applied sciences

MDPI

Article

Ensemble Clustering in GPS Velocities: A Case Study of Turkey

Batuhan Kılıç and Seda Özarpacı *

Department of Geomatic Engineering, Yildiz Technical University, Istanbul 34220, Turkey
* Correspondence: ozarpaci@yildiz.edu.tr

Abstract: Block modeling is an effective way to understand Earth's crustal deformation. However, the choice of block boundaries and the number of blocks affect the model results. Therefore, the subjectivity of this analysis should be avoided. Clustering analysis can be used to define the blocks of GPS (Global Positioning System) velocity fields without a priori information. Unfortunately, clustering methods also have unique solutions and differ with various algorithms. Ensemble methods could be an answer to enhance the clustering results for GPS velocities. In this study, we use ensemble clustering to identify block boundaries before block modeling without a priori information about the data. The ensemble clustering method is used for the first time in the clustering of GPS velocities and the case of Turkey is discussed. The published horizontal GPS velocities were first clustered with five different clustering methods and the optimum classes were determined using ensemble clustering methods. It is proven that the Meta-CLustering Algorithm can be used in terms of ensemble clustering for this region.

Keywords: clustering analysis; GPS velocities; ensemble clustering; Meta-CLustering algorithm

check for updates

Citation: Kılıç, B.; Özarpacı, S. Ensemble Clustering in GPS Velocities: A Case Study of Turkey. *Appl. Sci.* **2022**, *12*, 12636. https://doi.org/10.3390/app122412636

Academic Editors: Małgorzata Charytanowicz and Piotr A. Kowalski

Received: 26 October 2022
Accepted: 3 December 2022
Published: 9 December 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Turkey is in the eastern Mediterranean region in which the Arabian and African plates collide with the Eurasian Plate [1]. This plate collision zone reveals itself in various tectonic processes such as continental strike-slip faults such as the dextral North Anatolian Fault (NAF) or sinistral East Anatolian Fault (EAF) and Dead Sea Fault, oceanic lithosphere subduction and results such as Cyprus and Hellenic Arcs [2], continental extensions such as Sea of Marmara, Aegean region [3], western Turkey, westward extrusion of the Anatolian plate. All these interactions make Turkey an open laboratory and help researchers to understand and interpret crustal deformation.

Many models have been developed to determine the behavior of the earth's crust such as continuum or kinematic models. Among these, block models are closer to surface deformation [4]. Plate kinematic models aim to quantify the slip rates for major faults and rotation rates of crustal blocks [1,5]. The utility of the kinematic model depends on how well the block rotations are determined and the presence of significant intra-block deformation. These depend on the data distribution and can be considered subject-specific. Moreover, the determination of blocks and block boundaries is generally subjective and sometimes questionable [4]. Therefore, the utility of block models depends on how well-defined the block boundaries are.

When starting block modeling for the GPS velocity field, candidate blocks are first determined by the location of major faults [4]. Reilinger et al. [5] used mapped faults, historical earthquakes, and the seismicity of the region to determine the block boundaries. However, the location of the ductile shear zone at depth does not always directly reflect by the surface traces of strike-slip faults and clustering could be used to locate the shear zones at depth in the absence of a priori of the surface fault traces [6,7].

Clustering analysis has been employed in numerous real-world problems such as data mining, pattern recognition, statistics, document retrieval, and machine learning [8,9]. Various clustering algorithms such as *k*-means, *k*-medoids, Hierarchical Agglomerative

Clustering (HAC), Euler-vector, and Gaussian Mixture Model (GMM) have been used in geodetic studies as well to find proper solutions that are acceptable and GPS velocities have been clustered to determine boundaries before block modeling [7,10–17]. However, as it is known, clustering is considered to be an inherently unsupervised learning method and forms normally a bottleneck problem. In other words, there is no a priori information about the underlying data distributions or about any specific properties that we want to find, or about what we consider proper solutions to the data. Each of the different clustering algorithms may produce different results by implicitly or explicitly imposing a particular structure on a similar data set. On the other hand, based on the impossibility theorem [18], there is no single clustering algorithm that can produce consistent and optimal results for different problems and there is no consensus on a universal standard for choosing any clustering algorithm for a specific problem. At this point, there is a need for efficient methods that can benefit from combining the strengths of many individual clustering algorithms [19]. To overcome this problem and to improve the quality of the single clustering algorithm results, the concept of combining different clustering results known as cluster ensemble (consensus clustering) has been introduced [20]. It has been broadly employed in clustering research for the purpose of enhancing the quality and robustness of individual clustering techniques [9,21,22].

Therefore, the objective of this study is to explore the following questions "Is there a better way to cluster GPS velocities? Which one is the proper solution (individual or ensemble clustering) about the cluster/block boundaries?". In the direction of these research questions, we first gathered horizontal velocities inferred from the published cGPS (continuous GPS) measurements in Turkey for clustering GPS velocities. Then, once the determined the number of clusters k that best represents, we clustered GPS velocities using five different clustering techniques including Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH), k-means, mini batch k-means, HAC, and spectral clustering. After that, unlike the other studies, we investigated the performance of three cluster ensemble techniques known as Hybrid Bipartite Graph Formulation (HBGF), Meta-CLustering Algorithm (MCLA), and Non-Negative Matrix Factorization-based (NMF) consensus clustering to find out whether they produce proper results for GPS velocities or not. With this paper, ensemble clustering algorithms are used for the first time in the clustering of GPS-derived horizontal velocities.

2. Data and Methodology

The data used in this paper are the last published horizontal GPS velocities for Turkey in [15] shown in Figure 1. The velocity field is obtained from 10 years of data between 2008 and 2018 of Continuously Operating Reference Stations Turkey (CORS-TR) and some other networks to improve the velocity field [15]. The data consist of CORS-TR data which are homogeneously distributed in Turkey, Turkish National Permanent GNSS Network (TNPGN), and Marmara Region Continuous Network (MAGNET). 19 TNPGN stations operated by the General Directorate of Mapping, and 11 MAGNET stations from the Marmara Region Continuous Network (MAGNET) which is operated by the Scientific and Technical Research Council of Turkey (TUBITAK), Marmara Research Centre, Earth and Marine Sciences Research Institute (EMSRI) are included. The velocities derived from TNPGN and MAGNET GPS data cover the period between 2008 and 2015.

A total of 188 GPS stations' data were used for clustering analysis. Firstly, we determined the optimum number (k) of clusters using the Gap statistics method which is designed to be applied to any distance measure and clustering technique. After determining the k value, we generated clustering models from different clustering techniques (members) for building the ensemble clustering approach. Last but not least, we combined the outputs of the members to obtain the final clustering results using different ensemble clustering methods and determined the quality of the final solutions (Figure 2).

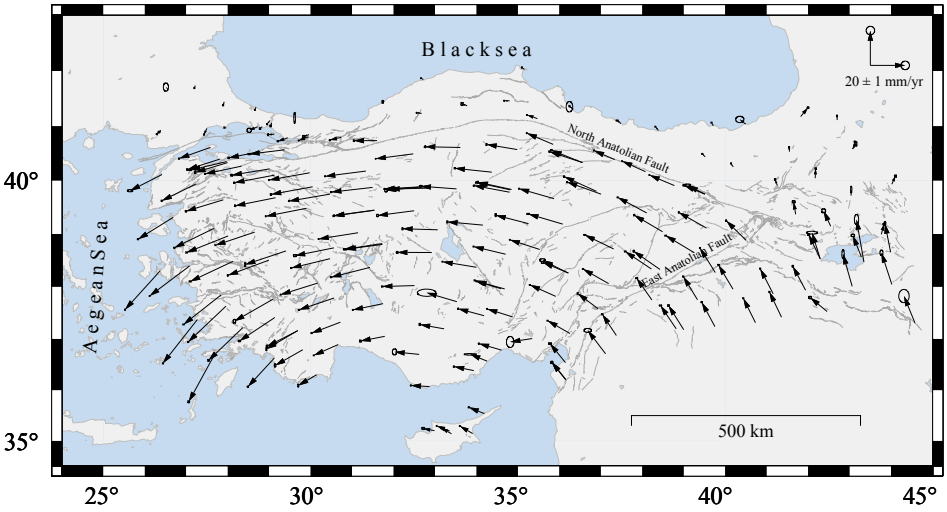

Figure 1. GPS velocity field with respect to Eurasian tectonic plate from [15], gray thin lines show active faults [23].

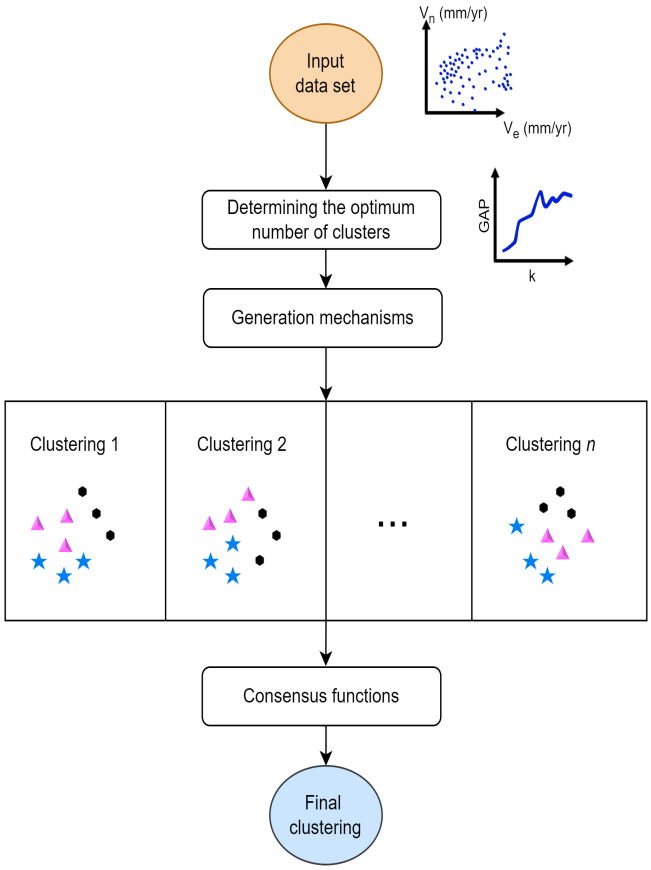

Figure 2. The flowchart of this study.

