International Journal of Computational Intelligence Systems
Vol. **14(1)**, 2021, *pp.* 140–147
DOI: https://doi.org/10.2991/ijcis.d.201102.002; ISSN: 1875-6891; eISSN: 1875-6883
https://www.atlantis-press.com/journals/ijcis/

Research Article

Semi-Supervised Density Peaks Clustering Based on Constraint Projection

Shan Yan, Hongjun Wang*, Tianrui Li, Jielei Chu, Jin Guo

School of Information Science and Technology, Southwest Jiaotong University, Chengdu, Sichuan, China

ARTICLE INFO

Article History
Received 15 Jul 2020
Accepted 22 Oct 2020

Keywords
Semi-supervised learning
Density peaks clustering
Pairwise constraint
Constraint projection

ABSTRACT

Clustering by fast searching and finding density peaks (DPC) method can rapidly identify the centers of clusters which have relatively high densities and high distances according to a decision graph. Various methods have been introduced to extend the DPC model over the past five years. DPC was originally presented as an unsupervised learning algorithm, and the thought of adding some prior information to DPC emerges as an alternative approach for improving its performance. It is extravagant to collect labeled data in real applications, and annotation of class labels is a nontrivial work, while pairwise constraint information is easier to get. Furthermore, the class label information can be converted into pairwise constraint information. Thus, we can take full advantage of pairwise constraints (or prior information) as much as possible. So this paper presents a new semi-supervised density peaks clustering algorithm (SSDPC) that uses constraint projection, which is flexible in loosening a few constraints over the learning stage. In the first stage, instances involving instance-level constraints and the remaining instances are concurrently projected to a lower dimensional data space led by the pairwise constraints, where viewing the distribution of data instances more clearly is available. Subsequently, traditional DPC is executed on the new lower dimensional dataset. Lastly, a few datasets from the Microsoft Research Asia Multimedia (MSRA-MM) image and UCI machine learning repository datasets are adopted in the experimental validation. The experimental results demonstrate that the proposed SSDPC achieves better performance than other three semi-supervised clustering algorithms.

1. INTRODUCTION

The swift advancement in information technology has caused a massive increase in the amount of data generated. As storage becomes more affordable, we will continue to witness rapid growth. How much of this data makes sense to a naive user remains a challenge to researchers in fields such as data mining. As much of the data is not labeled, annotated, or captioned. One tool that has been used by the research communities to attempt to organize such mixed data is clustering. Clustering is grouping of objects into classes according to objects' similarities [1]. These classes exhibit close intra-class similarity and wide interclass difference, and much of this process is performed in an unsupervised way [2], where clustering in its original form takes mixed data that is unlabeled and attempts to categorize it [3].

Clustering is a useful technique because when datasets are apportioned into groups based on data homogeneity, we can hence give labels to such small groups. The process can adapt to changes and classify groups using suitable features [1]. Features describing each object are all the algorithm has available to separate the objects. Clustering can pinpoint sparse and dense areas in object space, and we can easily observe the entire distribution shapes and find any correlations in data marks. It can also be a preprocessing input to other algorithms, like classification and feature subset selection, which can conveniently work on the identified groups.

The most common approach in the use of clustering algorithms is with unlabeled data and without a tutor; it is therefore unsupervised. The input is simply a collection of data instances to be clustered based on some conceptual relationship. Unsupervised clustering can be greatly improved by some supervision, and numerous algorithms have been recommended to enhance the quality of clustering by employing supervision.

The clustering algorithms are not given any information at all, yet the experimenter in the real-world application domain may have some clues regarding the dataset that could be useful to the algorithm, for example, where to place each instance within the partition. Traditional unsupervised clustering algorithms cannot benefit from such information when it does exit [2]. We therefore are interested in clustering that requires some minimal involvement by the users and we refer to it as semi-supervised clustering.

Such supervision could be to place constraints either to revise the cost function or to master the distance and similarity measures [4]. This type of supervision is generally known as semi-supervised learning, it involves acquiring knowledge from labeled and at the same time unlabeled data by computers and natural systems like human beings [5–7]. This method is better than unsupervised and supervised learning as long as it offers improved performance and

*Corresponding author. Email: wanghongjun@swjtu.edu.cn

accuracy [5]. The system of semi-supervised learning works both for classification and clustering [8], hence the capability of unsupervised clusters can improve with little amounts of supervision by way of labels on the data or constraints [2]. Studies on semi-supervised clustering show that it is much more effective than unsupervised classification techniques [9–13].

Semi-supervised learning has been researched under models such as graph-based methods, mixture models, self-training, multi-view learning and co-training. The recent algorithms of semi-supervised clustering generalizes to two types: constraint and distance based. Constraint-based processes operated by users providing labels or constraints to control the algorithms to even more correct data separation [4,14]. This is achieved by tuning the objective function for gauging clustering such that it satisfies the constraints [4,15]. The constraints can be enforced through the clustering steps [2], or the clustering can be initialized and then constrained based on labeled examples [16]. Constraint-based semi-supervised clustering could, for example, use pairwise constraints, where two instances are grouped in the same or different bundles [4].

Pairwise constraint is a type of supervision information that specifies whether two instances of data locate in the same group (a must-link [ML]) or separate groups (a cannot-link [CL]). The use of pairwise constraints from a practical perspective is often obvious selection in certain applications, meaning they are gathered spontaneously alongside unlabeled data [17,18]. For example, the co-occurring protein information of Interacting Proteins. The dataset can be taken as the obvious ML constraints in gene data clustering [18,19]. Notably, semi-supervised learning methods have unlocked access to the use of constraints in clustering, and consequently, with selection of active and effective pairwise constraints, clustering can be improved by specifying similarities between pairs of instances [14]. In fact, semi-supervised clustering can be referred as constrained clustering, since the supervision information is provided by the pairwise constraints [18]. So one has to consider the semi-supervised clustering problem as one where it is known that by varying the degree of inevitability, some sample pairs are (or are not) in the same class [20]. For label-breeding algorithms, the available labels are disseminated to unlabeled points whereas the available labels are changed to pairwise constraints, in constrained-clustering algorithms, then a controlled cut is made as a tradeoff between the cut cost minimization and the constraint satisfaction maximization [21]. The use of partly labeled information as pairwise constraints has been investigated by Nguyen and Caruana [22].

