Efficient Mining of Correlation Patterns in Spatial Point Data

Marko Salmenkivi

Helsinki Institute for Information Technology, Basic Research Unit
Department of Computer Science, P.O. Box 68, FI-University of Helsinki, Finland
`marko.salmenkivi@cs.helsinki.fi`

Abstract. We address the problem of analyzing spatial correlation between event types in large point data sets. Collocation rules are unsatisfactory, when confidence is not a sufficiently accurate interestingness measure, and Monte Carlo testing is infeasible, when the number of event types is large. We introduce an algorithm for mining correlation patterns, based on a non-parametric bootstrap test that, however, avoids the actual resampling by scanning each point and its distances to the events in the neighbourhood. As a real data set we analyze a large place name data set, the set of event types consisting of different linguistic features that appear in the place names. Experimental results show that the algorithm can be applied to large data sets with hundreds of event types.

1 Introduction

Consider a large set of spatial point objects. Each object may be an instance (*event*) of zero, one, or several event types of interest, and the number of event types is large (e.g, 100-2000). We are interested in analyzing spatial correlation between the point patterns of different event types.

As an example of real data, Fig. 1 illustrates the variation of frequency of place names in East Finland in a corpus of the National Land Survey of Finland. A small dot is plotted at all named locations. For illustration purposes, the events of three linguistic features occurring in the place names are also indicated. The indicated event types are some of the lexical features that are of interest in the study of the settlement in East Finland during the Iron Age, and the ancient interaction between the Finns and the Saami in the region.

These features are just examples. As a very large set of different lexemes, and other linguistic features, appear in place names, the number of event types may be large, hundreds, or even more.

Fig. 1 shows that the overall frequency of place names is not constant across the area. We see, for instance, that lakes are discerned from the surroundings with lower frequencies of events, whereas on the shores the density of events is high. Instances of event types may correlate simply because they both occur more frequently in the regions of high overall intensity, and less frequently in the regions of low intensity. Thus, to obtain reliable results, we desire to relate the interestingness of an observed correlation to the overall frequency of objects.

There are no methods available that can do with a lot of data and more than a few event types. Collocation rules only consider the plain frequencies of nearby instances of event types in each rule, ignoring the overall frequency of events. They can include several event types, but in practice the algorithms may be very slow if the number of event types is large [9]. In spatial statistics Monte Carlo tests are usually employed. They are computationally infeasible when the number of event types is large.

Fig. 1. Locations of place names (very small dots) in East Finland in the corpus of the National Land Survey. Instances of three event types: 1) unfilled (blue) triangles = name elements assumed to be of Saami origin; 2) filled (red) triangles = instances of lexemes *akka, akko*, 'old woman'; 3) circles = instances of lexeme *louhi* (the female ruler of *Pohjola* in *Kalevala*, the Finnish national epic.)

We propose a method that can be seen as a non-parametric bootstrap test that, however, avoids the actual resampling from the data by scanning each point and its distances to the points in the neighbourhood in turn. A preliminary version of the test, using Monte Carlo methods, was outlined in [6]. In this paper we extend the idea to large sets of event types by introducing a novel algorithm that computes the tests without actual resampling. Theoretical complexity of the algorithm is $\mathcal{O}(n^2)$, where n is the number of events. In practice, a small fraction of distances between points has to be evaluated, and our experiments show that the algorithm can evaluate correlations in large data sets and event type sets in feasible time. The results provide more detailed knowledge than

collocation rules, since the distribution of all points is considered when assessing how interesting individual rules are.

This paper is organized as follows. In Section 2 the related work is introduced. In Section 3 the notion of correlation pattern is proposed for measuring the significance of two-feature collocation rules. A new algorithm for finding such patterns is introduced in Section 4. Experiment results on real and synthetic data sets are provided in Section 5. Section 6 is a conclusion.

2 Related Work

Collocation rules are often used to describe dependencies in spatial data. Collocation rule mining algorithms are typically based on first finding interesting *collocation patterns*, and then extracting rules from them. A neighbourhood relation R of spatial objects is assumed to be explicitly given. In case of points, the definition is usually based on the Euclidean distance of at most a predefined ϵ . Let $\mathcal{E}$ be a set of features (or *event types*). A collocation pattern is defined as a set $\mathcal{Q} \subseteq \mathcal{E}$ [3,4,8,9].

