Journal of Ambient Intelligence and Humanized Computing (2019) 10:2029–2040
https://doi.org/10.1007/s12652-018-0805-4

ORIGINAL RESEARCH

Feature space learning model

Renchu Guan[1] · Xu Wang[1] · Maurizio Marchese[2] · Mary Qu Yang[3] · Yanchun Liang[1,5] · Chen Yang[4]

Received: 1 September 2017 / Accepted: 20 April 2018 / Published online: 9 May 2018

Abstract
With the massive volume and rapid increasing of data, feature space study is of great importance. To avoid the complex training processes in deep learning models which project original feature space into low-dimensional ones, we propose a novel feature space learning (FSL) model. The main contributions in our approach are: (1) FSL can not only select useful features but also adaptively update feature values and span new feature spaces; (2) four FSL algorithms are proposed with the feature space updating procedure; (3) FSL can provide a better data understanding and learn descriptive and compact feature spaces without the tough training for deep architectures. Experimental results on benchmark data sets demonstrate that FSL-based algorithms performed better than the classical unsupervised, semi-supervised learning and even incremental semi-supervised algorithms. In addition, we show a visualization of the learned feature space results. With the carefully designed learning strategy, FSL dynamically disentangles explanatory factors, depresses the noise accumulation and semantic shift, and constructs easy-to-understand feature spaces.

Keywords Feature space learning · Semi-supervised learning · Affinity Propagation · k-means

1 Introduction

In the era of big data, tasks such as natural language processing and ImageNet large scale visual recognition competition make that it is not enough if we just rely on simple parametric models, because they cannot capture enough complexity of interest unless provided with the appropriate feature space (Bengio et al. 2013). However, how to explore and generate the feature space to support effective machine learning is a major question. Recently, much of the actual effort in deploying deep learning algorithms such as deep belief networks (Jiang et al. 2016), auto-encoders (Hinton and Salakhutdinov 2006), convolutional neural network (Esteva et al. 2017) and recurrent neural networks (Zhang et al. 2018) goes into exploring feature space and learning good representations; however, most of the deep architectures are too challenging to train effectively. Another problem lies in disentangling and explaining the highly abstracted concepts or representations obtained from deep learning. The source of their performance is still lack of interpretability (Karpathy et al. 2015) (See Table 1).

Meanwhile, among the data mining techniques, clustering plays an important role in exploratory recommendation systems (Bobadilla et al. 2013), public opinion analyses (Semetko and Valkenburg 2000), and information retrieval areas (Frakes and Baeza-Yates 1992). Many clustering applications can be found in image segmentation, object recognition, video tracking, and etc. (Jain et al. 1999; Huang et al. 2018; Wu et al. 2018). Instead of only using the unlabeled sources, semi-supervised algorithms have attracted considerable attention because they can learn from a combination of labeled and unlabeled data for better performance (Wang et al. 2012; Guan et al. 2011). In semi-supervised clustering,

✉ Chen Yang
yangc616@jlu.edu.cn

[1] Key Laboratory for Symbol Computation and Knowledge Engineering of National Education Ministry, College of Computer Science and Technology, Jilin University, Changchun 130012, China

[2] Department of Engineering and Computer Science, University of Trento, 9I-38123 Povo, Italy

[3] MidSouth Bioinformatics Center and Joint Bioinformatics, University of Arkansas at Little Rock and University of Arkansas Medical Sciences, Little Rock, AR 72204, USA

[4] College of Earth Sciences, Jilin University, Changchun 130061, China

[5] Zhuhai Laboratory of Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Zhuhai College of Jilin University, Zhuhai 519041, China

