Computers, Environment and Urban Systems xxx (2009) xxx–xxx

Contents lists available at ScienceDirect

Computers, Environment and Urban Systems

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

Detecting feature from spatial point processes using Collective Nearest Neighbor

Tao Pei [a,*], A-Xing Zhu [a,b], Chenghu Zhou [a], Baolin Li [a], Chengzhi Qin [a]

[a] State Key Laboratory of Resources and Environmental Information System, Institute of Geographical Sciences and Natural Resources Research, CAS, 11A, Datun Road Anwai, Beijing 100101, China
[b] Department of Geography, University of Wisconsin Madison, 550N, Park Street, Madison, WI 53706-1491, USA

ARTICLE INFO

Article history:
Available online xxxx

Keywords:
Classification entropy
Noise
Point pattern
Feature
Spatial scan method
Shared nearest neighbor
EM algorithm

ABSTRACT

In a spatial point set, clustering patterns (features) are difficult to locate due to the presence of noise. Previous methods, either using grid-based method or distance-based method to separate feature from noise, suffer from the parameter choice problem, which may produce different point patterns in terms of shape and area. This paper presents the Collective Nearest Neighbor method (CLNN) to identify features. CLNN assumes that in spatial data clustered points and noise can be viewed as two homogenous point processes. The one with the higher intensity is considered as a feature and the one with the lower intensity is treated as noise. As a result, they can be separated according to the difference in intensity between them. With CLNN, points are first classified into feature and noise based on the kth nearest distance (the distance between a point and its kth nearest neighbor) at various values of k. Then, CLNN selects those classifications in which the separated classes (i.e. features and noise) are homogenous Poisson processes and cannot be further divided. Finally, CLNN identifies clustered points by averaging the selected classifications. Evaluation of CLNN using simulated data shows that CLNN reduces the number of false points significantly. The comparison between CLNN, the shared nearest neighbor, the spatial scan and the classification entropy method shows that CLNN produced the fewest false points. A case study using seismic data in southwestern China showed that CLNN is able to identify foreshocks of the Songpan earthquake (M = 7.2), which may help to locate the epicenter of the Songpan earthquake.

1. Introduction

Clustered point patterns are referred to as subgroups of points which are distributed at the higher density in constrained areas compared with those outside the areas (Ripley, 1987; Cressie, 1991 (Chapter 8)). Clustered points usually represent meaningful point patterns (i.e. features), especially in many different natural and social areas, such as epidemic diseases, foreshocks, or aftershocks of strong earthquakes, criminal behaviors, and vehicle crashes (Openshaw, 1996; Lawson, 2001 (chapter 1); Hodge & Austin, 2004; Chainey & Ratcliffe, 2005 (chapter 1); Pei, Zhu, Zhou, Li, & Qin, 2006; Yang & Lee, 2007; Yamada & Thill, 2007). The detection of clustered point patterns may help to predict the forthcoming natural or social events and to develop respective plans. Thus, the research on the detection of clustered point patterns has been recognized as an important area in the spatial data mining and knowledge discovery community.

Due to its importance, numerous methods have been proposed to enhance the analysis of clustered point patterns (i.e. feature). The core issue in the detection of feature is how to separate dense clusters from noise, which relies on the computation of local density. Most previous approaches estimated local density either with window-based methods or distance-based methods (Boots & Getis, 1988; Cressie, 1991). The window-based methods are applied in three different ways. The first is dependent on the subdivision of study area. The local density is estimated according to counts of points in cells. Many approaches, such as STING (Wang, Yang, & Muntz, 1997), CLIQUE (Agrawal, Gehrke, Gunopulos, & Raghavan, 1998), and MAFIA (Nagesh, Goil, & Choudhary, 1999), achieve this by connected dense cells. Nevertheless, the features identified by those methods may be significantly affected by issues related to cells, such as size and shape. The second way for estimating local density is to utilize kernel functions, usually a predefined spatial probability density function (PDF) (for example, the Gaussian function), to model the distribution of density of points (Dasgupta & Raftery, 1998; Rogerson, 2001; Fraley & Raftery, 2003). Though kernel methods are not restricted within predefined cells, they may fail to locate features with complex shapes due to the influence imposed by the shapes of kernel functions. The third is based on the spatial scan statistics. Openshaw, Charlton, Wymer, and Craft (1987), Openshaw, Charlton, Craft, and Birth (1988) proposed an automatic cluster detector, i.e. the geographical analysis machine (GAM), to identify clusters via excluding the "false clusters"

* Corresponding author. Fax: +86 1064889630.
E-mail addresses: peit@lreis.ac.cn (T. Pei), axing@lreis.ac.cn, azhu@wisc.edu (A-Xing Zhu), zhouch@lreis.ac.cn (C. Zhou), Libl@lreis.ac.cn (B. Li), qincz@lreis.ac.cn (C. Qin).

0198-9715/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compenvurbsys.2009.08.001

which occur by chance. Nevertheless, GAM lacks a clear statistical standard for evaluating the number of significant circles. In addition, many identified significant circles, which overlap, often contain the same cluster of cases. As a result, the GAM maps may give the appearance of excess clustering, with a high percentage of "false positive" circles. To reduce the false positives, more sophisticated statistic indices were proposed to identify locations where there are more events than expected. In the spatial scan method, the scan window, defined as a circular (with a space radius) or cylinder (with a circular geographic base and the height corresponding to time), moved in space (and time) to detect regions of significant clustering (Kulldorff & Nagarwalla, 1995; Kulldorff, 1997). Although the spatial scan statistics have been widely used in disease surveillance (Kulldorff, Heffernan, Hartman, Assuncao, & Mostashari, 2005; Yan & Clayton, 2006; Gaudart et al., 2008), the embedded defect in the method is that the detected features may be significantly influenced by the shape of window. Inappropriate choice of window shape may split one feature into many small ones or merge different features into one.

Differing from the window-based methods, the distance-based methods use the distance between a point and its neighbor as an alternative to estimate local densities. The idea of the Shared Nearest Neighbor (SNN) (Jarvis & Patrick, 1973), in which the link between point p and q is created if and only if p and q have each other in their closest k nearest neighbor lists, was employed to distinguish clusters from noise. Because the SNN method links in uniform regions and break the ones in the transition regions, it can deal with clusters of varying tightness, which is referred to as "density independent". However, the threshold for separating clusters should be tuned interactively, for two distinct sets of points may be merged into one cluster if the threshold is small; or a natural cluster may be split into many small clusters due to natural variations within the cluster if the threshold is too high (Ertoz, Steinbach, & Kumar, 2002). To make clustering process more robust, Ester, Kriegel, Sander, and Xu (1996) used the kth nearest distance (the distance between the point and its kth nearest neighbor) as a measure of local density and proposed the DBSCAN method. Although DBSCAN is easily implemented and capable of identifying clusters with arbitrary shapes (Ester et al., 1996), the key parameters, i.e. Eps (the distance for defining the neighborhood of a given point) and Minpts (the minimum number of points in the neighborhood), in DBSCAN and its variants can only be determined visually and inappropriate choice of the parameters may lead to wrong results (Ankerst, Breunig, Kriegel, & Sander, 1999; Roy & Bhattacharyya, 2005; Lin & Chang, 2005). In order to reduce the subjectivity, Byers and Raftery (1998) proposed a classification model based on the Nearest Neighbor (NN) method (for simplicity, we refer to their method by NN hereafter), in which feature and noise are viewed as two homogenous Poisson processes with different intensities (in the following text, we use intensity when discussing a point process and density for a cluster). The feature process, with the higher intensity, is viewed as a group of clustered points in a restricted area while the noise process, with the lower intensity, is randomly distributed over the entire region. Features and noise may be separated according to the difference in their kth nearest distance. Pei et al. (2006) extended the method to a clustering model to group points into different clusters. The approaches to the distanced-based method have reduced the parameters to the only one (i.e. k). Nevertheless, the classifications produced by these methods are still sensitive to k. A poor choice of the parameter may lead to a high error rate of classification. Although the classification entropy (CE) is employed to determine k (Byers & Raftery, 1998) and later used to identify spatial patterns in inhomogeneous spatial processes (Yang & Lee, 2007), it is a subjective process and the derived value of k may fail to produce correct results. Therefore, identifying an optimum value for k is still a difficult problem.

In this paper, we present a new method, the Collective Nearest Neighbor (CLNN) approach which separates features from noise and bypasses the problem of determining the optimum value of k. The CLNN method is divided into three steps. The first is to classify points using the NN method at various values of k. The second is to select acceptable classification layers, in which both feature and noise are homogenous Poisson processes, from the classification results. The third is to classify points by averaging the acceptable layers.

The rest of the paper is arranged as follows. In Section 2, we review the NN method which is the base of the CLNN method. The CLNN algorithm is described in detail in Section 3. In Section 4, we illustrate the algorithm through simulated data sets and discuss the important issues when applying the CLNN algorithm. In Section 5, we provide a case study of earthquakes in southwestern China to evaluate the CLNN method. Conclusions are given in Section 6.

2. Framework of nearest neighbor method

2.1. Probability density function of *k*th nearest distance

The number of points k in any planar region S with area $|S|$ might be assumed to be generated by a homogeneous Poisson process if it follows the distribution below:

$$f_{\lambda|S|}(k) = \frac{e^{-\lambda|S|}(\lambda|S|)^k}{k!} \tag{1}$$

where the expected constant intensity is λ and the mean is $\lambda|S|$ (Lucio & Brito, 2004). Thus, for a given point p in the Poisson process, the probability distribution of its kth nearest distance D_k (the distance between p and its kth nearest neighbor) can be derived by computing the probability of including 0, 1, 2, …, $k-1$ points within the circle of $A(p,x)$, in which p is the center and x is the radius.

$$P(D_k \geqslant x) = \sum_{m=0}^{k-1} \frac{e^{-\lambda\pi x^2}(\lambda\pi x^2)^m}{m!} = 1 - F_{D_k}(x) \tag{2}$$

where $F_{D_k}(x)$ is the cumulative distribution function of D_k, λ is the intensity of the Poisson process. If D_k is larger than x, there must be 0 or 1 or 2 … $k-1$ points within the circle $A(p,x)$. The pdf of D_k ($f_{D_k}(x;k,\lambda)$) has proved to be the derivative of $F_{D_k}(x)$:

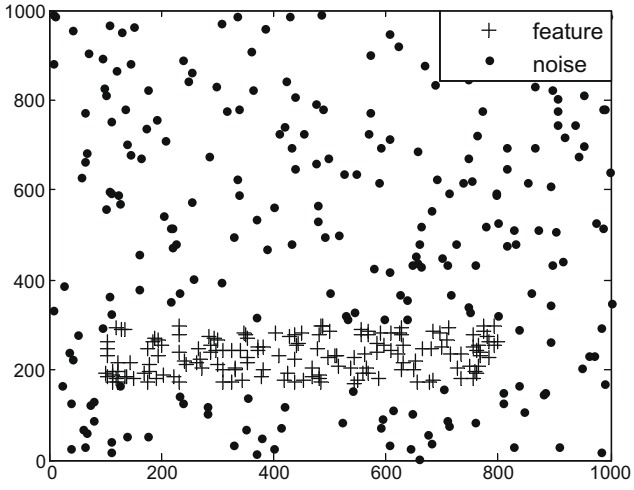

Fig. 1. The simulated data consisting of a rectangle feature and noise.