Further let $S = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$ be a set of spatial point objects. The type of data considered in this paper can be represented as a binary $S \times \mathcal{E}$ -matrix. We say that $p \in S$ is an instance (or *event*) of event type $A \in \mathcal{E}$ in data D if $D_{p,A} = 1$. A set S' of spatial objects is a (*row*) *instance* of $\mathcal{Q}$ if for all $o_i, o_j \in S'$, $\text{dist}(o_i, o_j) \leq \epsilon$, and S' contains instances of all the event types in $\mathcal{Q}$, and no proper subset of S' does so [3].

Let $\mathcal{P}$ and $\mathcal{Q}$ be collocation patterns, and $\mathcal{P} \cap \mathcal{Q} = \emptyset$. Then $\mathcal{P} \rightarrow \mathcal{Q}$ is a *collocation rule*. The *confidence* of $\mathcal{P} \rightarrow \mathcal{Q}$ in data D is the fraction of instances of $\mathcal{P}$ such that they are also instances of $\mathcal{P} \cup \mathcal{Q}$.

Interestingness measures for collocation patterns include *prevalence*, which is defined as $\min\{pr(F, \mathcal{P}), F \in \mathcal{P}\}$, and *maximum participation ratio* (MPR), $\max\{pr(F, \mathcal{P}), F \in \mathcal{P}\}$, where $pr(F, \mathcal{P})$ is the proportion of the events of type F that appear in the instances of $\mathcal{P}$. A high prevalence indicates that $\mathcal{P}$ can be used to generate confident collocation rules [8]. Correspondingly, a high MPR implies that at least one of the event types, denote it by T , rarely occurs outside $\mathcal{P}$. Hence, the collocation rule $\{T\} \rightarrow \mathcal{P} \setminus \{T\}$ is likely to be interesting [4].

Levelwise search algorithms for finding all interesting patterns were introduced by [4,8]. The algorithm proposed by Zhang et al. finds for each object in turn the maximal pattern instance in which the object participates [9].

Let $Z \subset \mathbf{R}^2$ be the observation region under investigation. In spatial statistics point patterns are typically modeled as point processes $\{X(\mathbf{t}) : \mathbf{t} \in Z\}$ which generate events $\mathbf{t}$ on Z . If there are k event types, $X(\mathbf{t})$ is a k -dimensional binary vector indicating the event types whose instance point $\mathbf{t}$ is. Hypothesis testing selects a point process for modeling the null hypothesis H_0 of no spatial correlation, and a test statistic T measuring the correlation in an observed point pattern. The simplest choice is the Poisson process conditioned on the numbers of events of the event types to be tested (complete spatial randomness, CSR). In practice, Monte Carlo methods are usually needed to simulate the selected process to obtain an approximation for the sample distribution of T [1,2].

Intensity function of a point process intuitively describes the average frequency of events in a unit area. It is defined as $\lambda(\mathbf{s}) = \lim_{|ds| \rightarrow 0} E(Y(ds))/|ds|$, where ds is a “small” region around $\mathbf{s} \in Z$, $|ds|$ is the area of ds , and $Y(ds)$ is the random variable indicating the number of events in ds . For the number of instances in $Z' \subseteq Z$, it holds that $E(Y(Z')) = \int_{Z'} \lambda(\mathbf{s}) ds$.

Leino et al. study significances of two-feature collocation rules in lake name data, where different lake names form the set of event types. Their approach takes the overall frequency of lakes into account, but unrealistically assumes the probability of an instance of each event type to be constant across Z [5].

3 Testing Collocation Rules

We next consider collocation rules of type $A \rightarrow B$, where A and B are single event types, in the context of hypothesis testing. The notation is summarized in Table 1.