Table 1 Pros and cons of the introduced models

Models	Pros and Cons	References
Deep learning algorithms	Pros: They can explore feature space and learning good representations. Cons: Deep architectures are too challenging to train effectively and lack of interpretability	Deep belief networks (Jiang et al. 2016), auto-encoders (Hinton and Salakhutdinov 2006), convolutional neural network (Esteva et al. 2017) and recurrent neural networks (Zhang et al. 2018; Karpathy et al. 2015)
Clustering	Pros: Only using the unlabeled sources can detect intrinsic distribution. Cons: Only using the unlabeled sources is lack of guidance for special aim	Recommendation systems (Bobadilla et al. 2013), public opinion analyses, (Semetko and Valkenburg 2000), and information retrieval areas (Frakes and Baeza-Yates 1992), image segmentation, object recognition, video tracking, and etc. (Jain et al. 1999; Huang et al. 2018; Wu et al. 2018; Guan et al. 2011)
Semi-supervised clustering	Pros: Use the information both from labeled data and unlabeled data. Cons: Unlabeled data should be explored carefully and it performed without feature learning procedure	Li and Zhou (2015); Tang et al. (2007); Wang et al. (2012); Guan et al. (2011); Shi et al. (2009); Xue et al. (2011)

the intuitive purpose is to use not only the available information from labeled data during clustering procedures, but also clues from unlabeled data to estimate data distribution. To exploit unlabeled data, most of the frameworks introduce the following two prior assumptions of consistency: (1) nearby points are likely to have the same label; and (2) points on the same structure (typically referred to a cluster or a manifold) are likely to have the same label (Xue et al. 2011). However, the biggest risk for directly using unlabeled samples is that the cluster could be adulterated with untrusted samples in its earlier stage. These bad pieces of information accompanied with good will propagate and be amplified in the following procedures and in other samples, which is similar to positive feedback in digital electronics and the Matthew effect in social psychology. Therefore, it is obvious that unlabeled points should be explored carefully because even the supervised classification performance can be degenerated by wrong unlabeled information (Li and Zhou 2015). In addition, most existing semi-supervised methods are lack of the ability for handling high-dimensional data (Tang et al. 2007).

To make good use of daily growing data (most of them are unlabeled samples) and avoid the wrong information propagation risk, we propose a novel feature space learning (FSL) model, which can perform adaptive feature space upgrading while fulfilling clustering. It is based on the hypothesis that unlabeled samples can provide useful information on distribution estimation (cluster center estimation) over feature space. Therefore, the new model creates label propagation in unlabeled texts with the help of clustering. Then, the incoming newly labeled samples are selected based on the objective function, which most of the clustering algorithms try to minimize. Moreover, a feature selection method inspired by a universal regularity for human language-Zipf's law (Zipf 1949) and word burstiness (Kleinberg 2002) is developed to further control the risks. This model relies on the universal rules for human language which are named as Zipfs law and word burstiness. Because it uses an algorithm instead of functional mapping to dynamically delineate the feature spaces, FSL is quite different from the mean-shift algorithm, which is another feature space model in image segmentation and video tracking (Comaniciu and Meer 2002; Leichter 2012). This model combines prior information and an assumption of consistency, which could not only embed the labeled information in similarity measurements, but also guide the clustering procedures.

To illustrate the performance of our model, we applied FSL to two classical clustering algorithms and implemented four FSL algorithms: feature space seeded k-means (FSSK-means), feature space constrained k-means (FSCK-means), feature space affinity propagation (FSAP) and feature space seeds affinity propagation (FSSAP). Experiment was conducted with two benchmark data sets to demonstrate the

effectiveness of the proposed algorithms. As a result, topical feature space for each cluster can be found accompanying the end of clustering. With the learned feature space, we can obtain a simple and compact representation of the data.

2 Feature space learning model

Studying the statistical properties and universal regularities of written texts can dig out clues about how our brains process information and model language computationally (Serrano et al. 2009). Among the studies in this area, the most notable regularities are Zipf's law and the bursty nature of words.

Zipf's law is one of the best-known universal regularities on word frequencies, wherein the frequency of terms n_i in a collection decreases inversely to the rank r of the terms: $n_i \sim 1/r_i$ or $P(n_i) \sim n_i^{-\alpha} \alpha \approx 2$ which indicates n_i is approximated by the power law. It applies to collections of texts in virtually all languages.