The density peaks clustering (DPC) algorithm is a clustering method for finding the clusters centers quickly. However, DPC was originally introduced as an unsupervised learning algorithm. As mentioned above, obtaining constraints from a dataset is achievable. Motivated by the DPC algorithm, the idea of adding some prior information to DPC emerges as an alternative means to improve its performance. In this paper, we introduce pairwise constraints to DPC and construct a framework of semi-supervised density peaks clustering (SSDPC). Pairwise constraint information is applied to guide projecting the original data. The data points then can be observed more obviously in the lower dimensional data space, and making use of some domain knowledge will enhance the clustering effect.

The rest of the paper is arranged as follows: Section 2 reports related works. Section 3 introduces constraint projection and the proposed SSDPC algorithm. Section 4 is the experimental procedures and results. Thereafter, the conclusions is drawn in Section 5.

2. RELATED WORKS

2.1. Clustering by Fast Search and Find of Density Peaks

Rodriguez and Laio proposed an advanced clustering algorithm by identifying density peaks (DPC) [23]. The concept of the DPC algorithm is simply based on distinguishing cluster centers from their neighbors by higher densities and comparatively large distances from other points with higher densities. The method uses the local density ρ and the distance δ of point $\boldsymbol{x}$ to locate class centers.

These assumptions involve features of cluster centers: in other words, cluster centers have neighbors with lower local densities and they are placed comparatively far from the points with high densities. According to [23], for a given dataset $X = [\boldsymbol{x}_1, \boldsymbol{x}_2, .., \boldsymbol{x}_n]$ where n is the quantity of data points, then the distance matrix between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ is constructed depend on the Euclidean distance of the data points.

$$d(\boldsymbol{x}_i, \boldsymbol{x}_j) = \| \boldsymbol{x}_i - \boldsymbol{x}_j \| \tag{1}$$

Here $\| \cdot \|$ denotes a 2-norm and in the next steps, we can compute the local density ρ_i and distance δ_i of point $\boldsymbol{x}_i$ as follows:

$$\rho_i = \sum_{j \neq i} \chi(d(\boldsymbol{x}_i, \boldsymbol{x}_j) - d_c) \tag{2}$$

Note that d_c is the cutoff distance. d_c is the neighborhood range of data point i and ρ_i. It specifies the quantity of points adjacent to the data point $\boldsymbol{x}_i$. However, the implementation code that was presented has another value of ρ_i, given by the Gaussian Kernel function.

$$\rho_i = \sum_{j \neq i} exp\left(- \left(\frac{d_{ij}}{d_c} \right)^2 \right) \tag{3}$$

In Eq. (3), d_c is a controllable parameter to handle the weight degradation rate and d_c is called the soft threshold while Eq. (2) represents the hard threshold. The lowest distance between the data point $\boldsymbol{x}_i$ and any other point with higher density δ_i is obtained by the equation below.

$$\delta_i = \begin{cases} \min\limits_{j:\rho_i > \rho_j} (d_{ij}), & \text{if } \exists j, \text{ s.t. } \rho_i > \rho_j \\ \max\limits_{j} (d_{ij}), & \text{otherwise} \end{cases} \tag{4}$$

Cluster centers are considered to be points with high ρ_i and δ_i, and these points are known as the peaks. These cluster center points have higher densities than other points. Any other point therefore belongs in the same cluster like its nearest neighbor peak. Once every cluster center is spotted, the algorithm attaches the rest of the points to the same cluster like their nearest neighbor with larger density. A plot of the distance δ_i and the local density ρ_i of all the points helps in making decisions on d_c and the cluster centers. Sometimes producing an estimate based on the density peaks is very difficult in situations when the quantity of clusters is up to the quantity of objects in the dataset. In the case of sparse data points, the number of peaks is obscure, so plotting $\rho_i \times \delta_i$ arranged

in a decreasing order is used to choose the number of clusters. It is denoted as γ_i [23]. The distance δ_i is given by $\max_j(d_{ij})$ for the points with the high density. It is clear the points that possess local or global maximum density have substantial δ_i. There can exist some characteristics where for example, (1) a point with low ρ and low δ implies that the point is placed in the boundary of a cluster, (2) a point with high ρ and low δ means the point is near to a cluster center, (3) a point with low ρ and high δ specifies the point is distant from any clusters and can be an outlier or noise. Therefore only the points having both high ρ and large δ are candidates of cluster centers.

The DPC algorithm described above indicates the process for a single step; in other words, no iterations are involved and there are a few parameters to initialize. Accordingly a variety of extended DPC algorithms are demonstrated in recent studies [24–32], in which part of the shortcomings in DPC algorithm are upgraded.

2.2. Pairwise Constraint

Pairwise constraint is an exemplary approach of using prior information of datasets as injecting labels into clustering, typically in the manner of *ML* and *CL* pairwise constraints. A given pair of data points in *ML* specifies they should in the same group, while they are in *CL* shows the pair of data points are from different groups. Abounding of semi-supervised clustering techniques exploit pairwise constraints as prior knowledge in [18,33–38].

For a dataset X, the pairwise constraints (ML set and CL) set are expressed as follows:

1. $ML = \{(\boldsymbol{x}_i, \boldsymbol{x}_j)|l_i = l_j, \forall i, j\}$(data points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are placed in the same group);
2. $CL = \{(\boldsymbol{x}_i, \boldsymbol{x}_j)|l_i \neq l_j, \forall i, j\}$(data points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are placed in different groups).

where l is the label of data point $\boldsymbol{x}$.