Let us assume that the data S (or $S \times \mathcal{E}$) is a sample from population F . It is natural to investigate a test statistic based on the definition of confidence of a collocation rule. Accordingly, let us denote the conditional probability of observing an instance of B within a distance $\leq \epsilon$ from a given instance of A by $G_{A,B}(\epsilon)$, and its observed value, which is the confidence of $A \rightarrow B$, in data by $g_{A,B}(\epsilon)$. We define a *correlation pattern* as a statistically significant collocation rule $A \rightarrow B$ as follows:

Definition 1. Let $\{X(s) : s \in Z\}$ be a bivariate spatial point process under a null hypothesis H_0 , that is, it specifies the probability distribution of point patterns of two event types, denote them by A and B , under H_0 . A collocation rule $A \rightarrow B$ is a correlation pattern in data $S \times \mathcal{E}$ with significance level α iff $\text{Prob}(G_{A,B}(\epsilon) \leq g_{A,B}(\epsilon) | X(s)) \leq \alpha$.

For practical purposes, the general definition is, of course, useful only if a computationally feasible H_0 can be found, and it describes in a meaningful way the vague concept of “no spatial correlation” between A and B . Below we devise methods to solve this key problem.

The CSR null hypothesis that assumes a constant intensity of the generating process everywhere in Z is too simple in practice. Assuming that, in addition to events of types A and B , a lot of point data is available, the following bootstrap test may be a better alternative: sample from S to obtain A^* and B^* , and compare $g_{A,B}(\epsilon)$ against the obtained distribution of $\hat{G}_{A,B}(\epsilon) | H_0$. Now, the intensity of the generating process under H_0 is not assumed to be constant but a bootstrap estimate $\lambda_{\hat{F}}(\mathbf{s})$ of λ in population F .

This testing procedure works, if the intensities λ_A and λ_B can be assumed to be proportional to λ_F . Unfortunately, this assumption is often not valid either. For the same reason permutation testing of the whole data is not meaningful. All event types do not usually occur in the whole area of investigation, and they may be rare somewhere, and common somewhere else, not necessarily following

Table 1. Notation

Symbol	Explanation
$\mathcal{E} = \{A, B, \dots\}$	set of event types
S	set of all spatial point objects in current data
F	population from which S is a sample
$\mathcal{A}, \mathcal{B}, \dots$	sets of events of event type A, B etc.
$G_{A,B}(\epsilon)$	conditional prob. in F : given $a \in \mathcal{A}, \exists b \in \mathcal{B}, s.t. dist(a, b) \leq \epsilon$.
$g_{A,B}(\epsilon)$	observed value of $G_{A,B}(\epsilon)$ in current data
$\lambda_A(\mathbf{s})$	intensity of the process (at location $\mathbf{s}$) that generated $\mathcal{A}$
$\mathcal{A}^*$	pseudosample of A (number of events equal to $ \mathcal{A} $)

the variation of frequency of all objects (see, e.g., the event type indicated by unfilled triangles in Fig. 1).

Thus, to be realistic H_0 should somehow depend on the spatial variation of frequencies of A and B , of course in a carefully controlled way. Accordingly, consider generating events independently for A^* and B^* from the processes whose intensities are kernel density estimates $\hat{\lambda}_A(\mathbf{s})$ and $\hat{\lambda}_B(\mathbf{s})$. A kernel density estimate $\hat{\lambda}_A$ is obtained by inserting, e.g., a bivariate Gaussian f , around each $a_i \in \mathcal{A}$, and $\hat{\lambda}_A(\mathbf{s}) = \sum_{i=1}^n \frac{1}{h^2} f(\frac{\mathbf{s}-a_i}{h})$. The parameter h (*bandwidth*) controls the degree of smoothing.

When $h \rightarrow \infty$, this corresponds to sampling under the CSR. On the other hand, when $h \rightarrow 0$ the procedure converges to the resampling from sets $\mathcal{A}$ and $\mathcal{B}$. This is the regular bootstrap approach for parameter estimation, but it is not a valid approach for testing, since the resamples are not drawn under H_0 . However, by conducting trials with several h we can study the value of the test statistic as the function of h , that is, the impact of “weighting” the CSR hypothesis by $\hat{\lambda}_A(\mathbf{s})$ and $\hat{\lambda}_B(\mathbf{s})$.