Due to the bursty nature of human behavior (Barabasi 2005) and the fact that bursty nature of rare words is connected with the topical organization of texts (Griffiths and Steyvers 2004), word bursts have attracted more and more attention. It is depicted as making a word f more likely to reappear in document d if it has already appeared, compared to its overall frequency across the collection (Serrano et al. 2009). Interestingly, those rare words are more evident with this property and they are more likely to be topical. To mine the potential information and use the consistency assumption, we introduced Zipf's law and word bursts to control the potential risk while learning from both labeled and unlabeled samples.

Based on Zipf's law and word bursts, the mainframe of the new feature space learning model is presented. Our hypothesis is that unlabeled samples can provide information about the distribution estimation (cluster center estimation) over feature space. For example, suppose that we determine that the word "computer" in a labeled document tends to be an important feature for its cluster vector. The most widely used text-mining model is the vector space model, which treats a document as a bag of words/phrases and uses plain language words as features. If we use this feature or "computer" to estimate the clusters of many unlabeled documents, we could find that the word "graphics" occurs frequently in the unlabeled examples that are now believed to belong to the "computer" cluster. In contrast, based on the prior assumption of consistency, it could also be expected that points (documents) with the same label are likely to share same or similar feature space. Therefore, the unlabeled samples can provide additional informative features to construct new feature space to provide further cluster estimation or change the cluster center vector to be more representative. However, based on the above hypotheses, there is a potential risk that noisy information may be picked out. To avoid the risk, we designed two constraint strategies–Zipfs' law and word bursts in our feature space learning (FSL) model to optimize the objective function.

To clarify the FSL model and procedures, all the cluster processes are illustrated in Fig. 1. Here, all the detailed explanations are depicted as follows:

1. Initialization: Initialize data to supersets to convert all the texts into vectors. Let $D = \{d_1, d_2, \cdots, d_N\}$ be a set of samples. Suppose that d_i and d_j are two objects in D; they can be represented as: $d_i = \{\langle f_i^1, n_i^1\rangle, \langle f_i^2, n_i^2\rangle, \cdots, \langle f_i^{L_i}, n_i^{L_i}\rangle\}$ $d_j = \{\langle f_j^1, n_j^1\rangle, \langle f_j^2, n_j^2\rangle, \cdots, \langle f_j^{L_j}, n_j^{L_j}\rangle\}$. where f_i^l and $f_j^m (1 \le l \le L_i, 1 \le m \le L_j)$ represent the lth feature of d_i and the mth feature of d_j, respectively. n_i^l and n_j^m are their feature values. L_i and L_j are the number of the objects' features.
2. Seed construction: An intuitive way is to use the few-labeled samples. However, after step 4 and step 5, the seeds can be updated with rules.
3. Similarity computation: Different similarity metrics can be selected according to different data, such as cosine coefficient for texts and Euclidean distance for images.
4. Clustering: In this step, several classical clustering algorithms such as k-means and affinity propagation in our case can be adopted.
5. Feature Space control: It is the key procedure of our model and is designed to avoid wrong updating with noise data and features. The right part of Fig. 1 is a diagram of feature space updating (The change of frame and arrows indicate the feature space tranformation). It is described in detail in the next section.
6. The termination condition judgment. If clusters are not changed for several iterations or the maximum number of iterations value is reached, then the clusters and their topic feature space are generated.

When we face a real problem, three things should be emphasized in the feature space updating: first, how to update feature space centers.