Please cite this article in press as: Pei, T., et al. Detecting feature from spatial point processes using Collective Nearest Neighbor. *Computers, Environment and Urban Systems* (2009), doi:10.1016/j.compenvurbsys.2009.08.001

$$f_{D_k}(x;k,\lambda)=\frac{dF_{D_k}(x)}{dx}=\frac{e^{-\lambda\pi x^2}2(\lambda\pi)^k x^{2k-1}}{(k-1)!} \quad (3)$$

where λ is the same as that in Eq. (2) (Byers & Raftery, 1998).

In this context, the noise and the feature can be thought of as two superimposed homogeneous Poisson processes with different intensities, say, λ_1 and λ_2. The bimodal pdf of D_k can be expressed as:

$$D_k \sim wf_{D_k}(x;k,\lambda_1)+(1-w)f_{D_k}(x;k,\lambda_2) \quad (4)$$

where $w \in [0\ 1]$ is the mixing coefficient, and λ_1 and λ_2 are the intensities for the feature process and the noise, respectively (Byers & Raftery, 1998).

Because points are in one-to-one correspondence with their kth nearest distances D_ks, the points can be classified as feature or noise according to the difference in their D_ks.

2.2. Estimation of λ_1, λ_2, and w

The Expectation–Maximization (EM) algorithm can be used to estimate the parameters λ_1, λ_2, and w, which characterize the mixture distribution of D_k. The EM algorithm is usually employed to solve the missing data problem (Celeux & Govaert, 1992; Moon, 1996). The missing data in this context are the classification into the two processes (the feature and noise), with the probability (membership) value $\delta_i \in [0,1]$ $(i=1,2,\ldots,n)$ for point q_i $(i=1,2,\ldots,n)$, where n is the number of data points. If $\delta_i \geqslant 0.5$, point q_i is classified as feature, otherwise, q_i is classified as noise. For more details about the estimation of λ_1, λ_2, and w, see Appendix A.

In fact, we need to know the optimum value of k before estimating these parameters (λ_1, λ_2, and w). As discussed in Introduction, k is very difficult to determine. To overcome this obstacle, we de-

Table 1
Classification results generated by NN.

k	Parameters			Number of false points		Indices for CSR	
	w	λ_1 (10^{-3})	λ_2 (10^{-4})	F_f	F_n	Feature	Noise
1	0.5131	1.3026	2.2486	75	7	0	0
2	0.6543	1.8623	2.4876	14	23	0	0
3	0.5903	1.4788	2.2825	22	9	0	0
4	0.5647	1.3159	2.2063	21	3	0	1
5	0.5794	1.3373	2.3034	13	3	0	1
6	0.5634	1.2655	2.2299	15	2	0	1
7	0.5659	1.2672	2.2345	15	2	0	1
8	0.5709	1.2649	2.2488	14	2	0	1
9	0.5670	1.2438	2.2583	14	2	1	1
10	0.5699	1.2338	2.2605	12	1	1	1
11	0.5680	1.2380	2.2511	13	0	1	1
12	0.5599	1.2002	2.2126	15	0	1	1
13	0.5522	1.1770	2.1714	18	0	1	1
14	0.5538	1.1695	2.1808	18	0	1	1
15	0.5487	1.1502	2.1711	20	0	1	1
16	0.5425	1.1222	2.1667	20	0	1	1
17	0.5378	1.1073	2.1621	25	0	1	1
18	0.5371	1.1052	2.1502	25	0	1	1
19	0.5324	1.0845	2.1550	26	0	1	1
20	0.5310	1.0766	2.1409	25	0	1	1
21	0.5269	1.0563	2.1279	27	0	1	1
22	0.5260	1.0528	2.1345	27	0	1	1
23	0.5239	1.0444	2.1408	29	0	1	1
24	0.5189	1.0293	2.1311	30	0	1	1
25	0.5185	1.0173	2.1146	31	0	1	1
26	0.514	1.0098	2.0946	33	0	1	1
27	0.5178	1.0207	2.1096	31	0	1	1
28	0.5231	1.0242	2.1290	29	0	1	1
29	0.5206	1.0137	2.1291	29	0	1	1
30	0.5198	1.0107	2.1172	31	0	1	1
31	0.5222	1.0101	2.1306	29	0	1	1
32	0.5209	1.0026	2.1256	30	0	1	1
33	0.5161	0.9858	2.1063	31	0	1	1
34	0.5123	0.9742	2.0947	34	0	1	1
35	0.5088	0.9586	2.0816	35	0	1	1
36	0.5066	0.9489	2.0731	36	0	1	1
37	0.5057	0.9384	2.0612	36	0	1	1
38	0.5066	0.9322	2.0611	36	0	1	1
39	0.5024	0.9201	2.0487	38	0	1	1
40	0.4997	0.9111	2.0413	38	0	1	1
41	0.498	0.9012	2.0258	39	0	1	1
42	0.4962	0.8914	2.0164	39	0	1	1
43	0.4966	0.8832	2.0166	41	0	1	1
44	0.4955	0.8779	2.0017	40	0	1	1
45	0.4931	0.8673	1.9763	40	0	1	1
46	0.4926	0.8638	1.9724	41	0	1	1
47	0.4860	0.8467	1.9581	44	0	0	1
48	0.4816	0.8338	1.9434	44	0	0	0
49	0.4754	0.8183	1.9304	47	0	1	0
50	0.4649	0.8006	1.9046	51	0	1	0

Note: λ_1 is the intensity of feature, λ_2 is the intensity of noise, F_f is the number of false feature points, F_n is the number of false noise points, "1" symbolizes homogeneity and "0" symbolizes inhomogeneity in the column of "indices for CSR".

Please cite this article in press as: Pei, T., et al. Detecting feature from spatial point processes using Collective Nearest Neighbor. *Computers, Environment and Urban Systems* (2009), doi:10.1016/j.compenvurbsys.2009.08.001

velop the CLNN method for classifying the data without determining the optimum value of k.

3. CLNN method

The idea of CLNN is to identify features by overlaying the selected classification layers, in which both feature and noise are tested to be homogenous. The CLNN method consists of three steps. In the first step, points are classified into clustered points and noise at all values of k by using the NN method. In the second step, CLNN selects classified layers, where subsets (i.e. feature and noise which are generated at different values of k) are tested to be homogenous Poisson processes (we call these layers acceptable layers), and save membership indicator $I_{i,k}$ for each point p_i (that is, for an acceptable layer, if point p_i belongs to a feature at k then $I_{i,k} = 1$; otherwise, $I_{i,k} = 0$). Note that $I_{i,k}$ is set to 0 for all points if the layer generated at k is not acceptable. In the third step, point p_i is finally classified as feature if $\sum_k I_{i,k} \geqslant T$, where T is the threshold for separating features from noise. Essentially, the CLNN method can be seen as an overlaying operator and the final result is acquired by overlaying the acceptable layers. For this reason, CLNN may reduce the number of false points. Here, false points (or misclassified points) include two types of points, i.e. feature points that are misclassified as noise and noise that is misclassified as feature. Below is a detailed description of the CLNN algorithm.

```
Main(Input: DataA, layer_threshold, K_max)
Begin
  nitialize Final_Membership;
  Let Accepted_LayerNumber = 0;
  For k = 1: K_max
      [Feature_Set, Noise_Set, Fuzzy_Membership] = Classify-
ByNN(DataA, k);
    Membership = Harden(fuzzy_Membership);
    If IsHomogenous(Noise_set) and IsHomogenous(Feature_
set)
      Final_Membership = Final_Membership + Membership;
      Accepted_LayerNumber = Accepted_LayerNumber + 1;
    Else
      Continue;
  End
End
    Feature_points = (Final_Membership>=layer_threshold);
    Noise_points = (Final_Membership<layer_threshold);
    Return Feature_points, Noise_points;
End
```

where *DataA* is the point set, $K_{\max}$ is the total number of layers, *Accepted_LayerNumber* is the number of accepted layers, *Final_Membership* is the matrix for saving the summation of the membership values of each point, *Feature_Set* and *Noise_Set* are the feature and noise generated by the NN at k, respectively, *Fuzzy_Membership* is a matrix for saving the fuzzy membership value of each point which is classified at k, *Membership* is a matrix for saving the membership of each point in *DataA*, *layer_threshold* is the threshold for separating features from noise, *Feature_Points* and *Noise_Points* are the features and noise, respectively, which are eventually separated by the CLNN algorithm.

Function ClassifiedByNN(*DataA*, k) is to classify *DataA* into feature and noise at k using the NN method, in which the parameters (λ_1, λ_2, w), used to construct the discrimination function (Eq. (4)), are estimated by the EM algorithm (see Appendix A for details). Function Harden is to harden the fuzzy membership values of points. If the fuzzy membership value of a point is less than 0.5, then the point belongs to noise; if that of a point is equal to or greater than 0.5, then the point belongs to a feature; feature points are indicated as 1 and noise as 0. IsHomogenous is a function used to determine if a process is homogeneous. If both *Noise_set* and *Feature_set* in a layer are deemed as homogenous Poisson processes, the layer is deemed as acceptable and *Membership* is added to *Final_Membership*; otherwise, the layer is deemed as unacceptable and excluded. A point is eventually classified as feature if the point is labeled as 1 at least *layer_threshold* times in the whole process; otherwise, it is classified as noise.

In the algorithm of the CLNN, the determination of Complete Spatial Randomness (CSR) is the key to the selection of acceptable layers. There are two types of indices, i.e. quadrat methods and distance methods, for the determination of CSR (Cressie, 1991). The quadrat methods need to partition the study area into subsets and collect counts of events in subsets whereas the distance methods only need to calculate the kth nearest distance of a point. As the distance methods make full use of the precise information on the locations of points and have the advantage of not depending on arbitrary choices of quadrat size and shape, we use a distance-based index ($2\pi\lambda\sum_i^n W_i^2$) for the determination of CSR (Skellam, 1952), where W_i is the nearest distance of a point, n is the number of points, λ is the intensity of the process. If the index follows the distribution χ^2_{2n}, the CSR is accepted; otherwise, the CSR is rejected. In this paper, CSR is synonymous with a homogenous Poisson process.