While this method is interesting, it still investigates only the events of types A and B , ignoring the information available in the distribution of S . A simple modification that yields a better solution is as follows. Consider sampling from S to obtain A^* and B^* . Let us weight each point $\mathbf{t} \in S$ by $\hat{\lambda}_A(\mathbf{t})$ (or $\hat{\lambda}_B(\mathbf{t})$). When $h \rightarrow \infty$, the same weight is assigned to all points, resulting in plain sampling from S . Similarly to the previous case, when $h \rightarrow 0$ the procedure converges to resampling from sets $\mathcal{A}$ and $\mathcal{B}$.

Fig. 2 illustrates this in case of two event types in the place name data (lexemes *musta* ‘black’ and *valkoinen* ‘white’). The indicated different bandwidths correspond to different weightings of the plain sampling from S , which is titled “no weight”. The errorbars show the 99 % intervals of $\hat{G}_{A,B}(\epsilon)|H_0$. The results show that the observed $g_{A,B}(1 \text{ km}) = 275$ cannot be obtained even by the strongly weighted null hypotheses ($h = 0.5 \text{ km}$), and, hence, the rule $A \rightarrow B$ is clearly very significant. The plain sampling from S indicates remarkable deviation from H_0 , when the neighbourhood is extended to 4 km, whereas $g_{A,B}(4 \text{ km})$ is included in the 99 %-intervals of more realistic weighted versions.

Fig. 2. Significance of rule $A \rightarrow B$ with different neighbourhood definitions (distance in y -axes), and weighted versions of H_0 (bandwidth h of Gaussian kernel). Errorbars and circles indicate 99 %-intervals of $\hat{G}_{A,B}(\text{distance})|H_0$, and $g_{A,B}(\text{distance})$, respectively.

All the described testing procedures rely on Monte Carlo methods. We next propose a method that avoids them. Consider the previous approach, but fix the events in $\mathcal{B}$, and only draw pseudosamples A^* from S . If $|S|$ is small, or no reference data is available, i.e. $S = \mathcal{A} \cup \mathcal{B}$, then it is easy to see that $E(\hat{G}_{A,B}|H_0) > g_{A,B}$, i.e., the pseudosamples are on average closer to some $b \in \mathcal{B}$ than the original data, and thus, the estimator is biased. Obviously, when more reference data are available, the dependence becomes weaker, and in the limit, when $|S| \rightarrow \infty$, $|\mathcal{A}|/|S| \rightarrow 0$, and $|\mathcal{B}|/|S| \rightarrow 0$, then $E(\hat{G}_{A,B}|H_0) \rightarrow G_{A,B}|H_0$. Thus, the estimator is asymptotically unbiased. Results on synthetic data in Sec. 5.1 demonstrate this.

Conditioning the method on one of the event types makes it possible to develop an efficient algorithm for computing the significant associations without actually generating the pseudosamples. The assumptions needed are, accordingly, that S is “large”, and the proportions $|\mathcal{A}|/|S|$, $|\mathcal{B}|/|S|$ of individual event types to be tested are “small”.

Let us again consider rule $A \rightarrow B$. Now, it holds, under H_0 , that $G_{A,B}(\epsilon) \sim \text{Bin}(|\mathcal{A}|, p_{A,B})$, where $p_{A,B}$ is the probability that during the resampling for A^* an event that is within ϵ from an event of type B is selected. The probability p is determined by the kernel estimates $\hat{\lambda}_A(\mathbf{t})$, $\mathbf{t} \in S$.

Denote by $S^* \subseteq S$ the set of events such that for each $x \in S^*$, $\text{dist}(x, b) \leq \epsilon$ for some $b \in \mathcal{B}$. Then,

$$p_{A,B} = \frac{\sum_{\mathbf{x} \in S^*} \hat{\lambda}_A(\mathbf{x})}{\sum_{\mathbf{y} \in S} \hat{\lambda}_A(\mathbf{y})} = \frac{\sum_{\mathbf{x} \in S^*} \sum_{\mathbf{a}_i \in \mathcal{A}} f\left(\frac{\mathbf{x} - \mathbf{a}_i}{h}\right)}{\sum_{\mathbf{y} \in S} \sum_{\mathbf{a}_i \in \mathcal{A}} f\left(\frac{\mathbf{y} - \mathbf{a}_i}{h}\right)}. \quad (1)$$

The sums in Eq. 1 can be approximately computed by inspecting the occurrences “sufficiently” close to the events in $\mathcal{A} \cup \mathcal{B}$. The significance of the observed number of co-occurrences can then be evaluated by comparing $g_{A,B}(\epsilon)$ with the binomial distribution of $G_{A,B}|H_0$ (e.g., by applying normal approximation).