2.1. Gap Statistic Algorithm

The Gap is one of several methods determining the number of clusters k that best represents a given data set. The Gap algorithm is a data mining statistic developed by Tibshirani et al. [24] to measure the significance of the clustering for each k. This procedure determines the validity or the goodness of fit of the clustering evaluations. Moreover, it is a popular internal indicator such as Silhouette and/or Davies–Bouldin in terms of test data processing of building the clustering algorithm. The advantage among other methods that have different computational models for determining the optimal number of clusters is that it is designed

to be applied to any clustering technique and distance measure. For any $x_{ij} = 1, 2, \ldots, n$ observation data with p features, this method compares the total intra-cluster variation in k different number of clusters with the expected values of the data under a distribution of reference null hypothesis (i.e., no significant clustering). It can be achieved by generating a random distribution of points (Monte Carlo sampling method) that occupy the same spatial extent in velocity space as does the data to be tested [12,24]. A measure of the quality of the rth cluster of given data set is calculated using Equation (1).

$$D_r = \sum_{i,i^{'}=1}^{n_r} d_{ii'} \tag{1}$$

In Equation (1), the sum for both i and $i^{'}$ is overall n_r points within the rth cluster and $d_{(ii')}$ is the squared Euclidean distance in velocity space between the observations i and $i^{'}$ in each class. After that, the sum over all k clusters (W_k) according to the mean of each cluster is defined as:

$$W_k = \sum_{r=1}^{k} (\frac{1}{2n_r} D_r) \tag{2}$$

where n_r is the number of points in cluster r. At this point, for different values of k, the total within-cluster variation between the observation and the reference data is calculated as in Equation (3) by standardizing $log(W_k)$ compared to the expected distribution of the reference data in the zero-mean distribution:

$$GAP_n(k) = E_n^* log(W_k^{ref}) - log(W_k^{obs}) \tag{3}$$

where E_n^* is expressed the expected value of n samples in the reference data distribution. While the superscript "ref" refers to a random, null, and reference data set, the superscript "obs" represents the observed data set being tested. The choice of the optimal number of clusters with the Gap statistic $(GAP_n(k))$ is the value of k at which $log(W_k)$ has the longest decline below the reference value.

2.2. Clustering Ensemble Approach

Generally speaking, a clustering ensemble, also referred to as a clustering aggregation or consensus clustering makes a combination of several clustering results into a single consolidated partition [20]. Theoretically, for a given data set $C = c_1, c_2, \ldots, c_n$ that has n data, X clustering algorithms are utilized to find X single partitions and cluster C (Figure 2). Then, the clustering ensemble member set can be formed with x partitions $\Gamma = P_1, P_2, \ldots, P_x$ and consensus function F, and denoted by $F = P_1, P_2, \ldots, P_x = F(\Gamma)$. Finally, the clustering ensemble process gets a partition P^* of data set C by combining these ensemble members $P_1, P_2, \ldots, P_x$ with Γ without access to original features or algorithms [25,26].

Ensemble clustering approaches promise efficient methods that would significantly benefit from combining the strengths of many individual clustering algorithms [19]. Therefore, by combining single clustering models, ensemble clustering can go beyond what a single clustering algorithm is capable of achieving in several respects:

- *Robustness: better average performance compared to individual clustering algorithms.*
- *Novelty: finding a new consolidated solution unattainable by any single clustering algorithm.*
- *Stability: Final solutions with lower sensitivity to noise and outliers.*
- *Scalability and Parallelization: Ability to compute distributed data and to integrate solutions (parallel clustering) from multiple sources of data or features.* See [9,19,27–29].

The clustering ensemble approach is generally conducted in two principal steps: generation mechanism to obtain the ensemble members and consensus function to combine the ensemble members.

2.3. Generation Mechanisms

Generation is the primary step in clustering ensemble models and its main purpose is to adopt n different clustering models for generating ensemble members. In theory, any individual clustering algorithm can be used in this step, as long as it is appropriate for certain data. Furthermore, the generated members are expected to be as different from each other as possible. Ensuring a high level of diversity among members denotes that they yield distinct information about the data and this can potentially help improve the performance of the ensemble clustering process [25]. In a particular problem, it is important to apply one or more suitable generation processes to accomplish reasonable quality but also diversity.

There are several ensemble member generation approaches: different clustering algorithms, different objects representation, different parameter initialization, projection to subspaces, and different subsets of objects. Among these approaches, different clustering algorithms strategy is used for each member with a consideration that different algorithms may generate more diverse members in this study. Several individual clustering algorithms such as *k*-means, *k*-medoids, Fuzzy *C*-means algorithm, self-organizing maps and hierarchical clustering were applied on different data sets to generate ensemble members in the literature [30–32].

2.3.1. Birch Clustering

BIRCH is a clustering algorithm that can cluster noisy and multidimensional data in particular massive data sets with a single scan [33]. It greatly reduces the temporal costs unlike *k*-means and dynamically and incrementally clusters acquired data instances. For a given data set $i = 1, 2, \ldots, n$ with N-dimensional data points in a cluster $\overrightarrow{X_l}$:

$$\overrightarrow{X0} = \frac{\sum_{t=1}^{N} \overrightarrow{X_l}}{N} \tag{4}$$

$$R = \left(\frac{\sum_{t=1}^{N} (\overrightarrow{X_t} - \overrightarrow{X0})}{N}\right)^{1/2} \tag{5}$$

$$D = \left(\frac{\sum_{t=1}^{N} \sum_{j=1}^{N} (\overrightarrow{X_t} - \overrightarrow{X_j})^2}{N(N-1)}\right)^{1/2} \tag{6}$$

where $\overrightarrow{X0}$ indicates centroid of each cluster. R refers to the average distance of member points from the centroid. D represents the diameter of the cluster. R and D are two alternative measures of the tightness of the cluster around the centroid [34]. After that, five alternative distances are defined to measure their proximity between two clusters. Given the centroids of two clusters $\overrightarrow{X0_1}$ and $\overrightarrow{X0_2}$, the Euclidean distance DO, Manhattan distance $D1$, the average inter-cluster distance $D2$, average intra-cluster distance $D3$ and variance increase distance $D4$ of the two clusters are calculated as follows:

$$D0 = \left((\overrightarrow{X0_1} - \overrightarrow{X0_2})^2\right)^{1/2} \tag{7}$$

$$D1 = \left|\overrightarrow{X0_1} - \overrightarrow{X0_2}\right| = \sum_{t=1}^{d} \left|\overrightarrow{X0_1}^{(t)} - \overrightarrow{X0_2}^{(t)}\right| \tag{8}$$

$$D2 = \left(\frac{\sum_{t=1}^{N_1} \sum_{j=N_1+1}^{N_1+N_2} (\overrightarrow{X_t} - \overrightarrow{X_j})^2}{N_1 N_2}\right)^{1/2} \tag{9}$$

$$D3 = \left(\frac{\sum_{t=1}^{N_1+N_2} \sum_{j=1}^{N_1+N_2} (\overrightarrow{X_t} - \overrightarrow{X_j})^2}{(N_1+N_2)(N_1+N_2-1)}\right)^{1/2} \tag{10}$$

$$D4 = \sum_{k=1}^{N1+N2}(\overrightarrow{X_k} - \frac{\sum_{l=1}^{N_1+N_2}\overrightarrow{X_l}}{N_1+N_2})^2 - \sum_{i=1}^{N_1}(\overrightarrow{X_i} - \frac{\sum_{l=1}^{N_1}\overrightarrow{X_l}}{N_1})^2 - \sum_{j=N_1+1}^{N1+N2}(\overrightarrow{X_j} - \frac{\sum_{l=N_1+1}^{N_1+N_2}\overrightarrow{X_l}}{N_2})^2 \quad (11)$$

The BIRCH method is based on two important concepts: the Clustering Feature (CF) and the CF tree. The CF is a structure consisting of three parameters that will represent in the main memory the subsets formed from small groups of data objects. The clustering vector of the cluster $\overrightarrow{X0_1}$ is expressed as $CF = (N, \overrightarrow{LS}, SS)$, where N is the number of data, $\overrightarrow{LS}$ is the linear sum of the N data, i.e., $\sum_{i=1}^{N}\overrightarrow{X_l}$, and SS is the square sum of the N data, i.e., $\sum_{i=1}^{N}(\overrightarrow{X_l})^2$.

The CF tree created using the CF values determined for the clusters has a structure representing a branching factor B and a threshold value T. The CF tree represents each node as a subset and each entry in the CF tree gives an input value to the sub-nodes which is the sum of this tree. The threshold value T sets the limit of the maximum input value at each node. The first step in the creation of the CF tree is to iteratively determine the appropriate leaf structure according to a chosen distance metric: DO, $D1$, $D2$, $D3$, and $D4$. If the current input values in the data set are greater than the threshold value as a result of the distance calculation process, a new sub-cluster is created [33].