Pairwise constraint is symmetric,

$(\boldsymbol{x}_i, \boldsymbol{x}_j) \in ML \Rightarrow (\boldsymbol{x}_j, \boldsymbol{x}_i) \in ML$

$(\boldsymbol{x}_i, \boldsymbol{x}_j) \in CL \Rightarrow (\boldsymbol{x}_j, \boldsymbol{x}_i) \in CL$

It is also transitive,

$(\boldsymbol{x}_i, \boldsymbol{x}_j) \in ML \& (\boldsymbol{x}_j, \boldsymbol{x}_k) \in ML \Rightarrow (\boldsymbol{x}_i, \boldsymbol{x}_k) \in ML$

$(\boldsymbol{x}_i, \boldsymbol{x}_j) \in CL \& (\boldsymbol{x}_j, \boldsymbol{x}_k) \in CL \Rightarrow (\boldsymbol{x}_i, \boldsymbol{x}_k) \in CL$

3. CONSTRAINT PROJECTION FOR SSDPC

3.1. Proposed Constraint Projection Model

Given a dataset $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n | \forall \boldsymbol{x}_i \in R^p\}$. p is the dimension of attributes of data point $\boldsymbol{x}_i$. The label of each data point $l_i \in \{c_1, c_2, ..., c_h\}, i = 1, 2, ..., n$, c_g is the category symbol of cluster $\boldsymbol{C}_g, g = 1, 2, ..., h$. h indicates there are h clusters in the dataset. For a data point, the information in each dimension represents different attributes. Considering the importance of each attribute is not equivalent, this requires selection of the key attributes. Constraint projection implies choosing the crucial q dimensions, which are used to reduce the p-dimensional original data to q-dimensional data by determining a projective matrix $\boldsymbol{W}_{p\times q} = [\boldsymbol{w}_1, \boldsymbol{w}_2, ..., \boldsymbol{w}_q], \boldsymbol{w}^T\boldsymbol{w} = \boldsymbol{I}$. The data points in lower dimensional space will be represented as $\boldsymbol{z}_i = \boldsymbol{W}^T\boldsymbol{x}_i$. The data after projection must maintain the relationship between pairwise constraints of the original data. That is, points involving the *ML* set must be close, while points involving the *CL* set must be distant.

The definition of the objective function is to maximize $J(\boldsymbol{W})$.

$$J(\boldsymbol{W}) = \frac{1}{2n_c}\sum_{(\boldsymbol{x}_i,\boldsymbol{x}_j)\in CL} \| \boldsymbol{W}^T\boldsymbol{x}_i - \boldsymbol{W}^T\boldsymbol{x}_j \|^2 - \frac{d}{2n_m}\sum_{(\boldsymbol{x}_i,\boldsymbol{x}_j)\in ML} \| \boldsymbol{W}^T\boldsymbol{x}_i - \boldsymbol{W}^T\boldsymbol{x}_j \|^2 \tag{5}$$

where n_c is the cardinal number of the cannot-link set *CL* and n_m is the cardinal number of the must-link set *ML*. d is a scaling parameter, whose role is to counteract the contributions of distances in *CL* set and *ML* set. The parameter d can be estimated by Eq. (6)

$$d = \frac{\frac{1}{n_c}\sum_{(\boldsymbol{x}_i,\boldsymbol{x}_j)\in CL} \| \boldsymbol{x}_i - \boldsymbol{x}_j \|^2}{\frac{1}{n_m}\sum_{(\boldsymbol{x}_i,\boldsymbol{x}_j)\in ML} \| \boldsymbol{x}_i - \boldsymbol{x}_j \|^2} \tag{6}$$

As we usually use only a portion of the pairwise constraint information, it is not necessary to calculate all pairwise distances in the constraint sets *CL* and *ML*, but just the distances of the portion of constraints that are utilized.

3.2. Inference

The purpose of maximizing $J(\boldsymbol{W})$ is to obtain a solution set of vectors $\boldsymbol{W}_{p\times q} = [\boldsymbol{w}_1, \boldsymbol{w}_2, ..., \boldsymbol{w}_q]$. The problem involving the maximization of $J(\boldsymbol{W})$ is a typical Eigen-vector/value problem.

$$\begin{aligned}
J(\boldsymbol{W}) &= \frac{1}{2n_c}\sum_{(\boldsymbol{x}_i,\boldsymbol{x}_j)\in CL} \| \boldsymbol{W}^T\boldsymbol{x}_i - \boldsymbol{W}^T\boldsymbol{x}_j \|^2 \\
&\quad - \frac{d}{2n_m}\sum_{(\boldsymbol{x}_i,\boldsymbol{x}_j)\in ML} \| \boldsymbol{W}^T\boldsymbol{x}_i - \boldsymbol{W}^T\boldsymbol{x}_j \|^2 \\
&= \frac{1}{2n_c}\sum_{(\boldsymbol{x}_i,\boldsymbol{x}_j)\in CL} \left[\boldsymbol{W}^T(\boldsymbol{x}_i - \boldsymbol{x}_j)\right]^T \cdot \left[\boldsymbol{W}^T(\boldsymbol{x}_i - \boldsymbol{x}_j)\right] \\
&\quad - \frac{d}{2n_m}\sum_{(\boldsymbol{x}_i,\boldsymbol{x}_j)\in ML} \left[\boldsymbol{W}^T(\boldsymbol{x}_i - \boldsymbol{x}_j)\right]^T \cdot \left[\boldsymbol{W}^T(\boldsymbol{x}_i - \boldsymbol{x}_j)\right] \\
&= \frac{1}{2n_c}\sum_{(\boldsymbol{x}_i,\boldsymbol{x}_j)\in CL} trace\left(\boldsymbol{W}^T(\boldsymbol{x}_i - \boldsymbol{x}_j) \cdot \left[\boldsymbol{W}^T(\boldsymbol{x}_i - \boldsymbol{x}_j)\right]^T\right) \\
&\quad - \frac{d}{2n_m}\sum_{(\boldsymbol{x}_i,\boldsymbol{x}_j)\in ML} trace\left(\boldsymbol{W}^T(\boldsymbol{x}_i - \boldsymbol{x}_j) \cdot \left[\boldsymbol{W}^T(\boldsymbol{x}_i - \boldsymbol{x}_j)\right]^T\right) \\
&= trace\left(\boldsymbol{W}^T\left(\frac{1}{2n_c}\sum_{\boldsymbol{x}_i,\boldsymbol{x}_j\in CL}(\boldsymbol{x}_i - \boldsymbol{x}_j)(\boldsymbol{x}_i - \boldsymbol{x}_j)^T \right.\right. \\
&\quad \left.\left. - \frac{d}{2n_m}\sum_{\boldsymbol{x}_i,\boldsymbol{x}_j\in ML}(\boldsymbol{x}_i - \boldsymbol{x}_j)(\boldsymbol{x}_i - \boldsymbol{x}_j)^T\right)\boldsymbol{W}\right)
\end{aligned} \tag{7}$$