4 Algorithm for Finding Associations

In this section we introduce an algorithm that computes the significances of correlation patterns between the instances of pairs of event types.

A useful kernel function decreases monotonically as a function of distance from its mode. For instance, 99.7% of the probability mass of a bivariate Gaussian kernel function f with bandwidth h is concentrated inside a circle with radius $3h$. Hence, by setting, for example $\eta = 4h$, we can safely assume that $f(\eta) \approx 0$, and ignore the events that are not within η from the mode. When inspecting each $\mathbf{t} \in S$ to compute the approximations for $p_{X,Y}, (X,Y) \in \mathcal{E} \times \mathcal{E}$ in Eq. 1, we use this: the “influence” of $\mathbf{t}$ does not extend to any location $\mathbf{s}$ such that $\text{dist}(\mathbf{s}, \mathbf{t}) > \eta$.

Consider location $\mathbf{t}$ in the left panel of Fig. 3. The events of type A , and the closest event of type B (b_1) are shown. The events a_4, a_5 that are not within a distance of η from $\mathbf{t}$ are indicated by dashed edges. One should compute $\hat{\lambda}_A(\mathbf{t})$, i.e., to evaluate the kernel function three times to obtain $w = \sum_{i=1}^3 f(\frac{\mathbf{t}-\mathbf{a}_i}{h})$. This yields the contribution of $\mathbf{t}$ to the denominator of Eq. 1. Further, if the distance of $\mathbf{t}$ and b_1 is less than ϵ , w should also be included in the sum of the numerator in Eq. 1.

The right panel of Fig. 3 describes a more realistic setting, where there are several event types, and one should compute $p_{i,j}$ for each ordered pair of event types $(i,j) \in \mathcal{E} \times \mathcal{E}$. Now all the events within η from the central node $\mathbf{t}$ are shown: $a_2 \in \mathcal{A}, b_1, b_2 \in \mathcal{B}$, and $c_1, c_2 \in \mathcal{C}$. The events (b_1, c_2) within a distance less than ϵ from $\mathbf{t}$ are indicated by thick edges. Instead of only one pair of event types, we now have to find the sums for all pairs $(i,j) \in \{A,B,C\}, i \neq j$, according to exactly the same simple procedure as in the previous case.

The procedure can be generalized as follows. Let $\mathbf{t}$ be the location of the central object, and further let y be an instance of Y such that $\text{dist}(\mathbf{t}, y) \leq \epsilon$. If there are instances $x_1, x_2, \dots, x_k$ of X such that $\text{dist}(\mathbf{t}, x_i) < \eta, 1 \leq i \leq k$ then, for computation of $p_{X,Y}$ of Eq. 1, the term $\sum_{i=1}^k f(\frac{\mathbf{t}-x_i}{h})$ has to be included in the numerator of Eq. 1. If several event types are allowed to occur at one location, the sums for each event type are updated at each location.

Fig. 3. Star graphs for computing sums of kernel density estimates

Hence, consider the following algorithm for testing collocation rules. The objects are first sorted according to their x-coordinates in order to avoid computing distances between all objects (line 1). The y-coordinates of two objects are inspected only if their x-coordinates are sufficiently close to each other, and the

kernel function is evaluated only if the distance of objects $< \eta$ (lines 7–11). The main loop (lines 3–19) scans the η -neighbourhood of each object. Lines 4–6 initialize the list L of features with an instance within η from o_i , and the sums of weights of each event type. Each event type whose instance is closer than η to o_i is added to L in line 13. Indicator function $f_X(o_i)$ returns TRUE if o_i is an instance of event type X (lines 12–13). The sums of each pair of event types are updated in function **update_pairwise_weights**. The counter of real co-occurrences of event types X and Y is $g[X, Y]$. These counters are updated in line 18. After iterating all objects in S , the significances of $g[X, Y]$ are evaluated by comparing them to the binomial distributions of the test statistics.