2.3.2. K-Means Clustering

The k-means method is a widely used clustering and unsupervised learning algorithm. In this method, first used by MacQueen [35], the main objective is to partition k clusters from a database consisting of n data, in which each object belongs to the cluster with the nearest mean. By this means, the similarities within the cluster are homogeneous (maximized), and the similarities between clusters are heterogeneous (minimized) after the clustering process. In other words, the within-cluster variances are minimized in k-means clustering (Equation (12)).

$$J_{k_{means}} = \sum_{j=1}^{k}\sum_{i=1}^{n}\left\|x_i^{(j)} - c_j\right\|^2 \quad (12)$$

where k is the number of clusters; n is the number of cases; x is the ith case c is the centroid for cluster j.

The process steps of the k-means clustering are as follows:

- *k points are chosen randomly as initial cluster centroids.*
- *The squared Euclidean distances from each point to the initially chosen centroids are calculated and then points are allocated to the nearest centroid.*
- *The new cluster centroids of the formed clusters are updated by taking the mean of the points in each cluster.*
- *The previous steps are repeated until the changes of all clusters remain stable and reach convergence. Notice that although it is possible to carry out k-means with other distance metrics such as Manhattan, Chebychev, etc., it is not suggested for they may prevent the convergence.*

2.3.3. Mini Batch K-Means Clustering

The mini batch k-means is first presented by Sculley [36] and is proposed as an alternative to the k-means algorithm for especially clustering large data sets. As in partitioning clustering methods, it is used to group data with similar characteristics in a given data set. The main idea underlying the mini batch k-means method is to make a large data set storable in memory by using small random batches of data of fixed size. A new random sample from the data set is retrieved in each iteration and then is used to update the clusters. Finally, this process is repeated until convergence. Unlike k-means clustering, this one does not need all the data sets in the main memory, and therefore greatly reduces the temporal and memory costs of the algorithm [37].

For a given data set $D = x_1, x_2, \ldots, x_y$, $x_i \in \mathbb{R}^m$, x_i refers to a network record. m represents the number of records found in the data set D. The optimization clustering problem is to find the set C of cluster centers $c \in \mathbb{R}^m$ so as to minimize over a data set D of examples $c \in \mathbb{R}^m$ the following function:

$$min \sum_{x \epsilon D} \|f(C, x) - x\|^2 \quad (13)$$

where $f(C, x)$ returns the nearest cluster centroid $c \in C$ to record x using the Euclidean distance. With $|C| = k$, k is the number of clusters that want to be found [36].

2.3.4. Hierarchical Agglomerative Clustering

HAC methods aim to combine the objects by taking into account the similarities (velocity vector in this study) in a data set at certain levels (distance measures). Differing from the partitioning clustering methods such as k-means, there is no need to determine the number of clusters beforehand in the HAC method [15,38].

With HAC, each object or each velocity observation is initially considered as a set. In short, if there are n velocity vectors in the data set, there are initially n clusters at the first step. Then the two closest clusters (with the smallest d_{ij} value) are merged and assembled in a new cluster. So, now $n - 1$ clusters are obtained and iterated distances matrix is calculated. The process is repeated $n - 1$ time and finally, the last two clusters are merged to form a single cluster containing all vectors. All these operations are based on grouping objects into tree structures known as a dendrogram. It is used to explore the relationship between each object in a "bottom-up continuation" of grouping results. In the dendrogram tree or chart, the distances between merged clusters are shown (Figure 3).

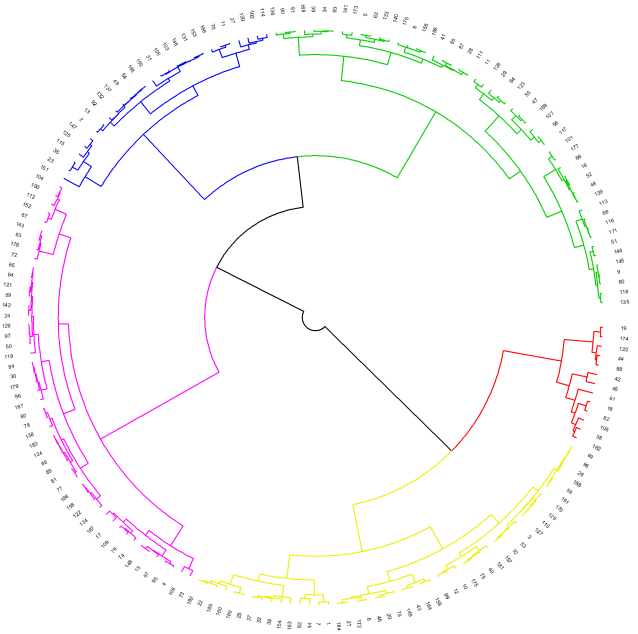

Figure 3. The dendrogram of the hierarchical agglomerative clustering in the form of a flower chart.

At this point, the higher the distance between the merged clusters, the clearer the distinction between them [15]. By visual inspection of the dendrogram, it is possible to decide where to cut the cluster tree which results in natural groupings and/or to determine the optimum number of clusters. In Figure 3, the flower chart dendrogram of HAC clustering of GPS velocities (from k = 2 to 5) is constructed using squared-Euclidean distance and ward linkage.

At the hierarchical clustering procedure, distance calculations and (dis)similarity between objects in the data set are updated at each iteration. The matrix of calculated

distance/(dis)similarity values is the basis for using the selected linkage criterion. To decide which objects/clusters should be merged, methods to measure the similarity between them are needed. Several distance metrics such as squared Euclidean, Manhattan, Cosine and Minkowski, etc., are used depending on different data sets and the purpose of the study. Note that the squared Euclidean and Manhattan distances consider both the orientation and magnitude of the velocity vectors in clustering GPS velocities, unlike, e.g., cosine, which treats all velocities as unit vectors. On the other hand, a linkage criterion is used to define the proximities of pairs of objects into clusters based on their similarity. In particular, various linkage criteria are used for this purpose: (1) Single or minimum (nearest neighbor); (2) Average or mean (unweighted pair group method with arithmetic mean); (3) Complete (farthest neighbor); (4) Weighted (weighted pair group method with arithmetic mean); (5) Centroid; (6) Median; and (7) Ward's minimum variance method [39]. For instance, Simpson et al. [10] and Takahashi et al. [16] used the centroid linkage approach to join GPS velocity clusters. It is important to note that the last three of these criteria as mentioned above require Euclidean distances for geometric precision in the calculation of the centers and it is not appropriate to use the Manhattan distance in combination with these.

2.3.5. Spectral Clustering

The Spectral algorithm is a modern clustering technique based on graph theory that constructs a graph by addressing each object as a node in given data set and weights the edges with pairwise similarities between them. This method is a powerful unsupervised learning algorithm based on the eigenvalue decomposition of this graph. It is very simple to implement by using standard linear algebra concepts and bioinformatics. In addition, this algorithm has a nonparametric model and the ability to capture a wider range of geometries, which makes it more flexible than other algorithms such as *k*-means. Despite their superiority, it is not widely used over classical algorithms such as hierarchical clustering and *k*-means due to their computational complexity for large-scale data sets [40].

The spectral clustering process is conducted in three stages: constructing the (Laplacian) graph based on the affinity matrix of the data set, spectral representation, and clustering. For calculating the affinity matrix, several approaches such as ϵ-neighborhood graph, κ-nearest neighbor graph, and fully connected graph are utilized [41]. A graph "G" commonly includes a set of V of vertices (nodes) together with a set E of edges (lines) and is shown $G = (V, E)$. Given a group of n data objects $X = x_1, x_2, \ldots, x_n$ in $\mathbb{R}^l$ that we want to cluster them into c clusters:

1. Compute affinity matrix $A \in \mathbb{R}^{nxn}$ defined using $A = exp(-\|x_i - x_j\|^2/(2\sigma^2))$ if $i \neq j$ and $A_{ij} = 0$, where σ is scale parameter and the choosing value of σ is performed manually.
2. Construct degree matrix (D) to be the diagonal matrix.
3. Calculate the normalized Laplacian matrix [42] defined using $L = D^{-1/2} * A * D^{-1/2}$.
4. Find $e_1, e_2, \ldots, e_c$, the k largest eigenvectors of Laplacian matrix (L), and construct the matrix $U = [x_1, x_2, \ldots, x_c] \in \mathbb{R}^{nc}$ by stacking the eigenvectors in columns.
5. Form the matrix Y from U by renormalizing each of U's rows to have unit length $(Y_{ij} = U_{ij}/(\sum_j U_{ij}^2)^{1/2})$.
6. Let each row of Y be points in $\mathbb{R}^C$, cluster them into c clusters using *k*-means or any other algorithm.

Finally, assign the original point x_i to cluster c if and only if the corresponding row i of the matrix Y was allocated to cluster c [43].

2.4. Consensus Functions

The consensus functions are the main step in any clustering ensemble algorithm and aggregate the outcome of ensemble members to produce the final data partition or consensus partition P^*. Several existing consensus functions have been used to generate final clustering in the literature and they are gathered into two main approaches according

to the review study conducted by Vega-Pons and Ruiz-Shulcloper [8]: object co-occurrence and median partition.

The object co-occurrence approach: It first calculates the co-occurrence of objects by identifying which must be the cluster label associated with each object in the consensus partition. To accomplish this, it counts how many times one or two objects belong to the same cluster, and computes the final clustering result through a voting process among the objects. This approach includes methods such as the Relabeling and Voting method [44,45], the Co-association matrix [46], and the Graph- and Hypergraph-based method [20,47].