The CLNN algorithm has two parameters, i.e. $K_{\max}$ and *layer_threshold*. $K_{\max}$ should be set to be large enough to ensure that all potential acceptable layers are selected and testified for CSR. As a result, we let $K_{\max} = num/8$ to $num/3$ when running the CLNN, where *num* is the number of points in the data. Nevertheless, one can increase $K_{\max}$ if the layer generated at $K_{\max}$ is acceptable. *layer_threshold* is used to determine the final membership of a point. In detail, if *Final_Membership* of a point is not less than *layer_threshold*, the point belongs to a feature; otherwise, it belongs to noise. Pei, Zhu, Zhou, Li, and Qin (2007) also found that a feature process, which is generated by the NN method, will be overestimated as k increases. Interested reader may refer to Pei et al. (2007) for the explanation. As a result, *layer_threshold* should be larger than *Accepted_LayerNumber*/*2* in order to produce fewer false points. In this paper, we let *layer_threshold* = *Accepted_LayerNumber*, which means that only those classified as feature in all acceptable layers are eventually deemed as feature points.

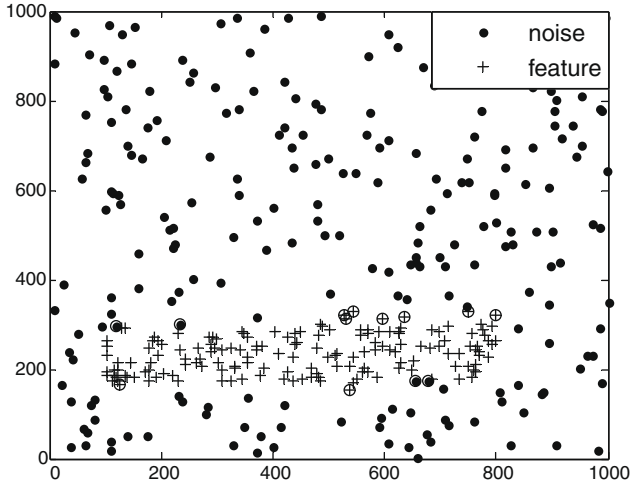

Fig. 2. Classification generated by CLNN: the number of false points is 13 ($F_f = 9$ and $F_n = 4$) (Symbols enclosed with a circle represent the false points).

Please cite this article in press as: Pei, T., et al. Detecting feature from spatial point processes using Collective Nearest Neighbor. *Computers, Environment and Urban Systems* (2009), doi:10.1016/j.compenvurbsys.2009.08.001

4. Identification of clustered point patterns

4.1. Classification result of simulated data

In order to evaluate the CLNN method, we used a simulated data. The simulated data, shown in Fig. 1, contain noise and a feature which is constrained in a rectangle. The x coordinates of feature were generated by simulating real numbers which follow the uniform distribution in [100 800]. The y coordinates of feature were uniformly distributed in [150 300]. The feature points were generated in a similar way, that is, x coordinates of feature points were generated form a uniform distribution in [0 1000] and y coordinates were generated in the same way. Note that noise points that fell into the area of feature were excluded. The data then were classified by the NN method as k increased from 1 to $K_{\max} = 50$. The results are listed in Table 1. The ratios (w) of the number of feature points to that of noise points are listed in the second column. The intensities (λ_1) of the feature are listed in the third column. The intensities (λ_2) of the noise are listed in the fourth column. The number of false feature points (F_f) listed in the fifth column indicate the numbers of noise points, which are classified as feature. The numbers of false noise points (F_n) listed in the sixth column refer to the numbers of feature points, which are classified as noise. The indices for CSR, listed in the seventh column and eighth column, indicate homogeneous Poisson processes (symbolized by "1") or inhomogeneous Poisson processes (symbolized by "0").

The actual values of w, λ_1, and λ_2 of the data in Fig. 1 are 0.5674, 0.001193, and 0.000219, respectively. Among those parameters (w, λ_1, λ_2) estimated in Table 1, the values which best approximate the actual values are acquired when $k = 9$, $k = 12$, and $k = 14$. According to the number of false points at various values of k, the minimum number of false points are acquired at $k = 10$ ($F_f = 12$,

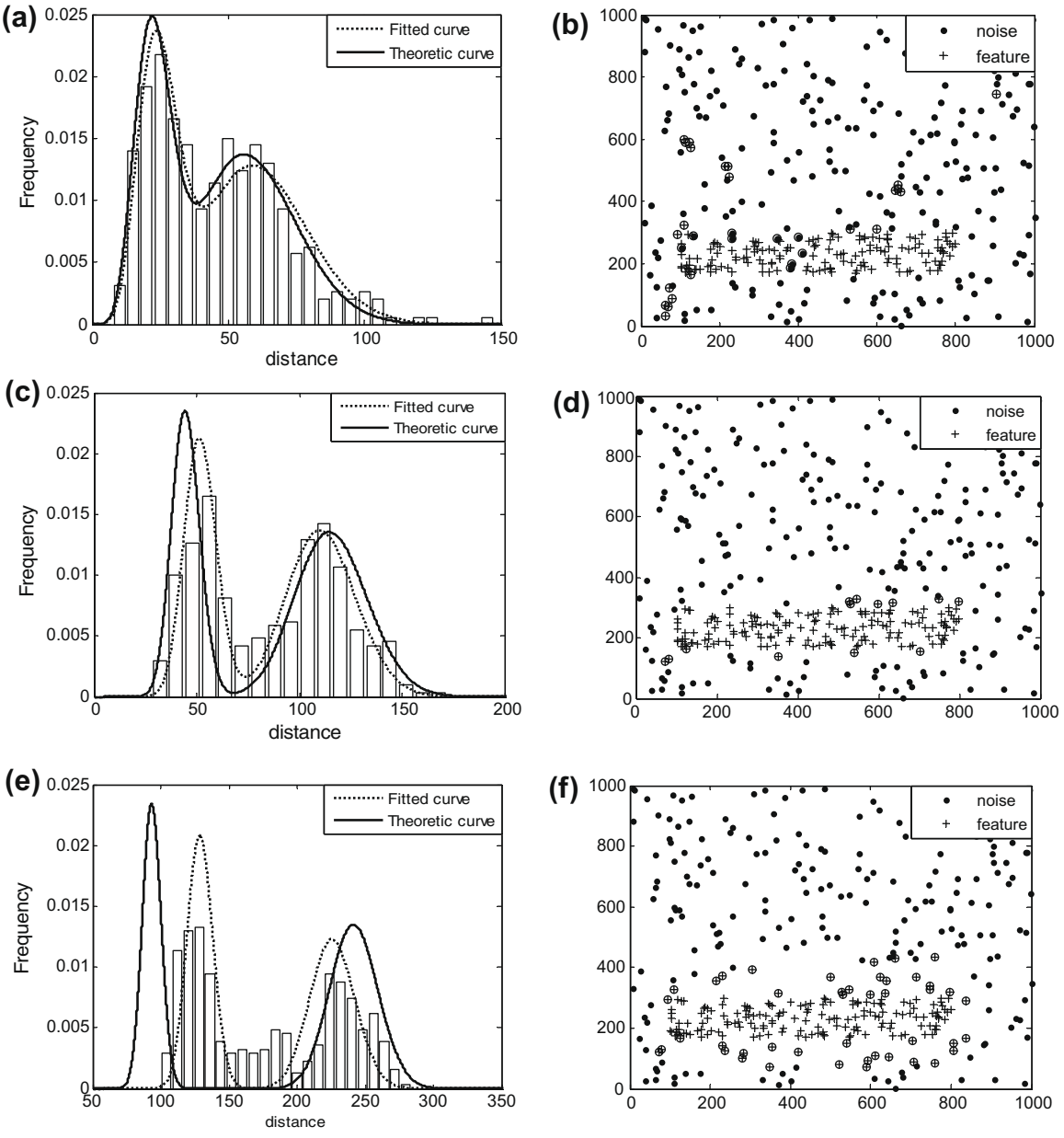

Fig. 3. The histograms of the kth nearest distances of points in Fig. 1 and the results of classification generated by NN at $k = 3$, 11, and 47 (symbols enclosed with a circle represent false points). (a) histogram at $k = 3$, (b) classification at $k = 3$ (31 false points), (c) histogram at $k = 11$, (d) classification at $k = 11$ (13 false points), (e) histogram at $k = 47$, and (f) classification at $k = 47$ (44 false points).

Please cite this article in press as: Pei, T., et al. Detecting feature from spatial point processes using Collective Nearest Neighbor. *Computers, Environment and Urban Systems* (2009), doi:10.1016/j.compenvurbsys.2009.08.001

F_n = 1), and 11 (F_f = 13, F_n = 0). According to the indices for CSR, 38 layers are identified as acceptable layers. We also note that the feature process shifts from inhomogeneity to homogeneity as k increases from 8 to 9, and shifts back to inhomogeneity when k increases to 47.

We then identify features by averaging the acceptable layers, in which both feature process and noise are labeled as "1". The classification is shown in Fig. 2. The number of false points is 13 (F_f = 13, F_n = 0), which is the same as the optimum result among those generated by the NN method as k increases from 1 to 50.

4.2. Shift between homogeneity and inhomogeneity in classified layers

In Table 1, acceptable layers can only be seen when k is between 9 and 46. To explain this phenomenon, we draw histograms of the simulated data along with the fitted curves and the theoretical curves generated when k = 3, 11, and 47, respectively (Fig. 3a–e).

In Fig. 3a, the fitted curve is similar to the theoretical one. Nevertheless, we find that the feature process and the noise process, generated when k = 3, show inhomogeneous (see Table 1). The histogram is not clearly bimodal and there is no strong distinction between the feature process and the noise process (Fig. 3a). Obviously, it is the large number of fuzzy points between these two processes (i.e. two peaks in the histogram) that cause many false points ($F_f = 22$, $F_n = 9$) (Fig. 3b). Due to the presence of the false points, both processes show inhomogeneous (see Table 1).

Compared with the histogram drawn at k = 3, the histogram at k = 11 is clearly bimodal, and the fitted curve derived from the histogram do not significantly deviate from the theoretic curve (Fig. 3c). Therefore, a better classification is acquired at k = 11 ($F_f = 13$ and $F_n = 0$) (Fig. 3d). As fewer false points are generated, both feature and noise are deemed as homogeneous Poisson processes (see Table 1).