Definition 1 The sample d^* is a trust sample for the kth cluster ($Trust_k$), if

$$d^* = \arg\max_{d \in Cluster_k} Mem(d, k), 1 \le k \le K \tag{1}$$

where K is the number of clusters, $Mem(d, k)$ is the membership function which indicates the extent of sample d belonging to cluster k. Different clustering algorithms try to

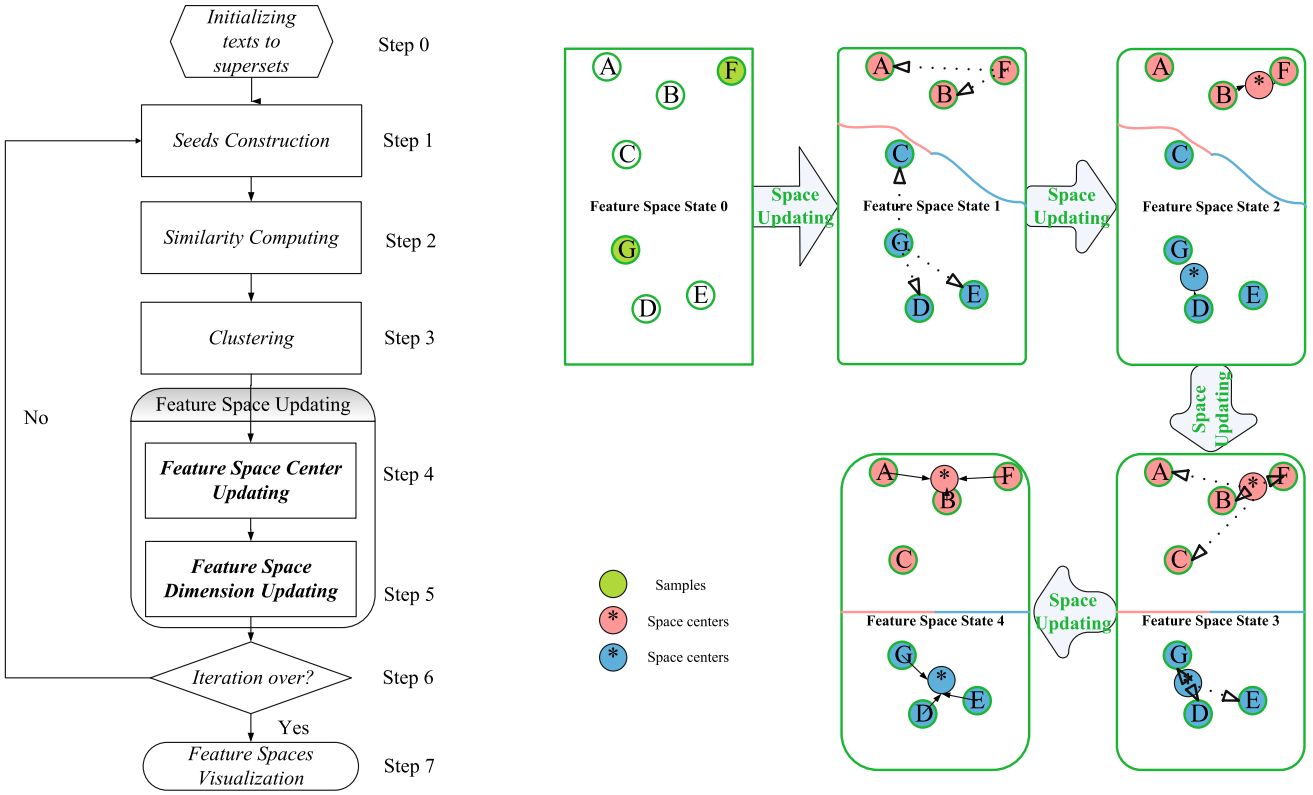

Fig. 1 Flowchart of feature space learning model and feature space updating diagram

maximize different membership functions, or to be equivalent, to minimize an object function opposed to membership functions. Therefore, trusted samples are selected by the algorithms and will provide useful update information for centers in the following processes. The count of "trust samples" could vary according to different applications (such as image processing, gene finding and so on) and requirements.