For simplicity, $\boldsymbol{H}_C$ and $\boldsymbol{H}_M$ are defined as follows:

$$\boldsymbol{H}_C = \frac{1}{2n_c} \sum_{(\boldsymbol{x}_i, \boldsymbol{x}_j) \in CL} (\boldsymbol{x}_i - \boldsymbol{x}_j)(\boldsymbol{x}_i - \boldsymbol{x}_j)^T \tag{8}$$

$$\boldsymbol{H}_M = \frac{1}{2n_m} \sum_{(\boldsymbol{x}_i, \boldsymbol{x}_j) \in ML} (\boldsymbol{x}_i - \boldsymbol{x}_j)(\boldsymbol{x}_i - \boldsymbol{x}_j)^T \tag{9}$$

Then Eq. (7) can be edited as

$$J(\boldsymbol{W}) = trace(\boldsymbol{W}^T(\boldsymbol{H}_C - d\boldsymbol{H}_M)\boldsymbol{W}) \tag{10}$$

Eq. (10) indicates that we can solve the problem in Eq. (5) by computing the top q eigenvalues of $\boldsymbol{H}_C - d\boldsymbol{H}_M$ and the corresponding eigenvectors. Suppose the first q eigenvalues of $\boldsymbol{H}_C - d\boldsymbol{H}_M$ are $\lambda_1 \geqslant \lambda_2 \geqslant \ldots \lambda_q$, and $\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots \boldsymbol{w}_q$ are corresponding eigenvectors. The projective solution can be expressed as $\boldsymbol{W} = [\boldsymbol{w}_1, \boldsymbol{w}_2, ..., \boldsymbol{w}_q]$.

Denote $\boldsymbol{\Lambda} = diag(\lambda_1, \lambda_2, ..., \lambda_q)$, Eq. (10) then will be reformulated as

$$trace(\boldsymbol{W}^T(\boldsymbol{H}_C - d\boldsymbol{H}_M)\boldsymbol{W}) = trace(\boldsymbol{\Lambda}) = \sum_{i=1} \lambda_i \tag{11}$$

From Eq. (11), it is clear that Eq. (7) will achieve the maximum value when the set of eigenvalues $(\lambda_1, \lambda_2, ..., \lambda_q)$ includes all the nonpositive eigenvalues of $\boldsymbol{H}_C - d\boldsymbol{H}_M$. Note that $\lambda_i = 0$ will not contribute to maximizing Eq. (11). In order to avoid losing too much of the characteristic information, all the nonnegative eigenvalues λ are chosen. From this process, $\boldsymbol{H}_C$ and $\boldsymbol{H}_M$ are positive semi-definite.

3.3. Semi-Supervised Density Peaks Clustering Algorithm

Figure 1 shows the flow of the presented SSDPC.

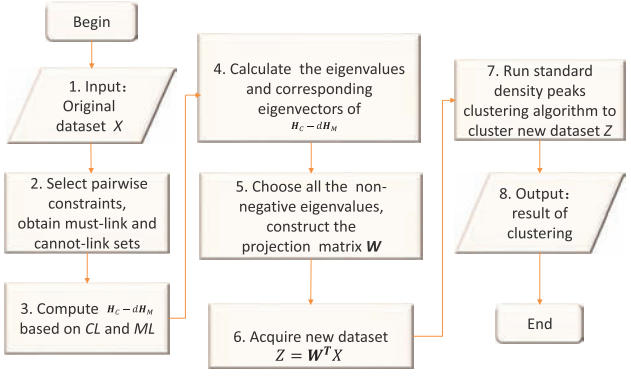

Figure 1. Depiction of the semi-supervised density peaks clustering (SSDPC) algorithm. Steps 1–6 involve the constraint projection of the original dataset X led by pairwise constraints, and steps 7–8 are clustering processes.

Based on the depiction in Figure 1, the SSDPC algorithm is expressed as follows:

From the process of SSDPC algorithm, we can get complexity of SSDPC model. The time complexity is $O(n^2)$, the space complexity is $O(n^2)$, n is the number of instances in the dataset.

SSDPC algorithm:

Input: Dataset $\{\boldsymbol{x_i}\}_{i=1}^n \in R^p$, $\boldsymbol{x_i}$ is the data point.

1. Establish the cannot-link set CL and must-link set ML according to the part of existing labels or other field information.
2. Calculate matrix $\boldsymbol{H}_C - d\boldsymbol{H}_M$ according to Eqs. (8) and (9).
3. Compute all the eigenvalues $\boldsymbol{\lambda}$ and corresponding eigenvectors $\boldsymbol{w}$ of matrix $\boldsymbol{H}_C - d\boldsymbol{H}_M$.
4. In order to maximize the objective function $J(\boldsymbol{W})$ in Eq. (5), all the non-negative eigenvalues and corresponding eigenvectors are selected.
5. Use the selected eigenvectors to set up the projection matrix $\boldsymbol{W} = [\boldsymbol{w}_1, \boldsymbol{w}_2, ..., \boldsymbol{w}_q]$.
6. Construct the new dataset Z depending on $Z = \boldsymbol{W}^T X$.
7. Run the standard density peaks clustering (DPC) algorithm on the projected dataset Z. Clustering procedures are completed.

Output: Clusters $\boldsymbol{C}_1, ..., \boldsymbol{C}_h$. $\boldsymbol{C}_g = \{i | \boldsymbol{x}_i \in \boldsymbol{C}_g\}$.

4. EXPERIMENTAL STUDIES

In this section, we describe the experiment settings and performance comparisons for our proposed algorithm SSDPC.

4.1. Datasets Setup

In this part, we investigate the performance of the SSDPC algorithm on a variety of datasets. In the experiments, 17 datasets are acquired from Microsoft Research Asia Multimedia (MSRA-MM) image datasets [39] and three datasets are picked up from the UCI machine learning repository [40]. The details of these 20 datasets are summarized in Table 1.

The MSRA-MM dataset was assembled from a commercial search engine with more than 1 million images and 20 thousand videos. The purpose of MSRA-MM is to promote research in the field of multimedia information retrieval and related areas. Seventeen image datasets are chosen for the assignment of semi-supervised density peaks clustering (DPC). Each image dataset embraces practically 1,000 instances with nearly 900 features.