In Fig. 3e, the histogram of feature (left component) shows significant right-bias compared with the theoretic curve, whereas the histogram of noise (right component) shows significant left-bias. As a result, the fitted curve derived from the histogram significantly deviates from the theoretic model when k = 47. We call this phenomenon the inner edge effect. That is, as k increases, the kth nearest distances of feature points near the border between feature and noise become longer on average compared with those in the center of feature process. This is because more noise points become the kth nearest neighbors of feature. On the contrary, the kth nearest distances of noise points near the border become shorter on average compared with those far away from the border. Due to the inner edge effect, the experimental mixture histograms may increasingly deviate from the theoretical curve as k increases. As a result, more false points were added to the processes which were

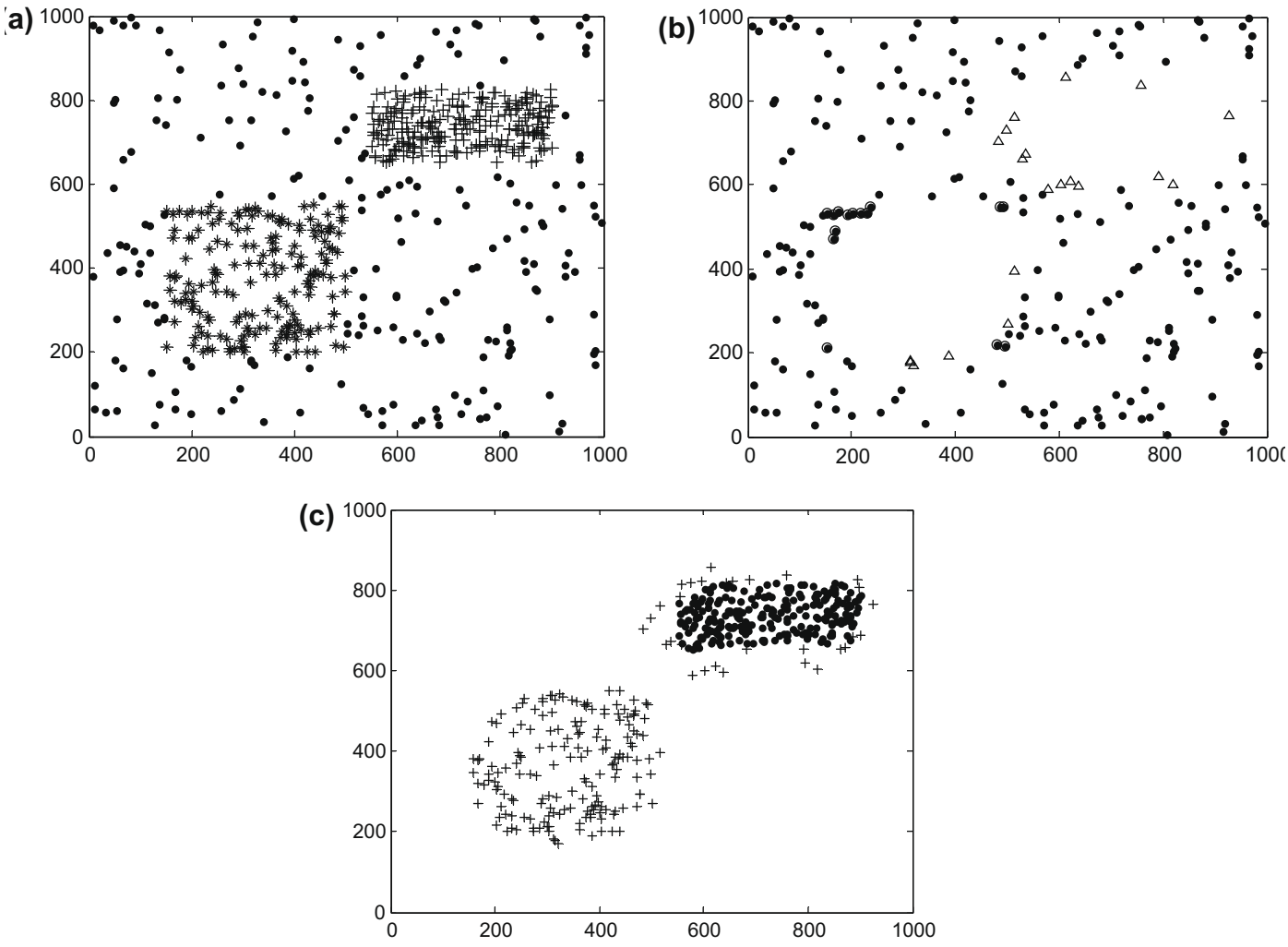

Fig. 4. Identification of features of different densities using CLNN. (a) Simulated data; (b) the first stage: separating features from noise (triangles indicate noise points which had been classified as feature and taken out; circles indicate feature point which had been classified as noise); and (c) the second stage: separating feature with high density from the one with low density (dots indicate the identified high-density-feature and crosses indicate the identified low-density-feature).

Please cite this article in press as: Pei, T., et al. Detecting feature from spatial point processes using Collective Nearest Neighbor. *Computers, Environment and Urban Systems* (2009), doi:10.1016/j.compenvurbsys.2009.08.001

classified (Fig. 3f). This will make the feature process or the noise process shift from homogeneity to inhomogeneity, that is, the feature process deviates from the CSR as k increases from 46 to 47 (Table 1).

4.3. Discussion on assumption

As discussed, CLNN is based on the assumption that the data under consideration are composed of two different homogenous point processes. However, in practice, real data may contain more than two point processes (i.e. multi-modal) or non-Poisson processes (for example, the Gaussian process). As a result, subgroups generated at any value of k may be testified to be inhomogeneous. This could result in *Accepted_layerNumber* = 0.

To solve the multi-modal problem, we can treat the inhomogeneous subgroup as a new data set and run the CLNN algorithm on the subgroup again until the result cannot be further divided. Here we use a simulated data set to illustrate the idea. Fig. 4a displays a data set which contains a rectangle feature with high density, a square feature with medium density and noise with low density. The generation of the data is same as that of the data in Fig. 1. We first applied NN to the data by setting k from 1 to 50. The result shows that the data were divided into two subgroups. The one with low density was testified to be homogeneous as $k = 5$ to 42

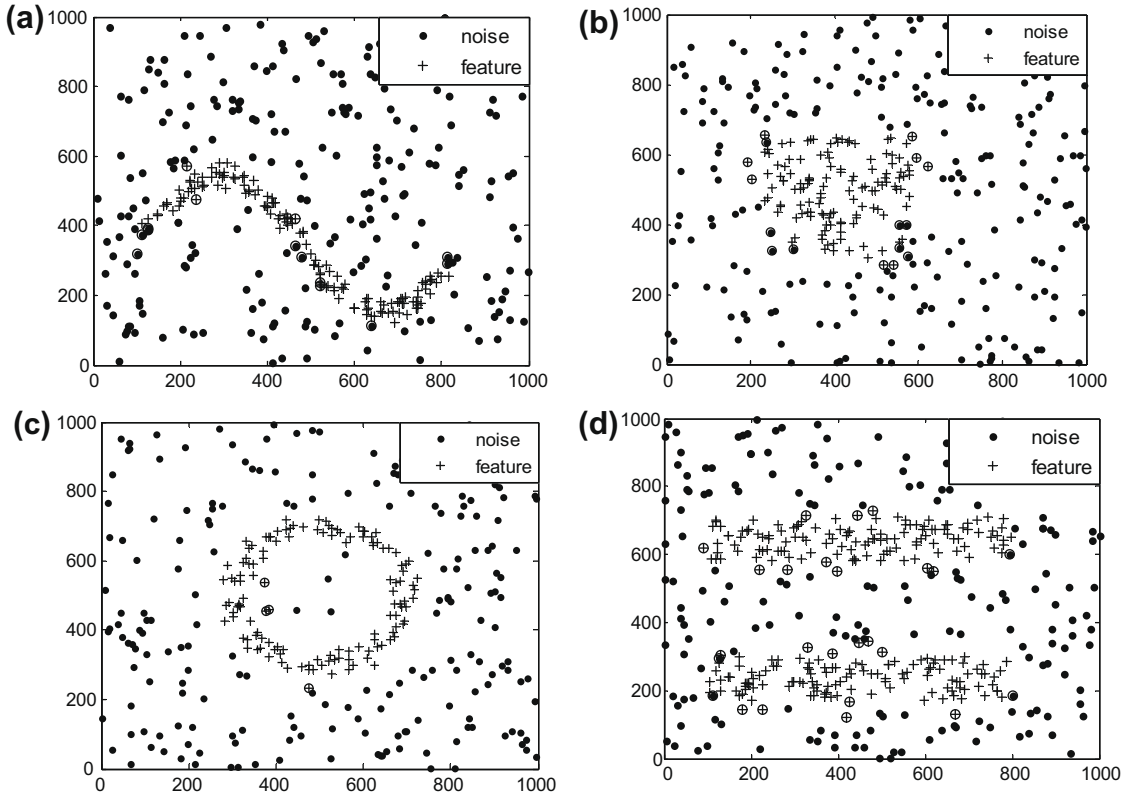

Fig. 5. Detection of features with different shapes using CLNN (symbols enclosed with a circle represent false points). (a) Sinusoid feature, (b) square feature, (c) ring feature, and (d) two-rectangle feature.

Table 2
Comparison between CLNN, ICLNN, CE, SaTScan and the optimum result among those generated by NN at various values of k.

		Sinusoid	Square	Ring	Two-rectangle
CLNN	Number of false points	13 ($F_f = 8$, $F_n = 5$)	16 ($F_f = 8$, $F_n = 8$)	4 ($F_f = 4$, $F_n = 0$)	25 ($F_f = 21$, $F_n = 4$)
	Clustering or random	Random	Random	Random	Random
CE	Number of false points	21 ($F_f = 17$, $F_n = 4$, $k = 5$)	21 ($F_f = 13$, $F_n = 8$, $k = 10$),	17 ($F_f = 13$, $F_n = 4$, $k = 5$)	29 ($F_f = 20$, $F_n = 9$, $k = 6$)
	Clustering or random	Clustering	Clustering	Random	Random
SaTScan (p value = 0.001)	Number of false points (circular)	38	24	49	70
	Clustering or random	Clustering	Random	Random	Clustering
	Number of false points (Elliptic)	57	33	42	42
	Clustering or random	Clustering	Clustering	Random	Random
Number of false feature points by the ICLNN ($\Delta k = 3$)		15 ($F_f = 9$, $F_n = 6$)	17 ($F_f = 10$, $F_n = 7$)	5 ($F_f = 5$, $F_n = 0$)	25 ($F_f = 17$, $F_n = 8$)
Minimum number of false points among those at various values of k		12 ($F_f = 8$, $F_n = 4$, $k = 7$)	17 ($F_f = 13$, $F_n = 4$, $k = 18$)	9 ($F_f = 8$, $F_n = 1$ ($k = 7$)	24 ($F_f = 15$, $F_n = 9$, $k = 9$)

Note: F_f and F_n are the same as those in Table 1. The "circular" option and the "elliptic" option (for "spatial window shape") were used when detecting features in all data sets with SaTScan.