Median partition: It produces solutions by treating the consensus function as an optimization problem of finding the median partition regarding the cluster ensemble [25]. Regarding Vega-Pons and Ruiz-Shulcloper [8], it is described as "the partition that maximizes the similarity with all partitions in the clustering ensemble". This approach is categorized into three groups: Non-Negative Matrix Factorization based method [48,49], Kernel-based method [50], and Genetic-based algorithms [51].

Vega-Pons and Ruiz-Shulcloper [8] explicitly state in their work that consensus functions based on the median partitioning approach are theoretically more proposed than those based on the object co-occurrence approach. Notwithstanding, Topchy et al. [52] presents theoretical arguments based on the validity of both approaches. Therefore, we have investigated two methods based on the object co-occurrence approach (Hybrid Bipartite Graph Formulation and Meta-CLustering Algorithm) and one method based on the median partitioning approach (Non-Negative Matrix Factorization consensus clustering) for clustering GPS velocities.

2.4.1. Hybrid Bipartite Graph Formulation

HBGF is a kind of clustering ensemble model which transforms the combination problem into a hypergraph partitioning problem. It was first introduced by Fern and Brodley [47]. This method models the clusters and the data points in the same graph simultaneously [53]. This method creates a bipartite graph with no edges between vertexes that are both instances or clusters. There is an edge between two nodes only if one node represents a cluster and the other node corresponds to an object belonging to that cluster. Then, traditional graph partition techniques such as the METIS algorithm [54] or spectral clustering [43] are applied to obtain this consensus partition.

2.4.2. Meta-Clustering Algorithm

MCLA defines the similarity between two clusters in terms of the number of objects grouped in them using the extended Jaccard index [55]. After that, the similarity matrix between the clusters is created and then a graph is formed by considering clusters as nodes and assigning a weight equal to the similarity between clusters to the edge between two nodes. Finally, this graph is partitioned using the METIS algorithm and the final partition known as "meta-clustering" is obtained.

2.4.3. Non-Negative Matrix Factorization

NMF is the clustering ensemble method based on a non-negative matrix factorization process. It was first introduced by Paatero and Tapper [56] and popularized in an article by Lee and Seung [57]. NMF corresponds to the problem of factorizing a given non-negative data matrix V into two matrix factors, i.e., $V \approx WH$, while requiring W and H to be non-negative. At this point, while W refers to the dimensionally reduced data matrix, H represents the associated coefficient matrix (weights associated with W). In theory, although various extensions have emerged to propose the clustering feature, the original model formulation does not include a clustering objective and was first presented as a dimensionality reduction algorithm.

In this method, the distance between partitions is defined using Equation (14):

$$\mu(P, P^{'}) = \sum_{i,j=1}^{n} \mu_{ij}(P, P^{'}) \tag{14}$$

where $\mu(P, P^{'}) = 1$ if c_i and c_j belong to different clusters in the other and belong to the same cluster in one partition, otherwise $\mu(P, P^{'}) = 0$. The consensus partition (final) is determined by the median partition model using μ [8].

3. Results and Discussion

3.1. How Many Clusters?

Conceptually speaking, determining the optimal number of clusters is usually performed subjectively. Depending on the distribution pattern, structure, and scale of the parameters in the data set, however, the choice of the best number of clusters poses a bottleneck problem. To determine the goodness of fit or validity of the clustering results, the gap statistic algorithm is used (Figure 4).

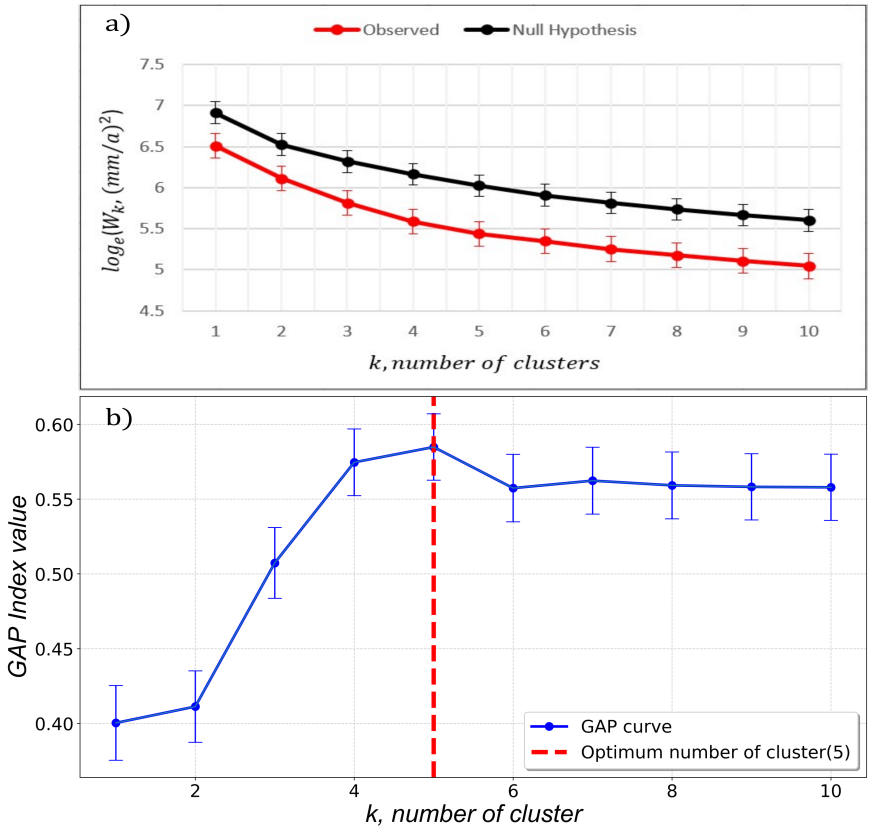

Figure 4. The gap statistic algorithm: (**a**) Plot of $log_e(W_k)$ as a function of k, the number of clusters; (**b**) the difference between the two curves in (**a**).

A plot of $log_e(W_k)$ versus the number of clusters k for this data is shown in Figure 4a. The null hypothesis in Figure 4a elementally represents the average of 20 different random reference data sets, and the error bars (two standard deviations on either side of the plotted point) correspond to the scatter of those individual reference sets about that mean value. The $log(W_k^{obs})$ values in Figure 4a fall significantly (more than two standard deviations) below the null hypothesis, indicating that clustering is significant for $k \geq 1$. Figure 4b is a plot of the gap statistic (the difference between the two curves in Figure 4a) for this data; the error bars in Figure 4b refer to two standard deviations on either side of the plotted point. The gap statistic increases as the number of clusters k is increased from k = 1 to 5,

but it is seen that there is no significant increase beyond $k = 5$. Therefore, it can be revealed that $k = 5$ is the optimum choice of k. This value is similar to the other internal indicator results regarding the determination of the optimum number of clusters [7,15]. For instance, ref. [15] the clustered velocity field of Turkey with the optimum number of clusters that best represent the data, 5 as they found with the Silhouette Index. Moreover, in [7], the authors used various optimum cluster number determination algorithms such as Elbow, Davies–Bouldin, GAP, and Silhouette to detect the best *fk* value that represents the data and they found the value as 5 for this region in a combined data set.

3.2. Identification of Blocks

Continental lithosphere deformation especially in tectonic regions such as Turkey can be provided by GPS data [10]. Elastic blocks that are bounded by active faults can describe velocities derived from GPS data. In geographic space, remaining cluster boundaries with increasing cluster numbers are identified as block boundaries, especially if the boundary of the cluster coincides with an active fault [13]. Ref. [7] gave the results from $k = 2$ to 10 and North and East Anatolian transform faults can be seen as the border of clusters because these two are the most prominent faults. Because Anatolia is divided into more clusters with the increasing number, one can note that there is not a block boundary inside this block. Here, we tried to assess the velocity field for $k = 5$, because over five is not tectonically significant. With five clusters in this region, also Aegean part identifies a cluster and it is compatible with [5]. On the other hand, the eastern part differs from [5] block boundaries because clustering methods determine clusters according to velocity discontinuities. The velocity gradient in the eastern part differs from the fault traces, consequently, cluster borders are not incompatible with [5] in this region.

3.3. Clustering Results

Clustering analysis results vary according to the number of clusters which is a challenging task to determine. Some algorithms have the advantage of running without identification of the k value such as HAC. Internal evaluation criteria are a powerful tool to assess the clusters such as Davies–Bouldin, Silhouette, or GAP as used in this paper. Unfortunately, these indexes take the similarities of the objects into account, not the tectonics or the velocity gradients. Moreover, algorithms such as k-means and/or mini batch k-means can not detect non-convex clusters. On the other hand, HAC, BIRCH, and spectral clustering are suitable for data set with arbitrary shape but they suffer from high time complexity [58]. As a solution, one can use the k-means algorithm and then apply HAC to take better results [15]. In our paper, the coincides borders with NAF and EAF, and the compatible Aegean cluster with the literature show that GAP gives a fine solution.

BIRCH, HAC, k-means, mini batch k-means, and spectral clustering algorithms were used to cluster GPS-derived velocity field into 5 different classes. The results are illustrated in Figure 5a–e, respectively. All clustering results identify NAF and EAF (seen in Figure 1) as borders of different clusters. In all figures (Figure 5a–e), red stars and blue diamonds show the Eurasian and the Arabian tectonic plates, respectively. The Aegean coast cluster, which is illustrated as pink squares, is changing depending on the clustering method. Moreover, the Anatolian part is divided in two but with different borders according to all methods. In the tectonic sense, Anatolia is accepted as a rigid body with no clusters inside as mentioned previous section. This situation is clearly explained in [7] which has been recently published. The reader is referred to [7] for a background on the clustering analysis of Turkey from published GPS velocities.