Moreover, the feature space dimension updating step provides another "firewall" for the security of adaptive information obtained from trust samples. It is believed that not all the features in the input space are important for clustering. This is because some of the features may be redundant or irrelevant to the cluster topics. Some may even misguide the clustering result, especially when the irrelevant features outnumber the relevant ones. Therefore, a great number of feature selection methods have been proposed, which could not only decrease the time and space complexity but also achieve improvements on clustering results. Instead of dimension reduction, we focus on the rich-information feature (e.g. words or key-phrases for text clustering) finding. We should emphasize that these features are selected from both labeled and unlabeled sources to generate a new space but not to reduce the dimension as an original purpose of feature selection for text mining. It is implemented to control the unlabeled utilization risk and select the burst items for providing adaptive information.

As the second issue, feature space updating needs an effective risk control strategy to avoid the risk of untrusted labels; meanwhile, the risk control strategy cannot be too complex to avoid the high computation complexity. Here we propose a simple but effective feature selection method with linear computational complexity based on Zipf's law and word burstiness. It borrows the ideas from the bursty nature of words and contributes to further extension, which indicates that bursty words in each individual text (with stop words or function words removed) are potential to be the bursty words in the corresponding cluster. These words are more informative and contain more relationships with each other. Furthermore, they could be the major part of the topic feature space. We call these words as rich-information features (RIFs).

Definition 2 Assume sample d^* is a trust sample of the kth cluster, and the lth feature f_l is a feature of sample d^*. Then f_l is a rich-information feature (RIF) for cluster k, if

$$n_*^l \geq \eta \sum_{l=1}^{L_*} n^l / L_*, m_k^i \geq \mu \sum_{p=1}^{P_k} m_k^p / P_k \tag{2}$$

where n_*^l is the frequency of feature f_i in trust sample d^*, m_i^k is the frequency of feature f_i in the kth cluster, L_* and P_k indicate the number of features in both d^* and the kth cluster center while η and μ are two control parameters, respectively.

Definition 2 is a double constraints problem. It is derived from mutual information in information theory. The *RIF* selecting method is easily scaled to a "big" dataset, because of its linear computation complexity $O(M + L)$.

Third, after the harsh feature selection, another rule is developed for updating weight in the adaptive processes to optimize the feature space. The feature space is constructed by adding rich-information features of trust samples into the original space iteratively. Therefore, the feature space can be considered as a linear combination of the vector of trust sample *RIF* and the original feature space. It should be noticed that the trust samples have different confidence levels at different iterations. Simply, the confidence of an *RIF* in t iteration is assigned as

$$conf_t = (T - t)/T \tag{3}$$

where t is the iteration times and T is the total number of iterations. Herein, the *RIF*s of the unlabeled samples' contribution for cluster feature space is decreased linearly with iteration t. Denote the total confidence from the beginning until the tth iteration as

$$conf_T = \sum_{i=1}^{t} conf_i = \sum_{i=1}^{t} \frac{T-t}{T} = t\left(1 - \frac{t+1}{2T}\right) \tag{4}$$

Then, for an RIF f_i in trust sample d^* of cluster k in the tth iteration, the weight updating rule (See step 5 in Figure 2) is set as follows:

$$(w_k^i)^t = \frac{conf_t}{conf_T} \times (w_k^i)^{t-1} + \frac{conf_t}{conf_T} \times n_*^i \tag{5}$$

where $(w_i^k)^t$ is the weight of f_i (a *RIF*) in cluster k for the tth iteration. In addition, considering the consistency assumption of semi-supervised learning, the similarity matrix needs to be updated for those new labeled samples. The matrix is re-computed as follows: If i and j belong to same cluster at the t iteration, their distance is set to the minimum; otherwise, the distance becomes the maximum. In particular, at the beginning of the algorithm, the similarities among all the labeled objects are also computed with this rule.

To examine the effectiveness of FSL, starting from two classical algorithms (k-means and affinity propagation) (Frey and Dueck 2007), we represent four FSL algorithms. These algorithms are named as feature space seed k-means (FSSK-means), feature space constrained k-means (FSK-means), feature space affinity propagation (FSAP), and feature-space-seeds affinity propagation (FSSAP).