Please cite this article in press as: Pei, T., et al. Detecting feature from spatial point processes using Collective Nearest Neighbor. *Computers, Environment and Urban Systems* (2009), doi:10.1016/j.compenvurbsys.2009.08.001

whereas the one with high density was testified to be inhomogeneous as $k = 1$ to 50. In this context, we selected the subgroups of low density, which were generated as $k = 5$ to 42 and homogeneous, and averaged the membership values of points in these subgroups. The noise was detected according to the overlaid result. We then took out the detected noise from the original data and kept the subgroup of high density for the next-round of classification. Result of the first stage is shown in Fig. 4b. In the second stage, we treated the subgroup of high density as a new data set and ran CLNN again. The separated features are displayed in Fig. 4c. If a data set contains more processes, we can separate features from noise by running CLNN repeatedly via the process above.

If the points are distributed as Gaussian processes, which cannot be seen as homogenous, the Gaussian-like cluster can be treated as a mixture of homogeneous point processes and further divided into finer homogeneous clusters by using the method above. The finer homogeneous clusters are then merged into natural clusters through post-processing. The reason that a Gaussian cluster can be seen as a mixture of homogeneous point processes is described as follows. In most practical applications, the intensity function of a given point process $\lambda(x)$ is a continuous function. Because of its continuity, the function can be divided into definite (or indefinite) distinctive intervals in each of which the intensity can be viewed as a constant (Illian, Penttinen, Stoyan, & Stoyan, 2008). According to the definition of homogeneous point process, portions in each of interval can be considered as a homogeneous point process. Another way of dealing with Gaussian-like cluster is to apply the model-based cluster method (for example, using Gaussian function as a kernel) (Fraley & Raftery, 1998; Fraley & Raftery, 2003).

4.4. Evaluation on power for identifying arbitrary-shape features

To see if the CLNN method could handle data of any arbitrary shape, we apply it to four other simulated data sets containing different shapes of features, i.e. sinusoid, square, ring, and two-rectangles (Fig. 5). The features of sinusoid and ring were generated as follow. First, the regions of sinusoid and ring were drawn. Second, the points were generated by simulating points which are uniformly distributed within the whole study area and only those that fell into the specific regions were selected. The classification results show that CLNN clearly reveals the features in different point sets. The number of false points generated by CLNN are 13, 16, 4, and 25 while the minimum numbers of false points indicated by the NN method as k increases from 1 to 50 are 12, 17, 9, and 24, respectively (see Table 2). Interestingly, the number of false points for the square feature and that for the ring, generated by CLNN, are even smaller than the corresponding minimum numbers of false points. The classifications on the four data sets display that the CLNN method may have the ability of identifying features with arbitrary shapes.

4.5. Comparison between CLNN, SNN, Spatial Scan method, and CE

In order to evaluate the efficiency of CLNN, we make a comparison between the CLNN, the classification entropy (CE) method, the SNN method, and the spatial scan method using the data in Fig. 1. CE is an index that can measure the improvement of the classification as k increases. The classification entropies up to $K_{max} = 50$, which are calculated from the data in Fig. 1, are sequentially plotted in Fig. 6. The change-point of the curve in Fig. 6 is estimated to be 5 with the CE method, which is very different from the optimum value ($k = 10, 11$). We find that when $k = 5$ the number of false feature points and that of noise are 13 and 3, respectively (Table 1 and Fig. 7a). Of the SNN, k is the only one parameter, we tested k with different values and found SNN produced the minimum number of false points when $k = 3$. The classification is displayed in Fig. 7b. To evaluate the spatial scan method, we used SaTScan (Version 7.0.3) which was downloaded from www.satscan.org. Because the simulated data have no temporal attribute, we chose the "Bernoulli model" with "pure spatial scan" (for details, see the help in the software). Three parameters need to be set before running the program, namely, the "gridding scheme", "percentage of the population at risk" and "spatial window shape". Here, we scan the data by fixing percentage of the population at risk (10%) and varying the gridding scheme ($25 * 25$, $50 * 50$, and $100 * 100$) and setting "elliptic" option. Results generated by the spatial scan method are shown in Fig. 7c–e.

To judge which result is the best, we consider two aspects. One is the number of false points and the other is the spatial distribution of the false points. If the distribution of false points demonstrates clustering patterns, the method may leave some clusters or a portion of a cluster out or produce false clusters. To assess this, we treat the false points as a new data set and use the statistics $2\pi\lambda\sum_i^n W_i^2$ (Skellam, 1952) to determine if the data set is CSR. The comparison showed that the CLNN produces the minimum number of false points while other methods produced more false points. Among the competitors, the SNN produce not only the largest number of false points but also more false features (Fig. 7b). In addition, we have proved that false points generated by the SNN and SaTScan ($25 * 25$, $50 * 50$, and $100 * 100$) demonstrate clustering.

Because SNN does not adapt to the data in which the density transit from feature to noise smoothly (Ertoz et al., 2002), we did not use the method for further comparison. As to the spatial scan method, since the middle size of grid (i.e. the size when the area is divided into $50 * 50$ cells) may produce better result, we use the $50 * 50$ cells for the following computation. We then apply the spatial scan method, the CE method to the point sets in Fig. 5. Note that both "elliptic" and "circular" options were tried in SaTScan when detecting features in all data sets. Results can be found in Table 2. The comparison shows that: (1) CLNN generate the minimum number of false points and (2) only do false points generated by CLNN show random. Both show that the results generated by CLNN are the best.

According to the comparison above, we find that the CLNN method produced more precise results and needs little prior knowledge in the estimation of parameters.

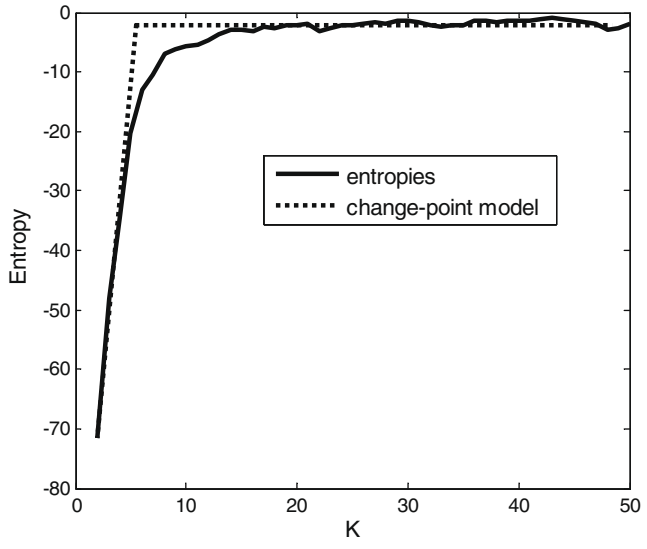

Fig. 6. Plot of classification entropies of the simulated data in Fig. 1 over k ($K_{max} = 50$).

Please cite this article in press as: Pei, T., et al. Detecting feature from spatial point processes using Collective Nearest Neighbor. *Computers, Environment and Urban Systems* (2009), doi:10.1016/j.compenvurbsys.2009.08.001

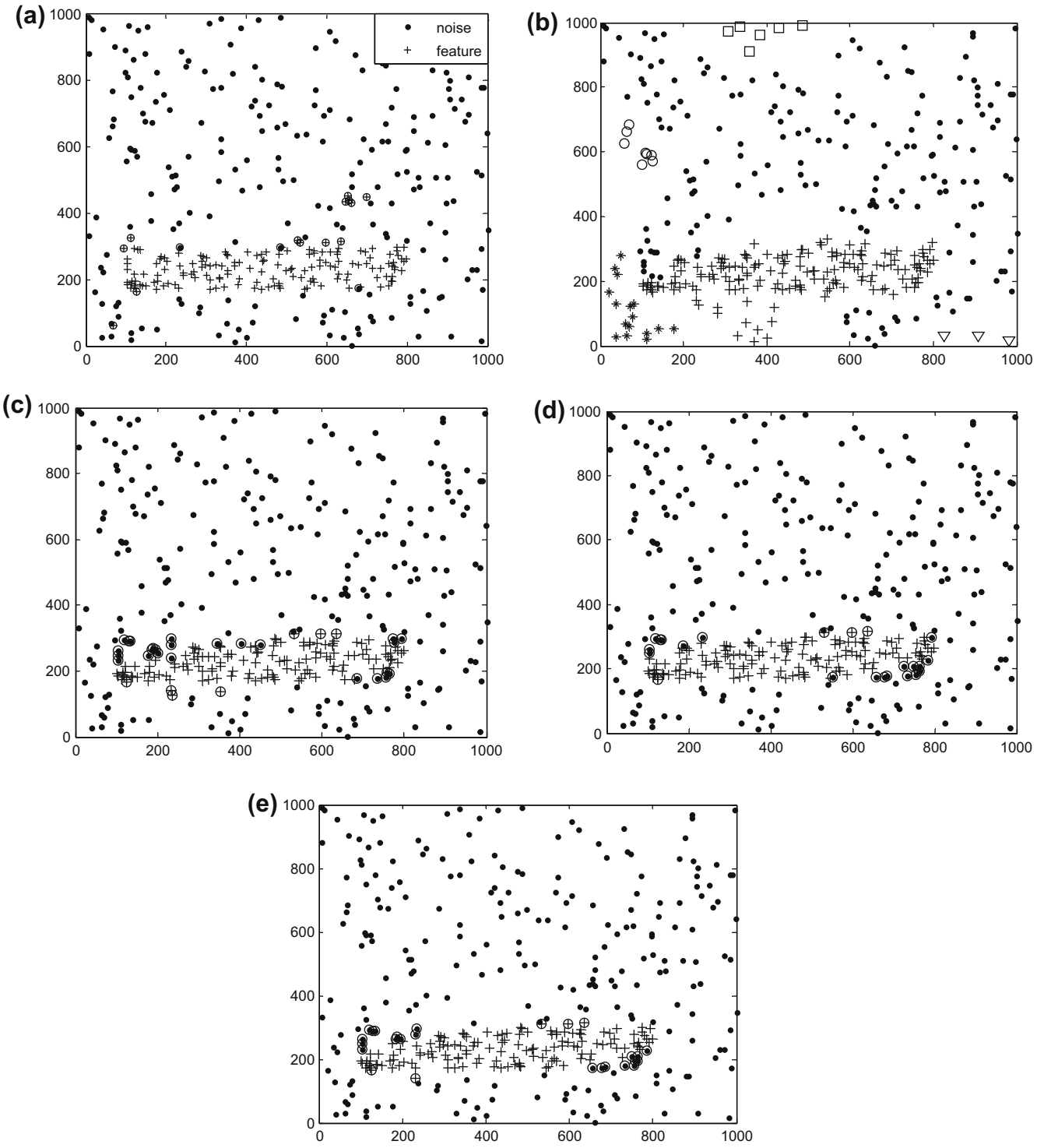

Fig. 7. Results generated by CE, SaTScan (Bernoulli model with pure spatial scan), and SNN. (a) CE ($k = 5$); (b) SNN ($k = 3$, 56 false positives); (c) SaTScan ($25 * 25$ cells, percentage of the population at risk = 10%, one most-likely cluster (p value = 0.001) and 32 false positives); (d) SaTScan ($50 * 50$ cells, percentage of the population at risk = 10%, one most-likely cluster (p value = 0.001) and 24 false positives); and (e) SaTScan ($100 * 100$ cells, percentage of the population at risk = 10%, one most-likely cluster (p value = 0.001) and 26 false positives).