Algorithm. **test_collocations** ($o_1, \dots, o_n \in S, \epsilon, \eta$, set of event types $\mathcal{E}$)

1. sort $o_1, \dots, o_n$ according to their x -coordinates;
2. $k \leftarrow 1$
3. **for** each $1 \leq i \leq n$ **do**
4. $L \leftarrow \emptyset$
5. **for** each $X \in \mathcal{E}$ **do**
6. $w_\epsilon[X] \leftarrow 0$; $w_\eta[X] \leftarrow 0$
7. **while** $o_i.x - o_k.x \geq \eta$ **do** $k \leftarrow k + 1$
8. $j \leftarrow k$
9. **while** $o_i.x - o_j.x < \eta$ **do**
10. **if** ($\text{dist}(o_i, o_j) < \eta$) **then**
11. $z = \text{weight}(o_i, o_j)$
12. **for** each $\{X \in \mathcal{E} | f_X(o_i) = \text{TRUE}\}$ **do** $W_X \leftarrow W_X + z$
13. **for** each $\{Y \in \mathcal{E} | f_Y(o_j) = \text{TRUE}\}$ **do** $L \leftarrow L \cup Y$
14. **if** ($\text{dist}(o_i, o_j) \leq \epsilon$) **then** $w_\epsilon[X] \leftarrow w_\epsilon[X] + z$
15. **else** $w_\eta[X] \leftarrow w_\eta[X] + z$
16. $j \leftarrow j + 1$
17. **for** each $Y \in L$ such that $w_\epsilon[Y] > 0$
18. $g[X, Y] \leftarrow g[X, Y] + 1$
19. **update_pairwise_weights**(L, w_ϵ, w_η)
20. **for** each ordered pair $(X, Y) \in \mathcal{E} \times \mathcal{E}$
21. compute significance of $g[X, Y]$ based on W_X and W_X^Y

Function **update_pairwise_weights**(L, w_ϵ, w_η)

1. **for** each $X \in L$
2. **for** each $Y \in L$
3. **if** $w_\epsilon[X] > 0$ **then** $W_X^Y \leftarrow W_X^Y + w_\epsilon[Y] + w_\eta[Y]$
4. **if** $w_\epsilon[Y] > 0$ **then** $W_X^Y \leftarrow W_X^Y + w_\epsilon[X] + w_\eta[X]$

In the worst case the algorithm has to compute the distances between all objects. Thus, the time complexity is $\mathcal{O}(n^2)$, where n is the number of objects. In practice, the number of potential nearby objects of an arbitrary object is a fraction of all objects. Further, the run time is influenced by the proportion of events and unlabelled locations, frequency of them, and bandwidth. We are conducting systematic experiments of the influence of different factors.

Fixing a distance of ϵ is inconvenient (demonstrated in Fig.2). Further, fixing bandwidth h corresponds to the problem that there is usually not a single correct

way of defining H_0 of "no correlation". Thus, it is, for data mining purposes, meaningful to study significance values as function of h (as in Fig.2). For clarity the algorithm was presented above in a form that takes as input only single values for ϵ and h . We extended it to inspect several ϵ_i and h_i in the same run, with a very small increase in computational cost, which is dominated by $\max(h_i)$.

If the data or data structures are too large to be stored in main memory, the hashing strategy introduced by [9] can easily be included as a preprocessing phase. The algorithm can be applied to each bucket separately, and the weights are finally summed up over all of them.

Though the analysis above used the probability $G_{A,B}(\epsilon)$ as the test statistic, the method and algorithm can be extended to the analysis of other test statistics. For instance, the so-called $K_{A,B}(\epsilon)$ -function considers the number of events within ϵ , instead of the nearest neighbour only.