As seen in Figure 5, clustering borders can vary and clusters of some GPS sites can change with different algorithms. Let us consider the two sites in Figure 5a (the sites in the navy-blue circle) and the one in Figure 5b (the site in the navy-blue triangle). As compared with Figure 1, these sites have an anomaly compared to the velocity field; therefore, this is natural to be in another cluster. However, site number 2 is in the right cluster in the HAC and mini batch k-means methods. Furthermore, in Figure 5b the marked site is different

only in the HAC method. Moreover, in the spectral clustering method, the four sites in Cyprus are in one of the Anatolia clusters (Figure 5e), while in the other methods they are all in the Eurasian cluster. These subjects will be discussed later in the comparison of clustering results with the ensemble clustering part.

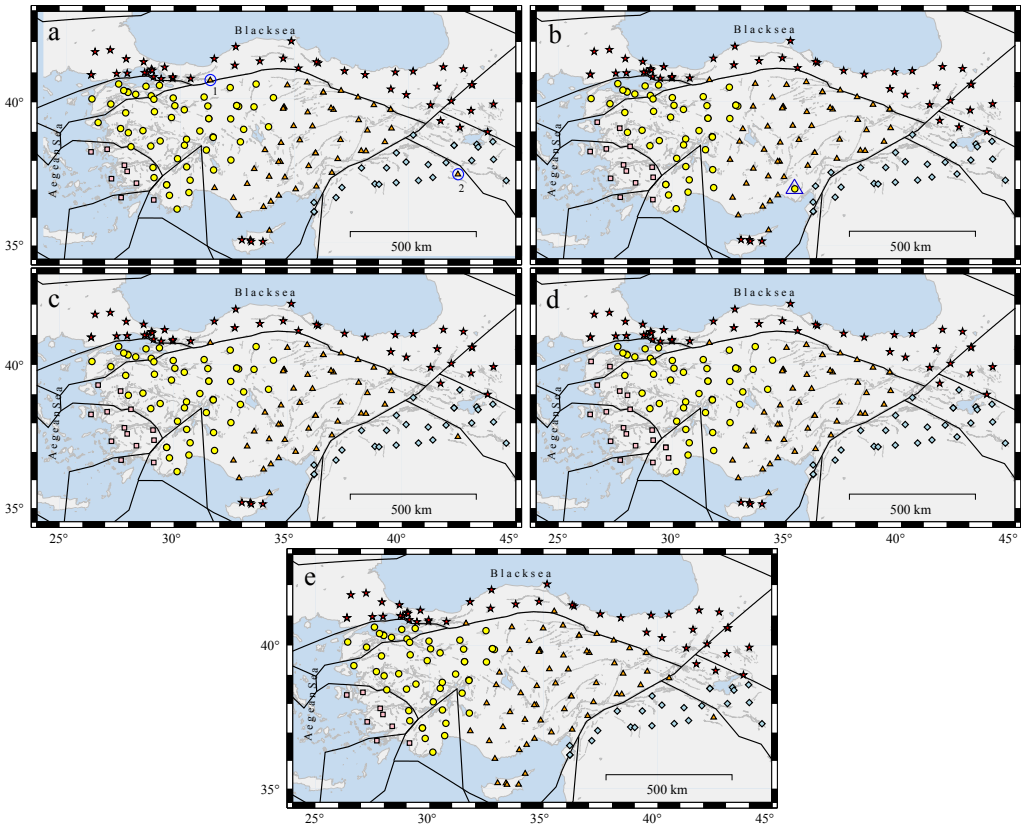

Figure 5. Turkey GPS-derived velocity clustering results for k = 5 clusters with: (**a**) BIRCH; (**b**) HAC; (**c**) k-means; (**d**) mini batch k-means; (**e**) spectral clustering algorithms. Black solid lines are the block model at [5]. Gray thin lines show active faults [23].

3.4. Ensemble Results

After clustering horizontal GPS velocities with the methods mentioned above, it is aimed to determine the most accurate clusters by ensemble clustering algorithms. Figure 6a–c gives the ensemble clustering results from HGBF, MCLA, and NMF, respectively. As seen in Figure 6a,c, HGBF and NMF results are not satisfactory. In both methods, the Eurasian plate does not consist of a group of GPS sites in one shape and color, but two mixed clusters. Furthermore, in the NMF method, there is no Aegean cluster, even though it exists in all clustering method results. Moreover, in the HGBF method, the Aegean cluster is mixed with Anatolian clusters. In Figure 7, one can see the clusters for HGBF, MCLA, and NMF in velocity space, respectively. This figure provides a more discriminating assessment of ensemble results. In Figure 7b, the clusters are in groups and not mixed, but in Figure 7a,c, the situation is the opposite, the objects are involved in different clusters, not split correctly. For example, HGBF performs a clustering process such that each cluster has approximately equal numbers of data. This is because it is a more suitable algorithm for balanced clusters [9]. In balanced clustering, the HGBF results are not consistent because it performs a special clustering process where cluster sizes are constrained. Moreover, in the NMF method, the success of the results depends on the diversity and distribution of the compounds in the data set. Since it neglects local relationships between data points, it fails to reveal the geometric structure of data distributions, which is one of its main drawbacks [59]. On the contrary, MCLA considers a different approach even though it has

a similar structure (graph-based) to HGBF. MCLA first creates meta-clusters for expected clusters and places data points into these meta-clusters through a voting process.

The results show that HGBF and NMF ensemble algorithms are not suitable for GPS velocity field clustering voting for Turkey. However, the MCLA method (Figure 6b) appears to be usable for the GPS velocity field for this region.

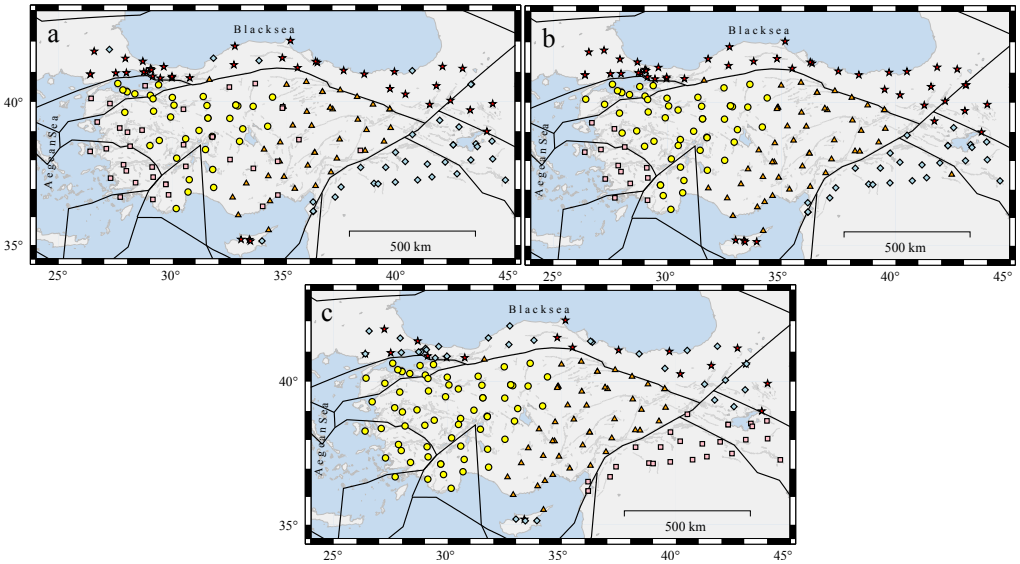

Figure 6. Ensemble results gathered from five clustering algorithms for horizontal GPS-derived velocity field with: (**a**) HGBF; (**b**) MCLA; (**c**) NMF methods. Black solid lines are the block model at [5]. Gray thin lines show active faults [23].

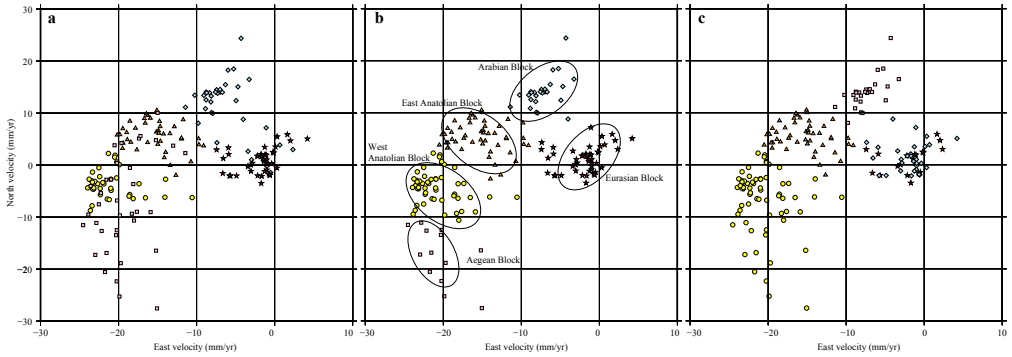

Figure 7. Ensemble results gathered from five clustering algorithms for horizontal GPS-derived velocity field in velocity space: (**a**) HGBF; (**b**) MCLA; (**c**) NMF methods. The colors and the shapes are the same as the map view of the locations in the previous figure.

As seen in Figures 6b and 7b, MCLA results are coherent with five clustering methods, and also with block borders given in [5]. Details will be discussed in the comparison of clustering results with the ensemble clustering part.