4.6. Complexity of CLNN and modified version of CLNN

The complexity of CLNN is determined by three factors: the computation of kth nearest distance, the sorting of k nearest distances and the total number of layers (K_{max}). The complexity of the computation of kth nearest distance is $O(n^2)$, where n is the number of points. The complexity of sorting k nearest distances is also $O(n^2)$. As a result, the total complexity of CLNN is $O(n^2 * K_{max})$, which is same as that of CE. Both CLNN and CE are time-consuming processes since they need to classify points at various values of k. In order to reduce the complexity of CLNN, we propose a simplified algorithm, named the Interval Averaging Nearest Neighbor (ICLNN) method. The idea of ICLNN is to classify a data set using NN at an interval of Δk, and then to follow the same steps

Please cite this article in press as: Pei, T., et al. Detecting feature from spatial point processes using Collective Nearest Neighbor. *Computers, Environment and Urban Systems* (2009), doi:10.1016/j.compenvurbsys.2009.08.001

as those of the CLNN algorithm. Thus, the total complexity of the ICLNN method reduces to $1/\Delta k$ of that of CLNN. Setting $\Delta k = 3$, we applied the ICLNN method to the data set of Fig. 1. The number of false points is 15 (15 false feature points and 0 false noise points). We then run ICLNN on the data sets in Fig. 5. The numbers of false points of sinusoid feature, square feature, ring feature and two-rectangle feature are 15 (9 false feature points and 6 false noise points), 17 (10 false feature points and 7 false noise points), 5 (5 false feature points and 0 false noise points), and 25 (8 false feature points and 17 false noise points), respectively (see Table 2), which are close to those produced by CLNN but are better than those produced by CE and the spatial scan method. The larger Δk is, the less complexity the ICLNN algorithm has. Nevertheless, it will trade off accuracy of the classification for simplicity.

5. Case study

5.1. Clustered earthquakes and background earthquakes

Clustered earthquakes are usually considered as foreshocks or aftershocks of a strong earthquake (Wu, Jiao, Lu, & Wang, 1990; Wyss & Toya, 2000). They might be perceived to be foreshocks if a strong earthquake occurs after them or to be aftershocks if a strong earthquake occurs before them (Umino, Okada, & Hasegawa, 2002). Thus, clustered earthquakes could serve as a primary clue to predict a strong earthquake if the possibility of them being aftershocks can be excluded (Wu et al., 1990 (chapter 1); Chen, Liu, & Ge, 1999; Reasenberg, 1999; Ripepe, Piccinini, & Chiaraluce, 2000).

Background earthquakes usually appear with a low intensity while clustered earthquakes usually occur with a higher intensity (Wyss & Toya, 2000; Pei et al., 2003; Matsu'ura and Karakama, 2005). In this regard, background earthquakes and clustered earthquakes can be treated as two superimposed homogenous spatial Poisson processes with different support domains and different intensities (Zhuang, Chang, Ogata, & Chen, 2005). Clustered earthquakes are difficult to identify due to the interference from background earthquakes. Therefore, the classification of these two types of earthquakes can be used as a case for evaluating the CLNN method on identifying features in a spatial point set.

5.2. Study area and seismic data

The study area is located from 100° to 107°E and from 28° to 34°N. It contains the eastern part of Tibet, the southern part of Sichuan and Chongqing, the northern part of Yunnan and the western part of Guizhou. It is an area in China with very intensive seismicity. There have been 27 strong earthquakes ($M \geqslant 6.0$) in this area between January 1, 1970 and July 31, 2008 (Feng & Huang, 1980; Feng & Huang, 1989; China Seismograph Network Data Management Center, 2009). The Wenchuan Earthquake, the most devastating earthquake in 2008, occurred on May 12, 2008 with

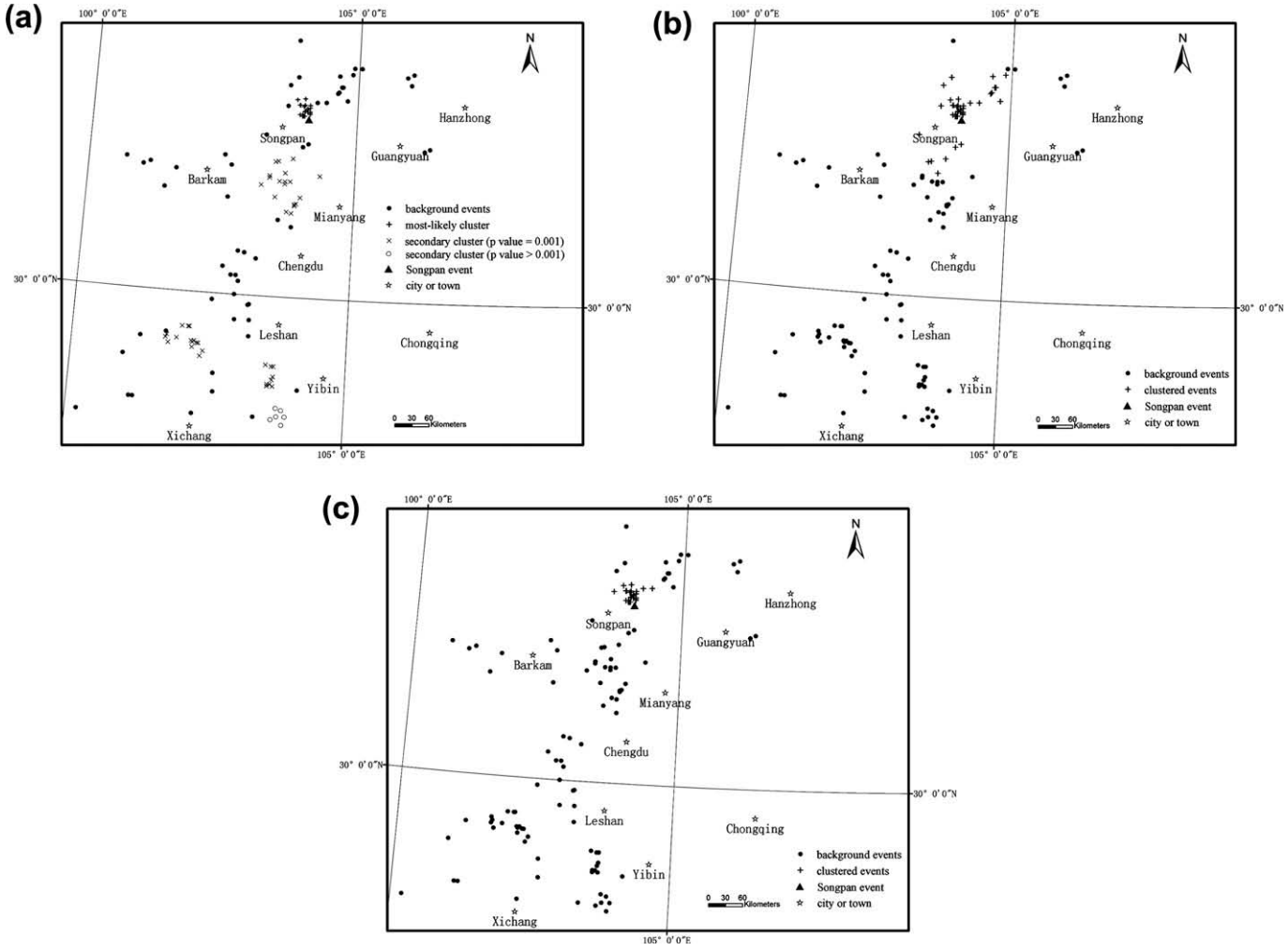

Fig. 8. The identification of clusters in earthquakes which occurred two months before the main earthquake. (a) SaTScan, (b) CE ($k = 20$), and (c) CLNN.

Please cite this article in press as: Pei, T., et al. Detecting feature from spatial point processes using Collective Nearest Neighbor. *Computers, Environment and Urban Systems* (2009), doi:10.1016/j.compenvurbsys.2009.08.001

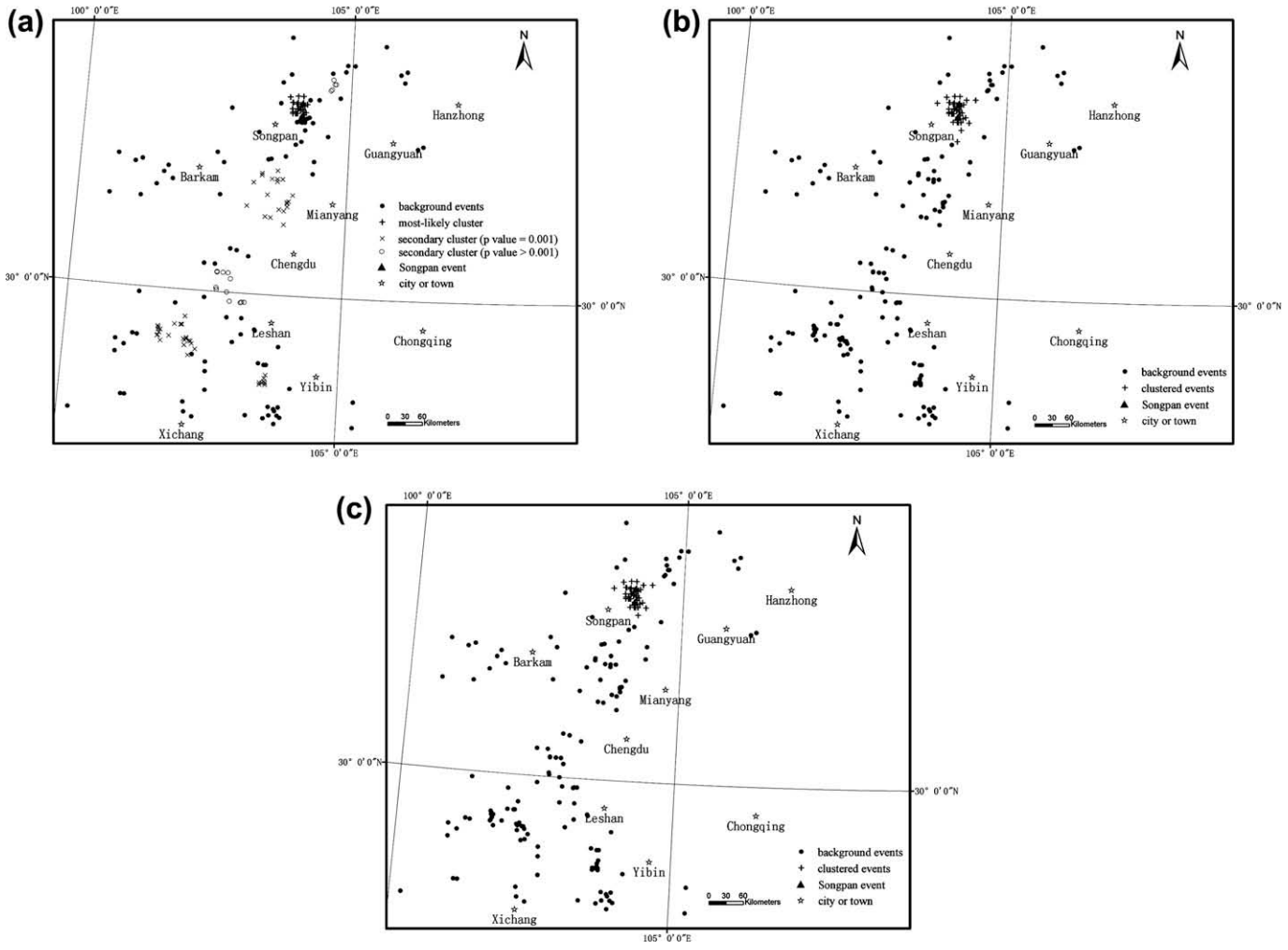

Fig. 9. The identification of clusters in earthquakes which occurred three months before the main earthquake. (a) SaTScan, (b) CE ($k = 18$), and (c) CLNN.