UCI Machine Learning Repository currently maintains 507 datasets as a service to the machine learning communities for experimental studies of machine learning methods. New datasets are constantly supported by researchers' donations from all over the world. The most popular UCI datasets are Iris, Cancer, Wine, Breast, Heart Disease, Bank Marketing, Adult, Car Evaluation, Forest Fires, Wisconsin (Diagnostic), Human Activity Recognition Using Smartphones, Wine Quality, Poker Hand, and Abalone. In this empirical analysis, we pick credit approval, vertebral column, and congressional voting records from the UCI datasets. The particulars of each dataset, including classes, the quantity of instances, and features are tabulated in Table 1.

4.2. Experiments

To evaluate how the proposed SSDPC algorithm performs, three other contemporary semi-supervised learning algorithms, constrained k-means clustering with background knowledge (Copkmeans) [2], constrained 1-spectral clustering (COSC) [41], and semi-supervised clustering based on affinity propagation (Semi-AP) [42] are also implemented. All experiments are organized on a

Table 1 Characteristics, classes, the quantity of instances, and features in each dataset.

Dataset	Characteristic	Classes	Instances	Features
Ambulances	Real	3	930	892
Aquarium	Real	3	922	892
Balloon	Real	3	830	892
Bed	Real	3	888	892
Birthdaycake	Real	3	932	892
Blog	Real	3	943	892
Boat	Real	3	857	892
Bonsai	Real	3	867	892
Bread	Real	3	885	892
Bugat	Real	3	882	892
Building	Real	3	911	892
Bus	Real	3	910	892
Butterflytattoo	Real	3	738	892
Cactus	Real	3	919	892
Credit approval	Mixed	2	690	15
Vertebral column	Real	2	310	6
Vista	Real	3	799	899
Vistawallpaper	Real	3	799	899
Voituretuning	Real	3	879	899
Congressional voting records	Discrete	2	453	16

computer with Intel (R) Core (TM) i5-4460M CPU 3.20 GHz and 4.00 GB RAM, running Matlab R2010b. Each algorithm experiment is repeated 10 times on the 20 datasets.

The crucial modification of Cop-kmeans is to make sure none of the specified constraints are violated when updating cluster assignments. Cop-kmeans allots each point $\boldsymbol{x}_i$ to the closest cluster $\boldsymbol{C}_g$ as long as violate-constraints $(\boldsymbol{x}_i, \boldsymbol{C}_g, ML, CL)$ is false. The definition of violate-constraints (data point $\boldsymbol{x}$, cluster $\boldsymbol{C}$, must-link constraints ML, cannot-link constraints CL) is: (1) For any $(\boldsymbol{x}, \boldsymbol{x}_=) \in ML$: If $\boldsymbol{x}_= \notin \boldsymbol{C}$, return true. (2) For any $(\boldsymbol{x}, \boldsymbol{x}_{\neq}) \in CL$: If $\boldsymbol{x}_{\neq} \in \boldsymbol{C}$, return true. (3) Else, return false. COSC is able to guarantee all the given constraints are satisfied in the field of constraint spectral clustering. must-link constraints in ML are integrated via sparsification, while cannot-link constraints should be considered for the bi-partitioning scheme. In Semi-AP, the similarity matrix $[\boldsymbol{S}(i,k)_{n\times n}]$ is modified depending on pairwise constraints with several principles: (1) $(\boldsymbol{x}_i, \boldsymbol{x}_j) \in ML \Rightarrow s(i,j) = 0 \& s(j,i) = 0$. In case data point k conforms to $(\boldsymbol{x}_i, \boldsymbol{x}_k) \notin ML \& (\boldsymbol{x}_i, \boldsymbol{x}_j) \in ML \Rightarrow s(i,k) = 0$. Then $ML = (\boldsymbol{x}_i, \boldsymbol{x}_k) \cup ML$. (2) $(\boldsymbol{x}_i, \boldsymbol{x}_j) \in CL, \Rightarrow s(i,j) = -\infty \& s(j,i) = -\infty$. (3) If point k is connected to data points i and j individually, $(\boldsymbol{x}_i, \boldsymbol{x}_j) \notin \{ML \cup CL\} \Rightarrow s(i,j) = max(s(k,j), s(i,j) + s(i,k))$. (4) $(\boldsymbol{x}_i, \boldsymbol{x}_j) \notin \{ML \cup CL\} \& (\boldsymbol{x}_i, \boldsymbol{x}_k) \in CL \& (\boldsymbol{x}_k, \boldsymbol{x}_j) \in ML \Rightarrow s(i,j) = -\infty \& s(j,i) = -\infty$ or $(\boldsymbol{x}_i, \boldsymbol{x}_j) \notin \{ML \cup CL\} \& (\boldsymbol{x}_i, \boldsymbol{x}_k) \in ML \& (\boldsymbol{x}_k, \boldsymbol{x}_j) \in CL \Rightarrow s(i,j) = -\infty \& s(j,i) = -\infty$. Then, $CL = (\boldsymbol{x}_i, \boldsymbol{x}_j) \cup CL$, and an exemplar is obtained for each data point. They also make some adjustments to the affinity propagation cluster result for the condition when a constraint is violated.

Parameters for Cop-kmeans, COSC, and Semi-AP are all determined as described in the original studies [2,41,42].

The proposed algorithm SSDPC is implemented in accordance with the depiction in Figure 1. In this case, 10% of pairwise constraints information are applied to construct the ML and CL sets. The cutoff distance d_c satisfies the average number of neighbors is 2% of the total numbers of instances in the dataset. In comparison with Cop-kmeans, COSC and Semi-AP, SSDPC is more flexible in following the constraints we got.

For the evaluation measurement, we apply micro-averaged-precision (MAP) [43] to evaluate the accuracy of each clustering result. MAP is defined as follows:

$$MAP = \frac{\sum_h a_h}{\sum_h (a_h + b_h)}$$

where h is the quantity of clusters, a_h is the quantity of instances accurately allotted to cluster $\boldsymbol{C}_h$, and b_h is the number of instances incorrectly allotted to cluster $\boldsymbol{C}_h$. Generally, $0 \leqslant MAP \leqslant 1$, so the larger the value of MAP, the better the quality of the clustering algorithm.