3.5. Comparison of Clustering Results with Ensemble Clustering

The MCLA ensemble clustering method has yielded quite consistent results with clustering results and one can say that it improves the results by removing the subjectivity of the clustering algorithms.

Figure 8 shows individual clustering and MCLA ensemble results for eastern Turkey, with the block model at [5], as black solid lines. In the tectonic sense, this is where Arabian and Eurasian plates converge and cause the extrusion of Anatolia towards the west [5]. Therefore, it is quite natural to see three different clusters at this point. Clustering algorithms are quite successful to divide GPS sites into clusters without prior information.

However, clusters of some sites vary with algorithms. One can see in Figure 8, MLZ1, BING, and SIRN GPS sites are in different clusters due to varied methods. As described above (Figure 5a—Number 2), the SIRN site probably has a local effect and is assigned to a cluster that is not even close. However, BING and MLZ1 sites are varied between neighbor clusters. The MCLA ensemble method (Figure 8f) could not do much about the SIRN site (because of the local effects in this site—See Figure 1) but assigned the other sites to the Arabian plate. Figure 8f gives the result of clustering the eastern part and the green dashed line shows the border of the Eurasian and Arabian plates according to five clusters of Turkey.

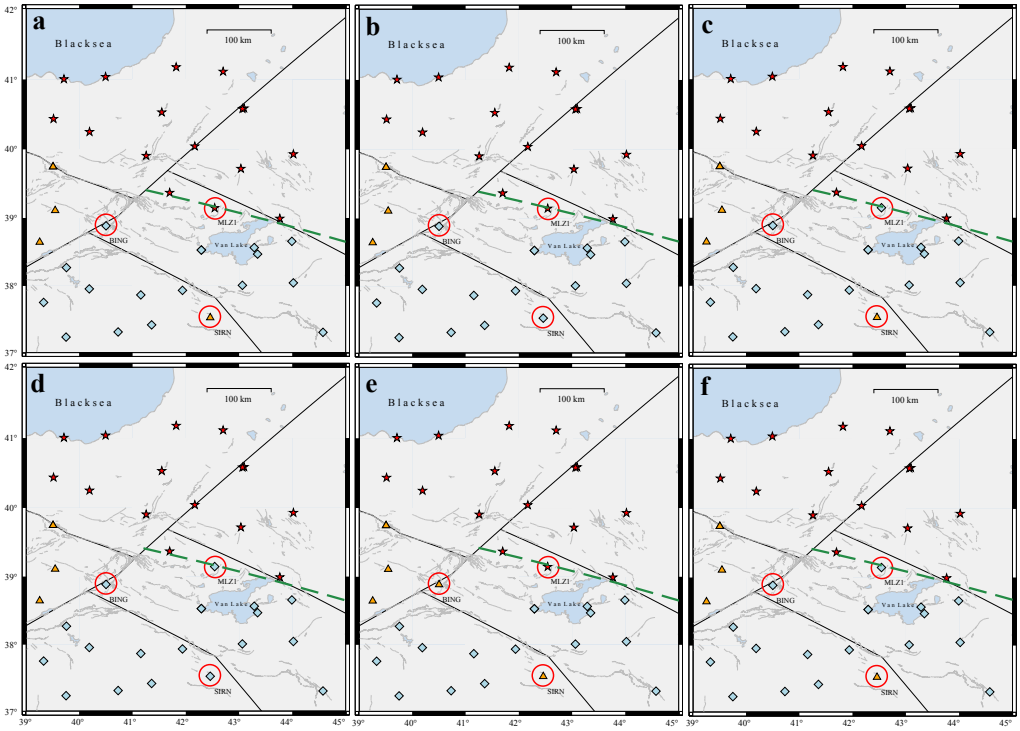

Figure 8. Comparison of clustering results with ensemble clustering for eastern Turkey with: (**a**) BIRCH; (**b**) HAC; (**c**) *k*-means; (**d**) mini batch *k*-means; (**e**) spectral clustering; (**f**) MCLA results. Red circles show the differences between algorithms. Black solid lines are the block model at [5]. The green dashed line is our block border according to the ensemble clustering result. Gray thin lines show active faults [23].

The boundaries of the MCLA clustering are shown with green dashed lines. Comparing our block border with [5], here we only have one border between Eurasia and Arabian plates. Our border is very close to their Caucasus block border but here we do not have that many blocks. In the eastern part while they have five blocks we only have three clusters. The reader should also remember that in [5] the area of the paper is quite large, the Africa-Arabia-Eurasia continental collision zone because they have worked on continental deformation.

As one can see in Figure 9, the western part of Turkey has a complex dynamic mechanism for extension in the Aegean because of the subduction and slab retreat along the Hellenic trench [5]. One can see the difference between clustering algorithms for the Aegean part GPS sites in Figure 9a–e. From north to south, AYVL, KIKA, SALH, DENI, and TVAS GPS stations are in two different clusters, the Aegean coast cluster and the Anatolian cluster, due to the varied algorithms. One can see in Figure 9d, even CAVD, CAV1, and ELMI GPS sites are in the Aegean cluster as a result of the mini batch *k*-means algorithm. Figure 9f shows the last decision for the western part of the Turkey cluster analysis. All these stations except CAVD, CAV1, and ELMI, decided to be in the Aegean cluster as a result of the MCLA algorithm (Figure 9f). Compared to the [5] the Aegean block, the clustering border,

as shown in the green dashed line in Figure 9, is a bit larger than their block border, but generally is compatible.

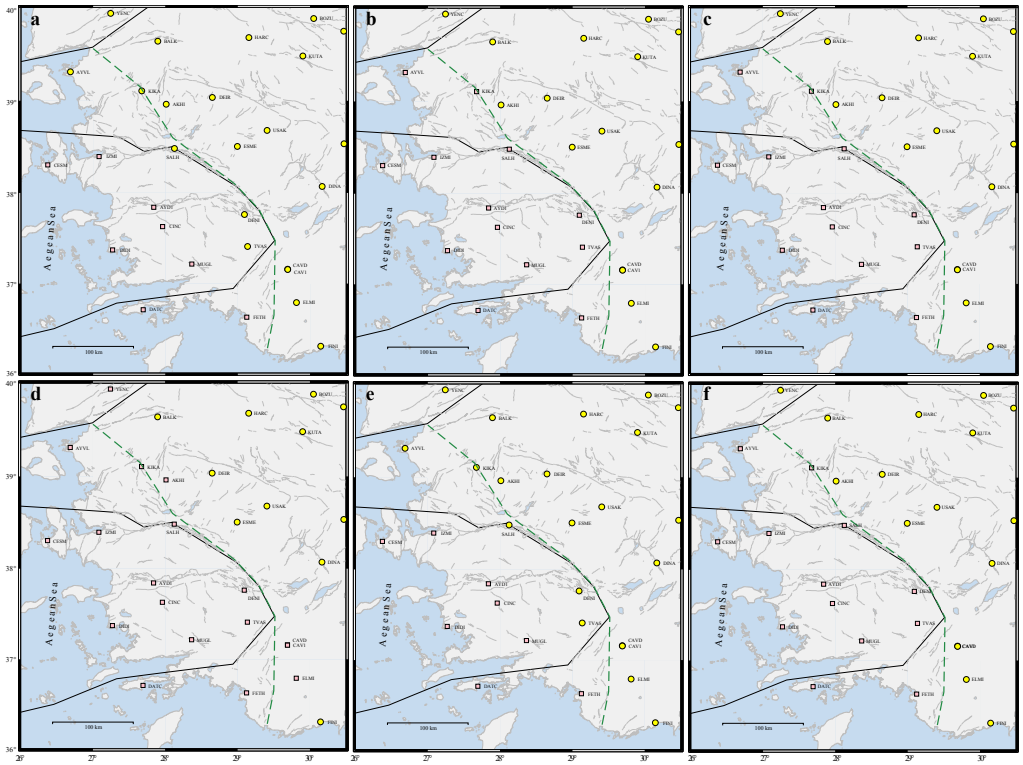

Figure 9. Comparison of individual clustering results with ensemble clustering for western Turkey with: (**a**) BIRCH; (**b**) HAC; (**c**) *k*-means; (**d**) mini batch *k*-means; (**e**) spectral clustering; (**f**) MCLA results. Black solid lines are the block model at [5]. The green dashed line is our block border according to the ensemble clustering result. Gray thin lines show active faults [23].

Figure 10 shows the southern part of Turkey where the African oceanic lithosphere subducts along the Cyprus trench [5]. In Cyprus island, we have five GPS sites and four of them are clustered in the Eurasian cluster except in the spectral clustering algorithm. One can think that Cyprus is another block that has similar GPS-derived velocities to Eurasia [7]. However, spectral clustering assigned these four sites in Anatolia apart from the other algorithms. Perhaps, a solution with Euler vector clustering [13] could enlighten this situation. One can see the result of the MCLA ensemble method that all four GPS sites were assigned in the Eurasian tectonic plate cluster.

The other GPS sites that are assigned in different clusters (Figure 10) are generally exchanged in neighbor clusters except for the ADAN site as explained before. Compared to the block borders with [5], in our study, there is no small block under the Aegean block seen in Figure 10. A small part of this block is in our Aegean coast block and the rest is in Anatolia.

In this study, ensemble clustering methods are tested and the results are analyzed. In the case of Turkey's GPS horizontal velocity field, it is seen that the MCLA method is compatible with clustering methods and eliminates the subjectivity of clustering methods. In addition, when compared with the literature, one can see that the results obtained can be used in block modeling without a priori information about the data. One should keep in mind that these results are specialized in the case of Turkey and should be supported with different examples.