$M = 8.0$ and was also located in this area (China Seismograph Network Data Management Center, 2009).

The seismic data in this case study were selected from the Seismic Catalog of West China (1970–1975, $M \geqslant 1$) (Feng & Huang, 1980) and Seismic Catalog of West China (1976–1979, $M \geqslant 1$) (Feng & Huang, 1989). In order to identify clustered earthquakes, which may be helpful to indicate locations of strong earthquakes, we need to analyze earthquakes that occurred over different temporal intervals. Here, we used two different intervals (i.e. 2-month and 3-month). One is between June 15th, 1976 and August 15th, 1976 (altogether 128 epicenters were collected), and the other is between May 15th, 1976 and August 15th, 1976 (altogether 193 epicenters were collected). All earthquakes are larger than 2 (M). After the occurrence of the Xingtai quake ($M = 7.2$) in 1966, a nationwide seismographic network was set up and the ability to monitor seismicity has been greatly improved (Jiao, Wu, & Yang, 1990). According to Jiao et al. (1990), any earthquake at the level of 2 (M) and above were monitored and measured with good quality.

5.3. Results of case study

To evaluate the efficiencies of different methods, we used them to identify clustered earthquakes in the selected seismic data sets and analyzed the distribution of the identified clustered earthquakes as well as the relationship between these earthquakes and the epicenter of Songpan earthquake ($M = 7.2$), which occurred at (32°42′N, 104°06′E) on August 16, 1976 and caused the deaths of 38 people (Zhang, 1990 (chapter 2)). Fig. 8 shows the classification results of 2-month seismic records which were produced by SaTScan (with elliptic scanning window), CE ($k = 20$) and CLNN ($K_{max} = 40$). Fig. 8a demonstrates that five clusters have been identified by SaTScan, which consist of one most-likely cluster and four secondary clusters. The most-likely cluster is near the epicenter of Songpan earthquake and other clusters are located in the central, the south and the southwest of the research area, respectively. Differed from those identified by SatScan, CE identified only one cluster which is located in the north of the research area and significantly larger than the most-likely cluster identified by SatScan (Fig. 8b). Compared with those generated by these two methods, clustered earthquakes identified by CLNN cover the smallest area and are more concentrated around the epicenter of the Songpan earthquake (Fig. 8c).

Fig. 9 shows the classification results of 3-month seismic records which were generated by SatScan (with elliptic scanning window), CE ($k = 18$) and CLNN ($K_{max} = 40$). In Fig. 9a, six clusters were identified (two more clusters newly appeared, one is located to the northwest of Leshan and the other is located to the northeast of the most-likely cluster), including one most-likely cluster and five secondary clusters. In Fig. 9b, we find that the cluster identified by CE is more concentrated around the epicenter of Songpan earthquake compared with that shown in Fig. 8b.

Please cite this article in press as: Pei, T., et al. Detecting feature from spatial point processes using Collective Nearest Neighbor. *Computers, Environment and Urban Systems* (2009), doi:10.1016/j.compenvurbsys.2009.08.001

Fig. 9c shows that only one dense cluster was identified by CLNN, which are still focused around the epicenter of Songpan earthquake. Fig. 9b and c show that results generated by CE and CLNN are almost the same.

The analysis of Figs. 8 and 9 shows the difference between these three methods. SaTScan identified more clustered earthquakes than the other two methods. Although the most-likely clusters in both data sets are found to be near the epicenter of Songpan earthquake, the earthquakes which cover the epicenter of the Songpan earthquake were not identified as a cluster in the 3-month data (see Fig. 9a). This was probably caused by the limitation of the shape of scanning window. The situation of secondary clusters is more complicated. Some of the clustered earthquakes might be "false alarms" because they appear in the 2-month result and disappear in the 3-month result. And the remaining secondary clusters are not seen as the aftershocks or foreshocks of strong earthquakes because no strong earthquakes (larger than 5(M)) occurred between 1975 and 1977 (Feng & Huang, 1980; Feng & Huang, 1989). Although the remaining secondary clusters do not directly indicate strong earthquakes, they might indicate some geological events. The reason for the emergence of these secondary clusters needs further research. CE could provide the clustered earthquakes which are distributed around the main earthquake. However, the identified clustered earthquakes in the 2-month data set and the 3-month one show significantly difference in terms of shape and area. The clustered earthquakes concentrated around the Songpan earthquake were both identified in Figs. 8c and 9c, which demonstrates that the clustered earthquakes generated by CLNN are consistent between the two data sets. The clustered earthquakes, located in the immediate vicinity of the strong quake, could provide valuable information for indicating the epicenter of the Songpan earthquake.

To sum up, SaTScan can provide more clusters with different p values, among which the most-likely cluster should be paid much attention to and the secondary clusters may indicate some other potential anomalies or be "false alarms". In addition, the most-likely cluster might be underestimated or overestimated by SaTScan due to the limitation of the shape of scanning window. The clustered earthquakes generated by CE are concentrated around the epicenter, but the unstable performance of CE could lead to the error in the prediction of the epicenter of strong earthquakes. Differed with SaTScan and CE, the performance of CLNN is most stable in indicating the clustered earthquakes.

6. Conclusions

Identifying clustered point patterns is one of the major challenges in spatial data mining. Most methods are unable to achieve this since they are sensitive to input parameters or need prior knowledge on the data set under consideration. In this paper, we present the CLNN method to accomplish this task. The CLNN method assumes that a given spatial point set is composed of homogenous point processes which are distributed in different intensities. Features and noise can be differentiated by their difference on the kth nearest distance in CLNN. Consequently, the method does not rely on the subdivision of the study area and can identify clusters with arbitrary shapes. As the CLNN method classifies points by averaging the layers which are generated at various values of k and tested to be homogeneous, almost no parameters need to be adjusted interactively. This makes that CLNN is more objective and need less prior knowledge about the data set compared with most previous methods. The results of the simulated data and the case study show that the CLNN method, compared with SNN, SaTScan and CE method, could provide more stable foreshocks which may be used to indicate the epicenter of Songpan earthquake. Moreover, it was found that results generated by the CLNN method sometimes are even superior to the best results among those generated by NN at all values of k. The analysis of the CLNN algorithm shows that the algorithm is not limited to two-process problem and can be extended to multi-process problem.

Acknowledgements

This study was funded through support from the National Key Basic Research and Development Program of China (Project Number: 2006CB701305), a grant from the State Key Laboratory of Resource and Environment Information System (Project Number: 088RA400SA) and a grant of the Knowledge Innovation Project of CAS.

Appendix A. The EM Algorithm to evaluate λ_1, λ_2 and P

Suppose $\theta = \{\lambda_1, \lambda_2, w\}$ represent the parameters of the mixture pdf (Eq. (4)). If we have a random sample $\{X = x_1, x_2, \ldots x_n\}$ of size n from the pdf of Eq. (4), then the likelihood is given by:

$$L(\theta|X) = \prod_{i=1}^{n} \{w f_{D_k}(x_i|k, \lambda_1) + (1-w) f_{D_k}(x_i|k, \lambda_2)\} \tag{5}$$

where k, w, λ_1 and λ_2 share the same meaning as those in Eq. (4).

Define

$$y_i = [x_i, \delta_i, (1-\delta_i)] \tag{6}$$

Thus the likelihood of y_i conditioning on θ is:

$$g(y_i|\theta) = \prod_{i=1}^{n} \left\{ \left[w f_{D_k}(x_i|k, \lambda_1)\right]^{\delta_i} \cdot \left[(1-w) f_{D_k}(x_i|k, \lambda_2)\right]^{1-\delta_i} \right\} \tag{7}$$

Hence we can write

$$\log g(y_i|\theta) = \sum_{i=1}^{n} \{\delta_i \log[w f_{D_k}(x_i|k, \lambda_1)] + (1-\delta_i) \log[(1-w) f_{D_k}(x_i|k, \lambda_2)]\} \tag{8}$$

The missing data (δ_i ($i = 1, 2, \ldots n$)) and the parameters (θ) can be derived through the EM algorithm, which is divided into two steps: the Expectation step (the E-step) and the Maximization step (the M-step).

The E-step in this context is:

$$E\left(\hat{\delta}_i^{(t+1)}\right) = \frac{\hat{w}^{(t)} f_{D_k}(x_i; k, \hat{\lambda}_1^{(t)})}{\hat{w}^{(t)} f_{D_k}(x_i; k, \hat{\lambda}_1^{(t)}) + (1-\hat{w}^{(t)}) f_{D_k}(x_i; k, \hat{\lambda}_2^{(t)})} \tag{9}$$

while the M-step is:

$$\hat{\lambda}_1^{(t+1)} = \frac{k \sum_{i=1}^{n} \hat{\delta}_i^{(t+1)}}{\pi \sum_{i=1}^{n} x_i^2 \hat{\delta}_i^{(t+1)}}$$

and $\hat{\lambda}_2^{(t+1)} = \frac{k \sum_{i=1}^{n} \left(1-\hat{\delta}_i^{(t+1)}\right)}{\pi \sum_{i=1}^{n} x_i^2 \left(1-\hat{\delta}_i^{(t+1)}\right)}$ with

$$w^{(t+1)} = \sum_{i=1}^{n} \hat{\delta}_i^{(t+1)} / n.$$

where n is the number of points, t is the number of iterations, and x_i is the independent variable representing the kth nearest distance of a given point q_i ($i = 1,2, \ldots n$). If we define the component with λ_1 as the feature, then $\hat{\delta}_i^{(t+1)}$ is the membership value belonging to the feature after $t+1$ iterations. That is, q_i can be classified as feature if $\hat{\delta}_i^{(t+1)}$ is not less than 0.5 while q_i can be classified as noise if $\hat{\delta}_i^{(t+1)}$ is less than 0.5. A detailed discussion about the EM algorithm can be found in (Celeux & Govaert, 1992; Moon, 1996; Byers & Raftery, 1998).