When testing the significance of correlation between point patterns of two event types, spatial autocorrelation of event types may be a distorting factor. Though the selected H_0 assumes independence *between* the event types, it should allow dependence of events that are instances of the same event type. In our method, the binomial distribution of $G_{A,B}(\epsilon)|H_0$ was based on the assumption of independence of the samples. However, the distorting effect of possible autocorrelation is reduced by two facts. First, though the samples are independent, the autocorrelation is implicitly taken into account by weighting the samples according to the kernel density estimated intensities. A remarkable concentration of events (i.e., autocorrelation) increases the probability of selecting a sample from the neighbourhood. Second, the testing of rule $A \rightarrow B$ is carried out by fixing the locations of $b \in \mathcal{B}$. Thus, the possible autocorrelation in $\mathcal{B}$ is kept unchanged. We are currently studying the subtle problem of autocorrelation by conducting further experiments with synthetic data.

5 Experiments

5.1 Synthetic Data

We tested the algorithm on synthetic data with 100 event types, each having 1000 uniformly randomly generated events. Different numbers of unlabelled points from zero up to 700,000 were also generated, keeping the locations of the labelled events fixed in each trial. The observation region was a square of size $100,000 \times 100,000$. We studied the values of $g_{i,j}(\epsilon)$ for each pair $(i, j) \in \mathcal{E} \times \mathcal{E}$, and the significances of the deviation from the distribution of $G_{i,j}(\epsilon)|H_0$ the algorithm assigned to them.

Since the data are independently generated, the average deviation from H_0 should be zero. As described in Sec. 3, the estimates are biased with small S . When increasing the number of unlabelled points, the average tended to zero. Fig. 4 summarizes the results in the case that S included the maximum of 700,000 unlabelled points, and the 100,000 labelled events. There is a dot for each ordered pair $(i, j) \in \mathcal{E} \times \mathcal{E}$. The x -axis indicates the value $g_{i,j}(1000)$. The y -axis shows the significance of the deviation from the H_0 -distribution that our

Fig. 4. Trial on synthetic data (100 event types, each with 1000 randomly generated events, 700,000 unlabelled points). Each dot represents an ordered pair (i, j) of event types: $g_{i,j}(1000)$ (x-axes) vs. the significance of deviation of $g_{i,j}(1000)$ from $\hat{G}_{i,j}(1000)|H_0$ of no spatial correlation (y-axes) assigned by the proposed algorithm.

algorithm assigned to each $g_{i,j}(1000)$. The range $[-1.96, 1.96]$ corresponds to 95 % confidence interval. The kernel function was the bivariate Gaussian, $h = 5000$ (symmetric, $\rho = 0$). The average significance was very slightly biased (0.08). The running time (Pentium 1.33 GHz Linux) was 2 h 25 min. Another test run with a larger generated data (500 event types, 1000 events in each, resulting in 500,000 labelled events, and $|S| = 1,000,000$) took 5 1/2 hours.

Given a value of the x -coordinate, the range of values of the y -coordinate indicates the range of different assessments of significance, the value of $g_{i,j}$ being the same. This spread is due to the statistical error inherent in any bootstrap approach, caused by the difference of population F , and sample S . Though the intensity of the generating process is constant, random variation causes events of some event types to be located, on average, in regions of larger overall density, resulting in different significance assessments.

5.2 Real Data: Linguistic Analysis of Place Names

The Place Name Register maintained by the National Land Survey of Finland contains 717,746 Finnish place names, including the coordinates and types of the named locations. Each named object is represented by a pair of coordinates.

Linguistic features appearing in place names are of interest for many different research fields. Place names preserve features that have disappeared from the current language. Thus, they are significant for the research on, e.g., the history of languages. Loan words are signs of interaction between cultures, and provide material for the research on cultural history and history of settlement. As an example, Finnish place names of Saami origin give evidence for Saami inhabitants in South and Central Finland during the Iron Age.

In many languages, e.g., in German and Finnish, place names are typically compound words. A natural approach is to consider the different lexemes (words) as different event types, and to study their relationships. As a tedious preprocessing

Fig. 5. Events of name elements *raja* 'boundary' (left-side map), and *pelto* 'arable land' (right-side map). Summary of significances of rules $raja \rightarrow B$ (top), $pelto \rightarrow B$ (bottom), $\epsilon = 1.5$ km, where B is in turn each of the 100 most common name elements indicated by x -axis in decreasing order of the number of instances. Errorbars indicate the different bandwidths of Gaussian kernel functions: 1500, 5000, and 10,000 metres, the widest bandwidth indicated by points.

phase of the data, we first extracted the individual name elements from the compound names, and found the basic forms of the inflected words.