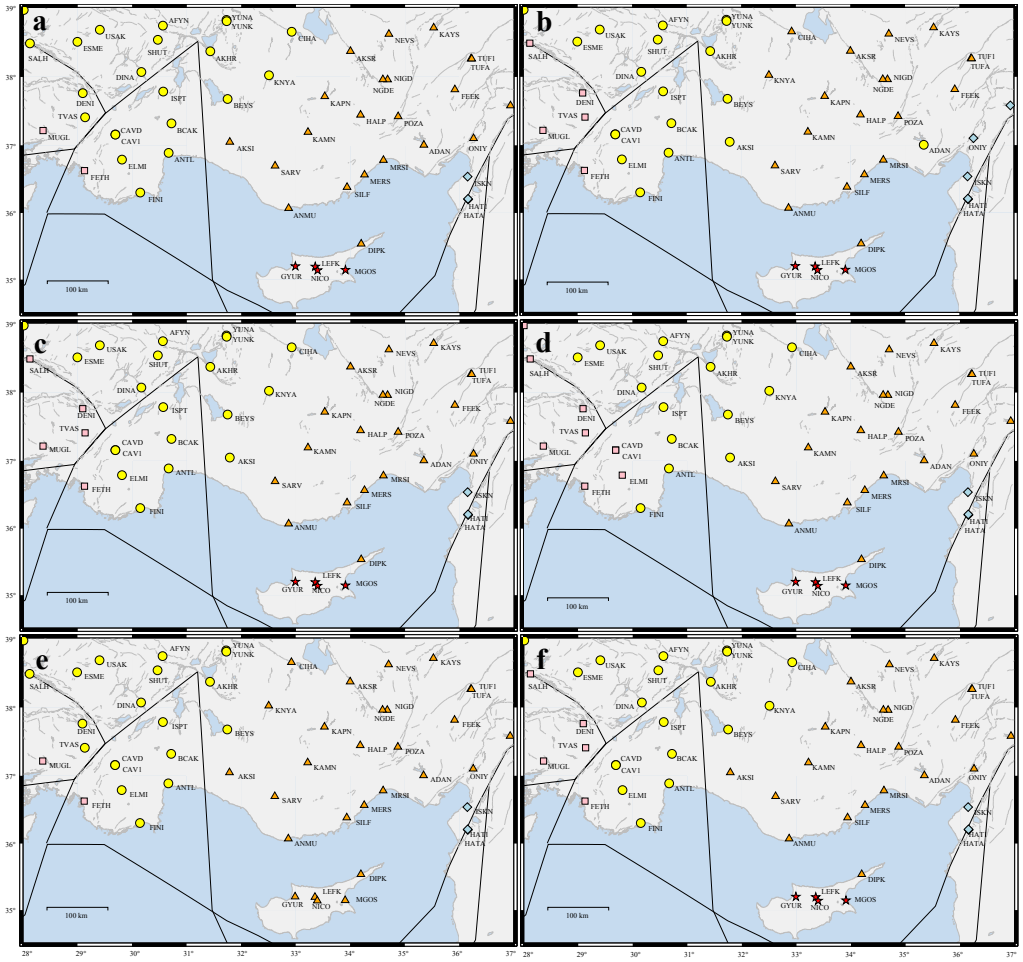

Figure 10. Comparison of individual clustering results with ensemble clustering for western Turkey with: (**a**) BIRCH; (**b**) HAC; (**c**) *k*-means; (**d**) mini batch *k*-means; (**e**) spectral clustering; (**f**) MCLA results. Black solid lines are the block model at [5]. Gray thin lines show active faults [23].

4. Conclusions

In this study, the authors tried to answer the following questions "Is there a better way to cluster GPS velocities? Which one is the proper solution (individual or ensemble clustering) about the cluster/block boundaries?". In the direction of these research questions the results are given as follows:

- Before clustering, the GAP algorithm was used to obtain a priori information about the distribution of GPS velocities and classify the data into five classes.
- Five different individual clustering methods, BIRCH, *k*-means, mini batch *k*-means, HAC, and spectral clustering, were used to classify the published horizontal GPS velocities into five clusters. In general, the individual clustering methods separated NAF and EAF immediately. Furthermore, some sites in the western part of Turkey were assigned for the Aegean block. However, the number of GPS sites and the area of this block are changing due to the used methods. Moreover, in complex regions such as eastern or southern parts of Turkey, GPS sites were assigned to distinct generally neighbor clusters.
- To enhance the differences in clustering methods, the performance of three different ensemble clustering methods, HBGF, MCLA, and NMF-based consensus clustering is utilized for the first time with a GPS velocity field.
- Among the three ensemble clustering methods, HBGF and NMF did not give satisfactory results.

- On the other hand, MCLA consensus results are successful and proven here to be used with GPS-derived velocities.
- As a result, the block boundaries created with the MCLA ensemble clustering algorithm are compatible with the literature.

Author Contributions: Conceptualization, S.Ö. and B.K.; methodology, B.K.; software, B.K.; investigation, S.Ö.; resources, S.Ö.; Writing—Original draft preparation, S.Ö. and B.K.; Writing—Review and editing, S.Ö. and B.K.; visualization, S.Ö. and B.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data can be found in the appendix of the paper at https://link.springer.com/article/10.1007/s00190-019-01235-z (accessed on 15 February 2019). as site name, geographic coordinates, data span, horizontal velocities with standard deviations and their correlations.

Acknowledgments: We used GMT 6 for the map figures [60].

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

BIRCH	Balanced Iterative Reducing and Clustering using Hierarchies
CORS-TR	Continuously Operating Reference Stations Turkey
EAF	East Anatolian Fault
EMSRI	Earth and Marine Sciences Research Institute
GPS	Global Positioning System
GMM	Gaussian Mixture Model
HAC	Hierarchical Agglomerative Clustering
HBGF	Hybrid Bipartite Graph Formulation
MAGNET	Marmara Region Continuous Network
MCLA	Meta-CLustering Algorithm
NAF	North Anatolian Fault
NMF	Non-negative Matrix Factorization
TNPGN	Turkish National Permanent GNSS Network
TUBITAK	Scientifc and Technical Research Council of Turkey

References

1. McClusky, S.; Balassanian, S.; Barka, A.; Demir, C.; Ergintav, S.; Georgiev, I.; Gurkan, O.; Hamburger, M.; Hurst, K.; Kahle, H.; et al. Global Positioning System constraints on plate kinematics and dynamics in the eastern Mediterranean and Caucasus. *J. Geophys. Res. Solid Earth* **2000**, *105*, 5695–5719.
2. Lazos, I.; Papanikolaou, I.; Sboras, S.; Foumelis, M.; Pikridas, C. Geodetic Upper Crust Deformation Based on Primary GNSS and INSAR Data in the Strymon Basin, Northern Greece—Correlation with Active Faults. *Appl. Sci.* **2022**, *12*, 9391.
3. Reilinger, R.; McClusky, S.; Paradissis, D.; Ergintav, S.; Vernant, P. Geodetic constraints on the tectonic evolution of the Aegean region and strain accumulation along the Hellenic subduction zone. *Tectonophysics* **2010**, *488*, 22–30.
4. Thatcher, W. How the continents deform: The evidence from tectonic geodesy. *Annu. Rev. Earth Planet. Sci.* **2009**, *37*, 237–262. [CrossRef]
5. Reilinger, R.; McClusky, S.; Vernant, P.; Lawrence, S.; Ergintav, S.; Cakmak, R.; Ozener, H.; Kadirov, F.; Guliev, I.; Stepanyan, R.; et al. GPS constraints on continental deformation in the Africa-Arabia-Eurasia continental collision zone and implications for the dynamics of plate interactions. *J. Geophys. Res. Solid Earth* **2006**, *111*, B5.
6. Vernant, P. What can we learn from 20 years of interseismic GPS measurements across strike-slip faults? *Tectonophysics* **2015**, *644*, 22–39. [CrossRef]
7. Özarpacı, S.; Kılıç, B.; Bayrak, O.C.; Özdemir, A.; Yılmaz, Y.; Floyd, M. Comparative analysis of the optimum cluster number determination algorithms in clustering GPS velocities. *Geophys. J. Int.* **2022**, *232*, 70–80.
8. Vega-Pons, S.; Ruiz-Shulcloper, J. A survey of clustering ensemble algorithms. *Int. J. Pattern Recognit. Artif. Intell.* **2011**, *25*, 337–372. [CrossRef]