Please cite this article in press as: Pei, T., et al. Detecting feature from spatial point processes using Collective Nearest Neighbor. *Computers, Environment and Urban Systems* (2009), doi:10.1016/j.compenvurbsys.2009.08.001

References

Agrawal, R., Gehrke, J., Gunopulos, D., & Raghavan, P. (1998). Automatic subspace clustering of high dimensional data for data mining applications. In *Proceedings of 1998 ACM-SIGMOD international conference on management of data* (pp. 94–105). New York: ACM Press.

Ankerst, M., Breunig, M. M., Kriegel, H.-P., & Sander, J. (1999). OPTICS: Ordering points to identify the clustering structure. In *Proceedings of ACM-SIGMOD'99 international conference on management data* (pp. 46–60). USA: Philadelphia.

Boots, B., & Getis, A. (1988). *Point pattern analysis*. Beverly Hills, CA: Sage Publications.

Byers, S. D., & Raftery, A. E. (1998). Nearest-neighbor clutter removal for estimating features in spatial point processes. *Journal of the American Statistical Association, 93*, 577–584.

Celeux, G., & Govaert, G. (1992). A classification EM algorithm and two stochastic versions. *Computational Statistics and Data Analysis, 14*, 315–332.

Chainey, S., & Ratcliffe, J. (2005). *GIS and crime mapping (mastering GIS: Technol, applications & mgmnt)*. New York: John Wiley & Sons, Inc..

Chen, Y., Liu, J., & Ge, H. K. (1999). Pattern characteristics of foreshock sequences. *Pure and Applied Geophysics, 155*, 395–408.

China Seismograph Network Data Management Center. (2009). China Seismograph Network (CSN) Catalog: <http://www.csndmc.ac.cn> Accessed 05.01.09.

Cressie, N. A. C. (1991). *Statistics for spatial data* (1st ed.). New York: John Wiley & Sons, Inc..

Dasgupta, A., & Raftery, A. E. (1998). Detecting features in spatial point processes with clutter via model-based clustering. *Journal of the American Statistical Association, 93*, 294–302.

Ertoz, L., Steinbach, M., & Kumar, V. (2002). A new shared nearest neighbor clustering algorithm and its applications, In *Proceeding of Workshop on Clustering High Dimensional Data and its Applications* (pp. 105–115), Arlington, Va., USA.

Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceeding of 2nd International Conference on Knowledge Discovery and Data Mining* (pp. 226–231), Portland, OR.

Feng, H., & Huang, D. Y. (1980). *A catalogue of earthquake in western China (1970–1975, $M \geqslant 1$)*. Beijing: Seismological Press (in Chinese).

Feng, H., & Huang, D. Y. (1989). *Earthquake catalogue in west China (1976–1979, $M \geqslant 1$)*. Beijing: Seismological Press (in Chinese).

Fraley, C., & Raftery, A. E. (2003). Enhanced model-based clustering, density estimation, and discriminant analysis software: MCLUST. *Journal of Classification, 20*, 263–286.

Fraley, C., & Raftery, A. E. (1998). How many clusters? Which clustering method? Answers via model-based cluster analysis. *The Computer Journal, 41*, 578–588.

Gaudart, J., Poudiougou, B., Dicko, A., Ranque, S., Toure, O., Sagara, I., et al. (2008). Space–time clustering of childhood malaria at the household level: A dynamic cohort in a Mali village. *BMC Public Health, 6*, 1–13.

Hodge, V. J., & Austin, J. (2004). A survey of outlier detection methodologies. *Artificial Intelligence Review, 22*, 85–126.

Illian, J., Penttinen, A., Stoyan, H., & Stoyan, D. (2008). *Statistical analysis and modeling of spatial point patterns*. West Sussex: John Wiley & Sons Ltd. (Section 1.5).

Kulldorff, M., & Nagarwalla, N. (1995). Spatial disease clusters: Detection and inference. *Statistics in Medicine, 14*, 799–810.

Kulldorff, M. (1997). A spatial scan statistic. *Communications in Statistics – Theory and Methods, 26*, 1481–1496.

Kulldorff, M., Heffernan, R., Hartman, J., Assuncao, R., & Mostashari, F. (2005). A space–time permutation scan statistic for disease outbreak detection. *Plos Medicine, 2*, 216–224.

Lawson, B. A. (2001). *Statistical methods in spatial epidemiology* (1st ed.). New York: John Wiley & Sons.

Lin, C. Y., & Chang, C. C. (2005). A new density-based scheme for clustering based on genetic algorithm. *Fundamenta Informaticae, 68*, 315–331.

Lucio, P. S., & Brito, N. L. C. (2004). Detecting randomness in spatial point patterns: A 'stat-geometrical' alternative. *Mathematical Geology, 36*, 79–99.

Jarvis, R. A., & Patrick, E. A. (1973). Clustering using a similarity measure based on shared nearest neighbors. *IEEE Transactions on Computers, C-22*, 1025–1034.

Jiao, Y. B., Wu, K. T., & Yang, M. D. (1990). Assessment about capability and quality of our country earthquake observation network. *Earthquake Research in China, 6*, 1–7 (in Chinese).

Matsu'ura, R. S., & Karakama, I. (2005). A point-process analysis of the Matsushiro earthquake swarm sequence. The effect of water on earthquake occurrence. *Pure and Applied Geophysics, 162*, 1319–1345.

Moon, T. K. (1996). The expectation–maximization algorithm. *IEEE Signal Processing Magazine, 13*, 47–60.

Nagesh, H., Goil, S., & Choudhary, A. (1999). Mafia: Efficient and scalable subspace clustering for very large data sets, Technical report TR #9906-010, Northwestern University, 1999, <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.36.8684> Accessed January 2009.

Openshaw, S. (1996). Using a geographical analysis machine to detect the presence of spatial clusters and the location of clusters in synthetic data. In F. E. Alexander & P. Boyle (Eds.), *Methods for investigating localised clustering of disease IARC scientific publication no 135* (pp. 68–87). France: Lyon.

Openshaw, S., Charlton, M., Wymer, C., & Craft, A. W. (1987). A mark I geographical analysis machine for the automated analysis of point data sets. *International Journal of Geographical Information Systems, 1*, 335–358.

Openshaw, S., Charlton, M., Craft, A. W., & Birth, J. M. (1988). An investigation of leukaemia clusters by the use of a geographical analysis machine. *Lancet, I*, 272–273.

Pei, T., Yang, M., Zhang, J. S., Zhou, C. H., Luo, J. C., & Li, Q. L. (2003). Multi-scale expression of spatial activity anomalies of earthquakes and its indicative significance on the space and time attributes of strong earthquakes. *Acta Seismologica Sinica, 3*, 292–303.

Pei, T., Zhu, A. X., Zhou, C. H., Li, B. L., & Qin, C. Z. (2006). A new approach to the nearest-neighbour method to discover cluster features in overlaid spatial point processes. *International Journal of Geographical Information Science, 20*, 153–168.

Pei, T., Zhu, A. X., Zhou, C. H., Li, B. L., & Qin, C. Z. (2007). Delineation of support domain of feature in the presence of noise. *Computers and Geosciences, 33*, 952–965.

Reasenberg, P. A. (1999). Foreshock occurrence rates before large earthquakes worldwide. *Pure and Applied Geophysics, 155*, 355–379.

Ripepe, M., Piccinini, D., & Chiaraluce, L. (2000). Foreshock sequence of September 26th, 1997 Umbria–Marche earthquakes. *Journal of Seismology, 4*, 387–399.

Ripley, B. D. (1987). Spatial point pattern analysis in ecology. In P. Legendre & L. Legendre (Eds.), *Developments in numerical ecology*. *NATO AS1 Series* (Vol. G14, pp. 407–429). Berlin: Springer-Verlag.

Rogerson, P. A. (2001). A statistical method for the detection of geographic clustering. *Geographical Analysis, 33*, 215–227.

Roy, S., & Bhattacharyya, D. K. (2005). An approach to find embedded clusters using density based techniques. *Lecture Notes in Computer Science, 3816*, 523–535.

Skellam, J. G. (1952). Studies in statistical ecology, I: Spatial pattern. *Biometrika, 39*, 346–362.

Umino, N., Okada, T., & Hasegawa, A. (2002). Foreshock and aftershock sequence of the 1998 $M \geqslant 5.0$ Sendai, northeastern Japan, earthquake and its implications for earthquake nucleation. *Bulletin of the Seismological Society of America, 92*, 2465–2477.

Wu, K. T., Jiao, Y. B., Lu, P. L., & Wang, Z. D. (1990). *Panorama of seismic sequence*. Beijing: University Press (in Chinese).

Wyss, M., & Toya, Y. (2000). Is background seismicity produced at a stationary Poissonian rate. *Bulletin of the Seismological Society of America, 90*, 1174–1187.

Wang, W., Yang, J., & Muntz, R. (1997). STING: A statistical information grid approach to spatial data mining. In *Proceeding of the 23rd international conference on very large data bases (VLDB'97)* (pp. 186–195), Athens, Greece, August.

Yan, P., & Clayton, M. K. (2006). A cluster model for space–time disease counts. *Statistics in Medicine, 25*, 867–881.

Yamada, I., & Thill, J.-C. (2007). Local indicators of network-constrained clusters in spatial point patterns. *Geographical Analysis, 39*, 268–292.

Yang, T. Y., & Lee, J. C. (2007). Bayesian nearest-neighbor analysis via record value statistics and nonhomogeneous spatial Poisson processes. *Computational Statistics and Data Analysis, 51*, 4438–4449.

Zhang, Z. C. (1990). *Earthquake cases in China (1976–1980)*. Beijing: Seismological Press (in Chinese).

Zhuang, J. C., Chang, C. P., Ogata, Y., & Chen, Y. L. (2005). A study on the background and clustering seismicity in the Taiwan region by using point process models. *Journal of Geophysical Research – Solid Earth, 110*. doi:10.1029/2004JB003157. B05S18.

Please cite this article in press as: Pei, T., et al. Detecting feature from spatial point processes using Collective Nearest Neighbor. *Computers, Environment and Urban Systems* (2009), doi:10.1016/j.compenvurbsys.2009.08.001