4.3. Results and Discussion

In this part, results of our experiments are provided. Table 2 shows the accuracy results of the different algorithms on 20 datasets. Records are tabulated in terms of averaged mean accuracies and standard deviations over 10 repetitions of the experiments. The mean accuracies of the SSDPC algorithm are higher than the other three semi-supervised learning algorithms (Cop-kmeans, Semi-SC, Semi-AP) on the 20 different datasets, and the highest accuracy value is for the congressional voting records dataset, which reaches 0.8805. The proposed SSDPC algorithm also has the highest average accuracy for all datasets with an average MAP value of 0.5797.

To provide a robust comparison among the four algorithms, we carry out a $1 \times n$ comparison by way of the Friedman Aligned test [44]. The presented method SSDPC is the control algorithm. Table 3 reflects the aligned ranks and the aligned results in parentheses for the four algorithms and 20 datasets. From this table, SSDPC ranks first with an average rank value of 10.85, Cop-kmeans ranks second with average rank value of 38.35, Semi-AP ranks third with average rank value of 48.5, and COSC ranks last with average rank value of 64.3. The purpose of the Friedman Aligned Rank test is to analysis if the gauged sum of aligned ranks are notably different

Table 2 Performance of different algorithms in terms of accuracy on 20 datasets (mean ± std %).

Datasets	Cop-kmeans	COSC	Semi-AP	SSDPC
Ambulances	0.4512±0.0435	0.3967± 0.0371	0.4261± 0.0271	**0.5970**± 0.0536
Aquarium	0.4344± 0.0493	0.3989±0.0560	0.4055± 0.0270	**0.6505**±0.0731
Balloon	0.4536±0.0416	0.3796±0.0277	0.4589± 0.0452	**0.5530**± 0.0409
Bed	0.4468± 0.0592	0.4314± 0.0822	0.4337± 0.0418	**0.5680**± 0.0702
Birthdaycake	0.5004± 0.0264	0.3820± 0.0354	0.4923± 0.0257	**0.5599**± 0.0514
Blog	0.4467± 0.0590	0.4340± 0.0680	0.4060± 0.0351	**0.5883**± 0.0891
Boat	0.4288± 0.0585	0.4260± 0.0571	0.4097± 0.0347	**0.5718**± 0.0819
Bonsai	0.3978± 0.0393	0.3973± 0.0406	0.4321± 0.0334	**0.5608**± 0.0924
Bread	0.4660± 0.0225	0.4546± 0.0801	0.4460± 0.0513	**0.5786**± 0.0615
Bugat	0.4418± 0.0600	0.4206± 0.0568	0.4011± 0.0388	**0.5808**± 0.0814
Building	0.5568± 0.0608	0.4617± 0.0920	0.4651± 0.0619	**0.6422**± 0.0709
Bus	0.4700± 0.0384	0.4163± 0.0777	0.4293± 0.0359	**0.5522**± 0.0675
Butterflytattoo	0.6070± 0.0572	0.3737± 0.0204	0.5680± 0.0235	**0.6347**± 0.0634
Cactus	0.4868± 0.0753	0.4132± 0.0631	0.4771± 0.0574	**0.5929**± 0.0679
Credit approval	0.6162± 0.0142	0.5868 ± 0.0180	0.5619± 0.0007	**0.6446**±0.0268
Vertebral column	0.6955± 0.0216	0.6448± 0.0652	0.6697± 0.0436	**0.7177**± 0.0671
Vista	0.4426± 0.0410	0.3930± 0.0234	0.4464± 0.0291	**0.5601**± 0.0834
Vistawallpaper	0.4451± 0.0326	0.4126± 0.0814	0.4606± 0.0207	**0.5603**± 0.0652
Voituretuning	0.4207± 0.0288	0.4354± 0.1005	0.4243± 0.0365	**0.5994**± 0.0643
Congressional voting records	0.8678± 0.0141	0.5460± 0.0311	0.8618± 0.0138	**0.8805**± 0.0184
Average	0.5038	0.4402	0.4839	**0.5797**

COSC, constrained 1-spectral clustering; SSDPC, semi-supervised density peaks clustering; AP, affinity propagation.

Table 3 Aligned results of the four algorithms. The rank value in parentheses are employed in the calculation of the Friedman Aligned Rank test. The lowest rank value is the best one.

Datasets	Cop-kmeans	COSC	Semi-AP	SSDPC	Total
Ambulances	−0.0166(40)	−0.0711(74)	−0.0416(60)	0.1292(3)	177
Aquarium	−0.0380(55)	−0.0734(75)	−0.0668(71)	0.1782(1)	202
Balloon	−0.0077(33)	−0.0817(77)	−0.0024(31)	0.0917(13)	154
Bed	−0.0231(45)	−0.0386(56)	−0.0363(52)	0.0980(11)	164
Birthdaycake	0.0168(26)	−0.1017(78)	0.0086(29)	0.0762(19)	152
Blog	−0.0221(44)	−0.0347(51)	−0.0627(69)	0.1196(5)	169
Boat	−0.0303(47)	−0.0331(49)	−0.0494(64)	0.1127(7)	167
Bonsai	−0.0492(62)	−0.0497(65)	−0.0149(37)	0.1138(6)	170
Bread	−0.0203(43)	−0.0317(48)	−0.0403(57)	0.0923(12)	160
Bugat	−0.0193(42)	−0.0405(58)	−0.0600(68)	0.1197(4)	172
Building	0.0253(24)	−0.0697(73)	−0.0663(70)	0.1107(8)	175
Bus	0.0030(30)	−0.0507(66)	−0.0376(54)	0.0852(17)	167
Butterflytattoo	0.0612(21)	−0.1722(79)	0.0222(25)	0.0888(16)	141
Cactus	−0.0057(32)	−0.0794(76)	−0.0154(38)	0.1004(9)	155
Credit approval	0.0138(27)	−0.0156(39)	−0.0405(59)	0.0422(22)	147
Vertebral column	0.0135(28)	−0.0371(53)	−0.0123(35)	0.0358(23)	139
Vista	−0.0180(41)	−0.0675(72)	−0.0141(36)	0.0996(10)	159
Vistawallpaper	−0.0246(46)	−0.0570(67)	−0.0091(34)	0.0907(15)	162
Voituretuning	−0.0493(63)	−0.0346(50)	−0.0456(61)	0.1295(2)	176
Congressional voting records	0.0788(18)	−0.2430(80)	0.0728(20)	0.0914(14)	132
Total	767	1286	970	217	
Average rank	38.35	64.3	48.5	10.85	