9. Golalipour, K.; Akbari, E.; Hamidi, S.S.; Lee, M.; Enayatifar, R. A From clustering to clustering ensemble selection: A review. *Eng. Appl. Artif. Intell.* **2021**, *104*, 104388.
10. Simpson, R.W.; Thatcher, W.; Savage, J.C. Using cluster analysis to organize and explore regional GPS velocities. *Geophys. Res. Lett.* **2012**, *39*, 18.
11. Savage, J.C.; Simpson, R.W. Clustering of GPS velocities in the Mojave Block, southeastern California. *J. Geophys. Res. Solid Earth* **2013**, *118*, 1747–1759. [CrossRef]
12. Savage, J.C.; Simpson, R.W. Clustering of velocities in a GPS network spanning the Sierra Nevada Block, the northern Walker Lane Belt, and the central Nevada Seismic Belt, California-Nevada. *J. Geophys. Res. Solid Earth* **2013**, *118*, 4937–4947.
13. Savage, J.C.; Wells, R.E. Identifying block structure in the Pacific Northwest, USA. *J. Geophys. Res. Solid Earth.* **2015**, *120*, 7905–7916. [CrossRef]
14. Savage, J.C. Euler-vector clustering of GPS velocities defines microplate geometry in southwest Japan. *J. Geophys. Res. Solid Earth* **2018**, *123*, 1954–1968. [CrossRef]
15. Özdemir, S.; Karslıoğlu, M.O. Soft clustering of GPS velocities from a homogeneous permanent network in Turkey. *J. Geod.* **2019**, *93*, 1171–1195.
16. Takahashi, A.; Hashimoto, M.; Hu, J.C.; Takeuchi, K.; Tsai, M.C.; Fukahata, Y. Hierarchical cluster analysis of dense GPS data and examination of the nature of the clusters associated with regional tectonics in Taiwan. *J. Geophys. Res. Solid Earth* **2019**, *124*, 5174–5191.
17. Granat, R.; Donnellan, A.; Heflin, M.; Lyzenga, G.; Glasscoe, M.; Parker, J.; Pierce, M.; Wang, J.; Rundle, J.; Ludwig, L.G. Clustering Analysis Methods for GNSS Observations: A Data-Driven Approach to Identifying California's Major Faults. *Earth Space Sci.* **2021**, *11*, e2021EA001680. [CrossRef]
18. Kleinberg, J. An impossibility theorem for clustering. *Adv. Neural. Inf. Process Syst.* **2002**, *15*, 463–470.
19. Ghaemi, R.; Sulaiman, M.N.; Ibrahim, H.; Mustapha, N. A survey: Clustering ensembles techniques. *World Acad. Sci. Eng. Technol.* **2009**, *50*, 644–653.
20. Strehl, A.; Ghosh, J. Cluster ensembles—A knowledge reuse framework for combining multiple partitions. *J. Mach. Learn. Res.* **2002**, *3*, 583–617.
21. Li, F.; Qian, Y.; Wang, J.; Dang, C.; Jing, L. Clustering ensemble based on sample's stability. *Artif. Intell.* **2019**, *273*, 37–55. [CrossRef]
22. Zhou, P.; Du, L.; Liu, X.; Shen, Y.D.; Fan, M.; Li, X. Self-paced clustering ensemble. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *32*, 1497–1511.
23. Emre, Ö.; Duman, T.Y.; Özalp, S.; Elmacı, H.; Olgun, Ş.; Şaroğlu F. *Açıklamalı Türkiye Diri Fay Haritası. Ölçek 1:1.250.000*; Maden Tetkik ve Arama Genel Müdürlüğü, Özel Yayın Serisi-30: Ankara, Turkey, 2013.
24. Tibshirani, R.; Walther, G.; Hastie, T. Estimating the number of clusters in a data set via the gap statistic. *J. R Stat. Soc. Ser. B Methodol.* **2001**, *63*, 411–423. [CrossRef]
25. Alqurashi, T.; Wang, W. Clustering ensemble method. *Int. J. Mach. Learn. Cybern.* **2019**, *10*, 1227–1246. [CrossRef]
26. Wu, X.; Ma, T.; Cao, J.; Tian, Y.; Alabdulkarim, A. A comparative study of clustering ensemble algorithms. *Comput. Electr. Eng.* **2018**, *68*, 603–615. [CrossRef]
27. Topchy, A.P.; Jain, A.K.; Punch, W. Clustering ensembles: Models of consensus and weak partitions. *IEEE Trans. Pattern Anal. Mach. Intell.* **2005**, *27*, 1866–1881.
28. Ghosh, J.; Acharya, A. Cluster ensembles. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2011**, *1*, 305–315.
29. Hamidi, S.S.; Akbari, E.; Motameni, H. Consensus clustering algorithm based on the automatic partitioning similarity graph. *Data Knowl. Eng.* **2019**, *124*, 101754. [CrossRef]
30. Gionis, A.; Mannila, H.; Tsaparas, P. Clustering aggregation. *ACM Trans. Knowl. Discov. Data.* **2007**, *10*, 341–352.
31. Tsai, C.F.; Hung, C. Cluster ensembles in collaborative filtering recommendation. *Appl. Soft Comput.* **2012**, *12*, 1417–1425. [CrossRef]
32. Yi, J.; Yang, T.; Jin, R.; Jain, A.K.; Mahdavi, M. Robust ensemble clustering by matrix completion. In Proceedings of the IEEE 12th International Conference on Data Mining, Brussels, Belgium, 10–13 December 2013; pp. 1176–1181.
33. Zhang, T.; Ramakrishnan, R.; Livny, M. BIRCH: An efficient data clustering method for very large databases. *ACM Sigmod Rec.* **1996**, *25*, 103–114.
34. Zhang, T.; Ramakrishnan, R.; Livny, M. BIRCH: A new data clustering algorithm and its applications. *Data Min. Knowl. Discov.* **1997**, *1*, 141–182. [CrossRef]
35. MacQueen, J. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley Symposium on Mathematical Statistics and Probability*; University of California Press: Los Angeles, CA, USA, 1967; Volume 1, pp. 281–297.
36. Sculley, D. Web-scale k-means clustering. In Proceedings of the 19th International Conference on World Wide Web, Raleigh, CA, USA, 26–30 April 2010; pp. 1177–1178.
37. Peng, K.; Leung, V.C.; Huang, Q. Clustering approach based on mini batch kmeans for intrusion detection system over big data. *IEEE Access* **2018**, *6*, 11897–11906. [CrossRef]
38. Kaufman, L.; Rousseeuw, P.J. *Finding Groups in Data: An Introduction to Cluster Analysis*; Wiley: New York, NY, USA, 1990.
39. Ward, J.H., Jr. Hierarchical grouping to optimize an objective function. *J. Am. Stat. Assoc.* **1963**, *58*, 236–244.

40. Yan, D.; Huang, L.; Jordan, M.I. Fast approximate spectral clustering. In Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, 28 June–1 July 2009; pp. 907–916.
41. Von Luxburg, U. A tutorial on spectral clustering. *Stat Comput.* **2007**, *17*, 395–416.
42. Chung, F.R.K. *Spectral Graph Theory*; American Mathematical Society: Providence, RI, USA, 1997; Volume 92.
43. Ng, A.; Jordan, M.; Weiss, Y. On spectral clustering: Analysis and an algorithm. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 3–8 December 2001; Volume 14.
44. Zhou, Z.H.; Tang, W. Clusterer ensemble. *Knowl. Based Syst.* **2006**, *19*, 77–83. [CrossRef]
45. Ayad, H.G.; Kamel, M.S. On voting-based consensus of cluster ensembles. *Voting-Based Consens. Clust. Ensembles.* **2010**, *43*, 1943–1953. [CrossRef]
46. Fred, A.L.; Jain, A.K. Combining multiple clusterings using evidence accumulation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2005**, *27*, 835–850. [CrossRef]
47. Fern, X.Z.; Brodley, C.E. Solving cluster ensemble problems by bipartite graph partitioning. In Proceedings of the Twenty-First International Conference on Machine Learning, Banff, AL, Canada, 4–8 July 2004; p. 36.
48. Li, T.; Ding, C.; Jordan, M.I. Solving consensus and semi-supervised clustering problems using nonnegative matrix factorization. In Proceedings of the 7th IEEE International Conference on Data Mining, Omaha, NE, USA, 28–31 October 2007; pp. 577–582.
49. Cichocki, A.; Zdunek, R.; Phan, A.H.; Amari, S.I. *Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-Way Data Analysis and Blind Source Separation*; John Wiley & Sons: Hoboken, NJ, USA, 2009.
50. Vega-Pons, S.; Correa-Morris, J.; Ruiz-Shulcloper, J. Weighted partition consensus via kernels. *Pattern Recognit.* **2010**, *43*, 2712–2724. [CrossRef]
51. Luo, H.; Jing, F.; Xie, X. Combining multiple clusterings using information theory based genetic algorithm. *Int. Conf. Comput. Intell. Secur.* **2006**, *1*, 84–89.
52. Topchy, A.P.; Law, M.H.; Jain, A.K.; Fred, A.L. Analysis of consensus partition in cluster ensemble. In Proceedings of the 4th IEEE International Conference on Data Mining (ICDM'04), Brighton, UK, 1–4 November 2004; pp. 225–232.
53. Liang, W.; Zhang, Y.; Xu, J.; Lin, D. Optimization of basic clustering for ensemble clustering: An information-theoretic perspective. *IEEE Access.* **2019**, *7*, 179048–179062.
54. Karypis, G.; Kumar, V. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM J. Sci. Comput.* **1998**, *20*, 359–392. [CrossRef]
55. Strehl, A.; Ghosh, J. Value-based customer grouping from large retail data sets. *Data Min Knowl Discov. Theory Tools Technol.* **2000**, *4057*, 33–42.
56. Paatero, P.; Tapper, U. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics* **1994**, *5*, 111–126. [CrossRef]
57. Lee, D.; Seung, H.S. Algorithms for non-negative matrix factorization. In Proceedings of the Advances in Neural Information Processing Systems 13, Denver, CO, USA, 1 January 2000; Volume 13.
58. Xu, D.; Tian, Y. A comprehensive survey of clustering algorithms. *Ann. Data Sci.* **2015**, *2*, 165–193. [CrossRef]
59. Li, X.; Chen, M.; Wang, Q. Discrimination-aware projected matrix factorization. *IEEE Trans. Knowl. Data Eng.* **2019**, *32*, 809–814.
60. Wessel, P.; Luis, J.F.; Uieda, L.; Scharroo, R.; Wobbe, F.; Smith, W.H.F.; Tian, D. The Generic Mapping Tools version 6. *Geochem. Geophys. Geosystems.* **2019**, *20*, 5556–5564.