically-clustered networks of wireless sensors. *Computer Networks*, 44:1–16, 2004.

[5] J.D. Banfield and A.E. Raftery. Model-based Gaussian and non-Gaussian clustering. *Biometrics*, 49:803–821, 1993.

[6] L. Breiman. Better subset regression using the nonnegative Garrote. *Technometrics*, 37:373–384, 1995.

[7] L. Breiman. Random forests. *Machine Learning*, 45:5–32, 2001.

[8] T.J. Brereton, O. Stenzel, B. Baumeier, D.P. Kroese, and V. Schmidt. Efficient simulation of charge transport in deep-trap media. In *Proceedings of the Winter Simulation Conference*, Berlin, 2012 (to appear).

[9] S. Byers and A.E. Raftery. Nearest-neighbor clutter removal for estimating features in spatial point processes. *Journal of the American Statistical Association*, 93:577–584, 1998.

[10] D. Ciullo, G.D. Celik, and E. Modiano. Minimizing transmission energy in sensor networks via trajectory control. In *Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt), 2010 Proceedings of the 8th International Symposium on*, pages 132–141. IEEE, 2010.

[11] A. Dasgupta and A.E. Raftery. Detecting features in spatial point processes with clutter via model-based clustering. *Journal of the American Statistical Association*, 93:294–302, 1998.

[12] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern classification.* Wiley-Interscience, New York, second edition, 2001.

[13] M. Faessel and D. Jeulin. Segmentation of 3D microtomographic images of granular

materials with the stochastic watershed. *Journal of Microscopy*, 239:17–31, 2010.

[14] S. Foss and S. Zuyev. On a Voronoi aggregative process related to a bivariate Poisson process. *Advances in Applied Probability*, 28:965–981, 1996.

[15] C. Fraley and A.E. Raftery. How many clusters? which clustering method? answers via model-based cluster analysis. *The Computer Journal*, 41:578–588, 1998.

[16] G. Gaiselmann, M. Neumann, A. Spettl, M. Prestat, T. Hocker, L. Holzer, and V. Schmidt. Stochastic 3D modeling of $La_{0.6}Sr_{0.4}CoO_{3-\delta}$ cathodes based on structural segmentation of FIB-SEM images. *Computational Materials Science*, 2012 (to appear).

[17] M. Goldberg and S. Shlien. A clustering scheme for multispectral images. *IEEE Transactions on Systems, Man and Cybernetics*, 8:86–92, 1978.

[18] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning*. Springer Series in Statistics. Springer, New York, second edition, 2009. Data mining, inference, and prediction.

[19] C. Hirsch, D. Neuhäuser, and V. Schmidt. Percolation of graphs related to minimal spanning forests. (working paper).

[20] D.J. Illian, P.A. Penttinen, H. Stoyan, and D. Stoyan. *Statistical Analysis and Modelling of Spatial Point Patterns*. Wiley-Interscience, Chichester, 2008.

[21] A. Liaw and M. Wiener. Classification and regression by randomforest. *R News*, 2:18–22, 2002.

[22] P. Martin, T. Felka, U. Stoeckle, L. Handl, W.K. Aicher, V. Schmidt, and B. Rolauffs. Three-dimensional spatial organization of superficial chondrocytes: comparison of human knee articular cartilage and autologous chondrocyte implantation. (working paper).

[23] J. Mateu, G. Lorenzo, and E. Porcu. Detecting features in spatial point processes with clutter via local indicators of spatial association. *Journal of Computational and Graphical Statistics*, 16:968–990, 2007.

[24] J. Mateu, G. Lorenzo, and E. Porcu. Features detection in spatial point processes via multivariate techniques. *Environmetrics*, 21:400–414, 2010.

[25] R. Meesters and R. Roy. *Continuum Percolation*. Cambridge University Press, Cambridge, 1996.

[26] M. Meinhardt, S. Lück, P. Martin, T. Felka, W. Aicher, B. Rolauffs, and V. Schmidt.

Modeling chondrocyte patterns by elliptical cluster processes. *Journal of Structural Biology*, 177:447–458, 2012.

[27] Nicolai Meinshausen. Forest Garrote. *Electron. J. Stat.*, 3:1288–1304, 2009.

[28] F. Murtagh. A survey of algorithms for contiguity-constrained clustering and related problems. *The Computer Journal*, 28:82–88, 1985.

[29] F. Murtagh, D. Barreto, and J. Marcello. Bayesian segmentation and clustering for determining cloud mask images. In *Proceedings of the SPIE*, volume 4877, pages 144–155, 2002.

[30] F. Murtagh, C. Donalek, G. Longo, and R. Tagliaferri. Bayesian model selection for spatial clustering in 3D surveys. In *Proceedings of the SPIE*, volume 4847, pages 391–401, 2002.

[31] M. Penrose. *Random Geometric Graphs.* Oxford University Press, Oxford, 2003.

[32] F.J. Rohlf. Algorithm 76: hierarchical clustering using the minimum spanning tree. *The Computer Journal*, 16:93–95, 1973.

[33] B. Rolauffs, J.M. Williams, M. Aurich, A.J. Grodzinsky, K.E. Kuettner, and A.A. Cole. Proliferative remodeling of the spatial organization of human superficial chondrocytes distant from focal early osteoarthritis. *Arthritis & Rheumatism*, 62:489–498, 2010.

[34] B. Rolauffs, J.M. Williams, A.J. Grodzinsky, K.E. Kuettner, and A.A. Cole. Distinct horizontal patterns in the spatial organization of superficial zone chondrocytes of human joints. *Journal of Structural Biology*, 162:335–344, 2008.

[35] R. Thiedmann, A. Spettl, O. Stenzel, T. Zeibig, J.C. Hindson, Z. Saghi, N.C. Greenham, P.A. Midgley, and V. Schmidt. Networks of nanoparticles in organic-inorganic composites: algorithmic extraction and statistical analysis. *Image Analysis and Stereology*, 31:23–42, 2012.

[36] C.T. Zahn. Graph-theoretical methods for detecting and describing gestalt clusters. *IEEE Transactions on Computers*, 100:68–86, 1971.