COSC, constrained 1-spectral clustering; SSDPC, semi-supervised density peaks clustering; AP, affinity propagation.

from the total average aligned rank $R_j = \sum_{j=1}^{4} \hat{R}_j/4 = 810$ expected under the null hypothesis:

$$\sum_{j=1}^{4} \hat{R}_j^2 = 767^2 + 1286^2 + 970^2 + 217^2 = 3,230,074$$

$$\sum_{i=1}^{20} \hat{R}_i^2 = 177^2 + 202^2 + 154^2 ... + 132^2 = 529,658$$

In the above two formulas, $\hat{R}_j$ is the total aligned rank of the jth algorithm and $\hat{R}_i$ denotes the total aligned rank of the ith dataset. Accordingly, the Friedman Aligned Rank test statistic is determined as

$$T = \frac{(4-1)[3,230,074 - (4 \cdot 20^2/4)(4 \cdot 20 + 1)^2]}{\{[4 \cdot 20(4 \cdot 20 + 1)(2 \cdot 4 \cdot 20 + 1)]/6\} - (1/4)529,658} = 43.82$$

With four algorithms and 20 datasets, T is distributed in accordance with the chi-square distribution with $(4-1) = 3$ degrees of freedom. The p-value estimated by operating the $\chi^2(3)$ distribution is 1.64807×10^{-9}, which indicates the null hypothesis is repudiated at a notable level of significance. The p-value is much smaller

than 0.01 which implies the results of the algorithms are remarkably different.

5. CONCLUSION AND PROSPECTS

In this paper, we present a new SSDPC using pairwise constraints, as it is simpler to obtain pairwise constraint information than to acquire class tags (or labels). We consider applying pairwise constraint knowledge to project the original data onto a well-preserved lower dimensional space, which forces the distance between instances in ML pairs) to be decreased and the distance between instances in *CL* to be increased, resulting in a clearer observation of the instances. Pairwise constraints can be violated; it is not necessary for the clustering process to always satisfy constraints. Consequently, the novel SSDPC algorithm is a flexible semi-supervised clustering technique. Experiments over 20 datasets display that our SSDPC approach performs better than three other semi-supervised clustering methods.

As various data is achievable in daily life, for instance, signal lights data, railway track circuit data, diagnostic data, and so on. Applying the proposed SSDPC algorithm in analyzing different kinds of data is practicable. As we know, clustering by using density peaks is an efficient method when it was proposed in 2014 [23]. In this work, we extend DPC algorithm to a semi-supervised learning algorithm, the results show the semi-DPC algorithm is feasible. However, the algorithm is still robust with the value of cutoff distance d_c. A rule of thumb given by Rodriguez and Laio [23] is to choose d_c in a way that gives the average of neighbors between 1 and 2 % of the overall quantity of points in the dataset. We are going to do some research on how to choose an optimal distance d_c for a dataset in order to achieve better clustering performance. In general, extending a clustering method or introducing a new clustering approach is to get a valuable understanding of datasets we have.

CONFLICTS OF INTEREST

The authors declare they have no conflicts of interest.

AUTHORS' CONTRIBUTIONS

All authors contributed to this study. All authors read and approved the final manuscript.

ACKNOWLEDGMENTS

This research is supported by the National Natural Science Foundation of China for Youth (Grant No. 61703349), Science and Technology research and development project of China Railway Corporation (Grant No. N2018G062, K2018G011).

REFERENCES

[1] J.W. Han, M. Kamber, J. Pei, Data Mining: Concepts and Techniques, 3rd ed., Morgan Kaufmann, Waltham, MA, USA, (2012), pp. 444–448.

[2] K. Wagstff, C. Cardie, S. Rogers, Constrained k-means clustering with background knowledge, in Eighteenth International Conference on Machine Learning, Williamstown, MA, USA, 2001, pp. 577–584. https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.20.7363&rep=rep1&type=pdf

[3] A.K. Jain, M.N. Murty, P.J. Flynn, Data clustering: a review, ACM Comput. Sur. 31 (1999), 264–323.

[4] S. Basu, M. Bilenko, R.J. Mooney, A probabilistic framework for semi-supervised clustering, in Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, Seattle, WA, USA, 2004, pp. 59–68.

[5] V.J. Prakash, D.L. Nithya, A survey on semi-supervised learning techniques, Int. J. Comput. Trends Technol. 8 (2014), 25–29.

[6] M.R. Amini, P. Gallinari, The use of unlabeled data to improve supervised learning for text summarization, in Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, Tampere, Finland, 2002, pp. 105–112.

[7] Y. Mi, W. Liu, Y. Shi, J. Li, Semi-supervised concept learning by concept-cognitive learning and concept space, IEEE Trans. Knowl. Data Eng. (2020).

[8] N.N. Pise, P. Kulkarni, A survey of semi-supervised learning methods, in International Conference on Computational Intelligence & Security, IEEE, Suzhou, China, 2008, pp. 30–34.

[9] Y. Altun, D. McAllester, M. Belkin, Maximum margin semisupervised learning for structured variables, in Advances in Neural Information Processing Systems, Vancouver, Canada, 2006, pp. 33–40. https://proceedings.neurips.cc/paper/2005/file/e833e042f509c996b1b25324d56659fb-Paper.pdf

[10] S. Basu, A. Banerjee, R.J. Mooney, Active semi-supervision for pairwise constrained clustering, in Proceedings of the 2004 SIAM International Conference on Data Mining, Orlando, Florida, USA, 2004, pp. 333–344.

[11] M. Bilenko, S. Basu, R.J. Mooney, Integrating constraints and metric learning in semi-supervised clustering, in Proceedings of the Twenty-first International Conference on Machine Learning, Banff, Alberta, Canada, 2004, p. 11.

[12] O. Chapelle, A. Zien, Semi-supervised classfication by low density separation, in Proceeding of the Tenth International Workshop on Artificial Intelligence and Statistics, Barbados, 2005, pp. 57–64.

[13] O. Chapelle, M. Chi, A. Zien, Acontinuation method for semisupervised SVMS, in Proceedings of the 23rd International Conference on Machine Learning, Pittsburgh, USA, ACM, 2006, pp. 185–192.