Since the endings and first parts of compound names have different semantic functions, it is useful to analyze them separately. In the following we only consider the first parts. We studied the pairwise correlation patterns between the name elements such that the number of instances as the first parts of compound names is at least 30. The number of event types was 1,707. The running time of the algorithm was 37 minutes (1.33 GHz Linux Pentium) when using the Gaussian kernel, $h = 5000$ metres. For a detailed description of the results of the analysis, see [7].

We illustrate the analysis of profiles of correlation patterns by an example. Fig. 5 displays the instances of two name elements: *raja*, meaning 'boundary', and *pelto*, meaning 'field' (arable land). Though both event types have approx. 2,500 instances occurring in the same (larger) areas, they differ remarkably in a local level. Instances of *raja*, unlike those of *pelto*, are not located in the immediate neighbourhood of the other common name elements as indicated by Fig. 5. In this case the meaning of *raja*, boundary, gives reason to the explanation that the instances are located apart from the heart of the (earlier) settlement, unlike those of *pelto*, referring to arable land.

6 Conclusion

The methods of spatial data mining and spatial statistics have been quite separated when it comes to finding associations in spatial point data and evaluating

their significances. While data mining approaches have concentrated on developing algorithms in the spirit of association rules and frequent patterns, the statisticians have been working with point process models leading to tedious simulations for evaluating attraction and repulsion patterns in data.

We develop an intermediate approach, and introduce an algorithm for evaluating associations of features in large spatial data sets with even thousands of features. Compared to collocation rules, the associations found by the developed algorithm provide more detailed knowledge, since they take characteristics of the whole data set into account when assessing the significance of an observed association.

We tested the methods on synthetic data, and a large place name data set. The different name elements appearing in compound words were extracted, and they were treated as features with a location. The number of name elements is large, and thus, novel methods are needed in the analysis.

References

1. A. C. Davison, D.V.Hinkley. *Bootstrap Methods and their Application*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 1997.
2. P. J. Diggle. *Statistical Analysis of Spatial Point Patterns*. Mathematics in Biology. Academic Press, London, 1983.
3. Y. Huang, S. Shekhar, and H. Xiong. Discovering Colocation Patterns from Spatial Data Sets: A General Approach. *IEEE Transactions on Knowledge and Data Engineering*, 16 (12), 1472–1485, December 2004.
4. Y. Huang, H. Xiong, S. Shekhar, and J. Pei. Mining confident co-location rules without a support threshold. In *Proc. 2003 ACM Symposium on Applied computing*, pages 497–501, Melbourne, Florida, 2003.
5. A. Leino, H. Mannila, and R. Pitkänen. Rule discovery and probabilistic modeling for onomastic data. N. Lavrac, D. Gamberger, L. Todorovski, H. Blockeel (eds.), *Knowledge Discovery in Databases: PKDD 2003*, 291–302. Lecture Notes in Artificial Intelligence 2838. Springer, 2003.
6. M. Salmenkivi. Evaluating attraction in spatial point patterns with an application in the field of cultural history. In *Proceedings of the Fourth IEEE International Conference on Data Mining (ICDM'04)*, pages 511–514. Brighton, UK, November 2004.
7. M. Salmenkivi, S. Hyvönen, A. Leino, and H. Tuominen. Computational survey of clustering in Finnish place name elements. *Proc. of 22nd International Conference on Onomastic Sciences, ICOS XXII*. Pisa, Italy, August–September 2005.
8. S. Shekhar and Y. Huang. Discovering spatial co-location patterns: a summary of results. In *Proceedings of 7th International Symposium on Advances in Spatial and Temporal Databases (SSTD 2001)*, Redondo Beach, CA, USA, 2001.
9. X. Zhang, N. Mamoulis, D. Cheung, Y. Shou. Fast mining of spatial collocations. In *Proc. 10th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 384–393, Seattle, Washington, 2004.