[14] Y. Pei, L.P. Liu, X.Z. Fern, Bayesian active clustering with pairwise constraints, in: A. Appice, P. Rodrigues, V. Santos Costa, C. Soares, J. Gama, A. Jorge (Eds.), Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, Porto, Portugal, 2015, pp. 235–250.

[15] A. Demiriz, K.P. Bennett, M.J. Embrechts, Semi-supervised clustering using genetic algorithms, in Artificial Neural Networks in Engineering, 1999, pp. 809–814. http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=D09947D442823466737C629281F31E1E?doi=10.1.1.62.1542&rep=rep1&type=pdf

[16] S. Basu, A. Banerjee, R. Mooney, Semi-supervised clustering by seeding, in International Conference on Machine Learning, Sydney, Australia, 2002, pp. 27–34. https://www.cs.utexas.edu/~ml/papers/semi-icml-02.pdf

[17] R. Yan, *et al.*, A discriminative learning framework with pairwise constraints for video object classification, IEEE Trans. Pattern Anal. Mach. Intell. 28 (2006), 578–593.
[18] H. Zeng, Y. M. Cheung, Semi-supervised maximum margin clustering with pairwise constraints, IEEE Trans. Knowl. Data Eng. 24 (2012), 926–939.
[19] B. Kulis, *et al.*, Semi-supervised graph clustering: a kernel approach, Mach. Learn. 74 (2009), 1–22.
[20] Z. Lu, Semi-supervised clustering with pairwise constraints: a discriminative approach, in International Conference on Artificial Intelligence and Statistics, San Juan, Puerto Rico, 2007, pp. 299–306. http://proceedings.mlr.press/v2/lu07a/lu07a.pdf
[21] X. Wang, B. Qian, I. Davidson, Labels vs. pairwise constraints: a unified view of label propagation and constrained spectral clustering, in International Conference on Data Mining, Brussels, Belgium, 2012, pp. 1146–1151.
[22] N. Nguyen, R. Caruana, Improving classification with pairwise constraints: a margin-based approach, in: W. Daelemans, B. Goethals, K. Morik (Eds.), Machine Learning and Knowledge Discovery in Data Bases, Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Antwerp, Belgium, Springer, Berlin, Heidelberg, Germany, 2008, pp. 113–124.
[23] A. Rodriguez, A. Laio, Clustering by fast search and find of density peaks, Science. 344 (2014), 1492–1496.
[24] R. Bie, *et al.*, Adaptive fuzzy clustering by fast search and find of density peaks, Pers. Ubiquitous Comput. 20 (2016), 785–793.
[25] W.Q. Fan, *et al.*, Sdenpeak: semi-supervised nonlinear clustering based on density and distance, in IEEE Second International Conference on Big Data Computing Service and Applications, Washington D.C., USA, 2016, pp. 269–275.
[26] M. Parmar, *et al.*, A novel density peak clustering algorithm based on squared residual error, in 2017 International Conference on Security, Pattern Analysis, and Cybernetics, Shenzhen, China, 2017, pp. 43–48.
[27] X. Xu, *et al.*, A feasible density peaks clustering algorithm with a merging strategy, Soft Comput. 23 (2019), 5171–5183.
[28] R.H. Liu, *et al.*, Constraint-based clustering by fast search and find of density peaks, Neurocomputing. 330 (2019), 223–237.
[29] F. Gao, *et al.*, A novel semi-supervised learning method based on fast search and density peaks, Complexity. 2019 (2019), 1–23.
[30] S. Sieranoja, P. Fränti, Fast and general density peaks clustering, Pattern Recognit. Lett. 128 (2019), 551–558.
[31] X. Xu, A robust density peaks clustering algorithm with density-sensitive similarity, Knowl. Based Syst. 200 (2020), 106028.
[32] X.Q. Min, *et al.*, Automatic determination of clustering centers for "clustering by fast search and find of density peaks", Math. Prob. Eng. 2020 (2020), 1–11.
[33] M. Śmieja, *et al.*, A classification-based approach to semi-supervised clustering with pairwise constraints, Neural Netw. 127 (2020), 193–203.
[34] S. Fogel, *et al.*, Clustering-driven deep embedding with pairwise constraints, IEEE Comput. Graph. Appl. 39 (2019), 16–27.
[35] J.P. Mei, Pairwise constrained fuzzy clustering: relation, comparison and parallelization, Int. J. Fuzzy Syst. 21 (2019), 1938–1949.
[36] Y.Z. Ren, *et al.*, Semi-supervised DenPeak clustering with pairwise constraints, in Pacific Rim International Conference on Artificial Intelligence, Nanjing, China, 2018, pp. 837–850.
[37] Z.Y. Chen, *et al.*, An active semi-supervised clustering algorithm based on seeds set and pairwise constraints, J. Jilin Univ. 55 (2017), 664–672.
[38] S.F. Ding, *et al.*, Research of semi-supervised spectral clustering algorithm based on pairwise constraints, Neural Comput. Appl. 24 (2014), 211–219.
[39] H. Li, M. Wang, X.S. Hua, MSRA-MM 2.0: a large-scale web multimedia dataset, in IEEE International Conference on Data Mining Workshops, Miami, FL, USA, 2009, pp. 164–169.
[40] D. Dua, C. Graff, UCI Machine Learning Repository, Irvine, CA: University of California, School of Information and Computer Science, 2019. http://archive.ics.uci.edu/ml
[41] S.S. Rangapuram, M. Hein, Constrained 1-spectral clustering, Comput. Sci. International conference on Artificial Intelligence and Statistics, La Palma, Canary Islands, 22 (2012), 1143–1151. http://proceedings.mlr.press/v22/sundar12/sundar12Supple.pdf
[42] Y. Xiao, J. Yu, Semi-supervised clustering based on affinity propagation algorithm, J. Softw. 19 (2008), 2803–2813.
[43] I.S. Dhillon, S. Mallela, D.S. Modha, Information-theoretic Co-clustering, in Knowledge Discovery and Data Mining, Washington, DC, USA, 2003, pp. 89–98.
[44] S. Garcia, *et al.*, Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: experimental analysis of power, Inf. Sci. 180 (2010), 2044–2064.