2.1 K-means based FSL models

The main idea of k-means is to optimize the objective function:

$$G(x) = \sum_{k=1}^{K} \sum_{d_i \in Cluster_k} ||d_i - c_k||^2 \tag{6}$$

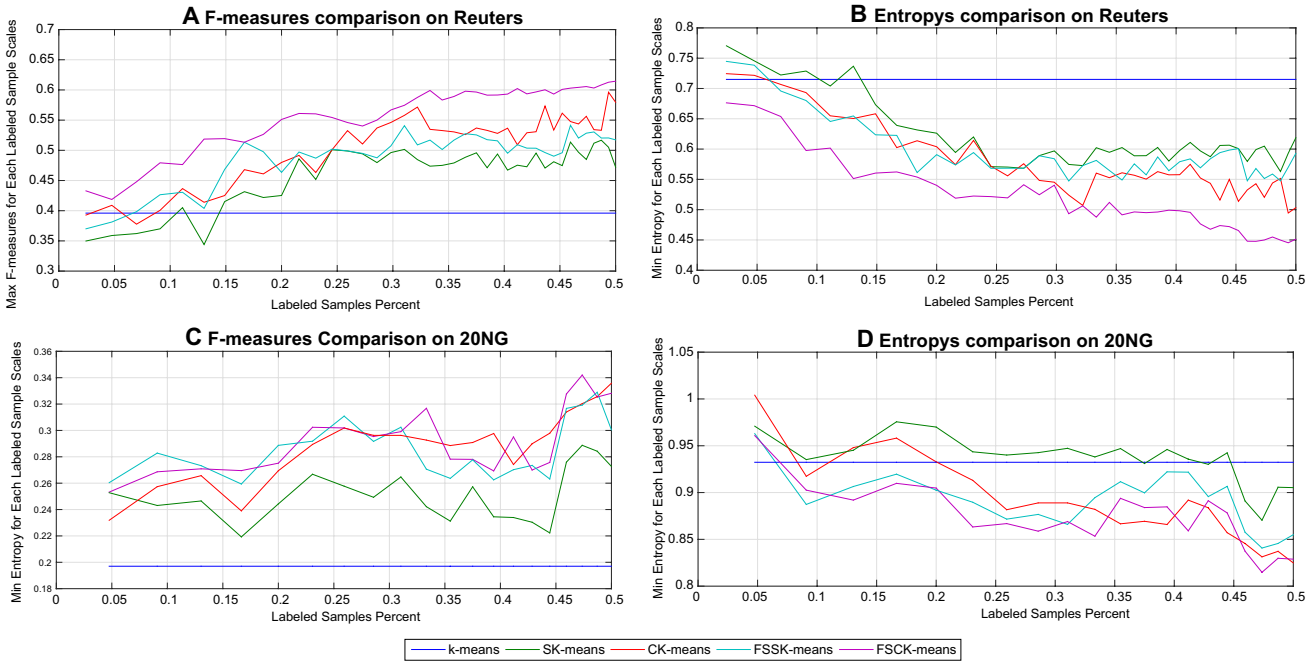

Fig. 2 Comparison results of K-means based methods

where K is the cluster number, d_i is a sample involved in cluster k and c_k is the cluster center, respectively. After the initialization with labeled samples and a complete k-means clustering, the next step is to select the trust samples. According to Eq. (6), the membership function in Eq. (1) should minimize the distance of the relative sample with its cluster center. Then the trust sample selection method could be represented by:

$$d^* = \arg\min_{d \in Cluster_k} ||d - c_k||^2, 1 \le k \le K \tag{7}$$

In the next step, those features satisfied in Eq. (2) are selected as RIFs and their weights are updated with Eq. (5). Moreover, it is very important in semi-supervised clustering to make full use of the information embedded in the labeled objects. The details of the k-means based clustering algorithms are as follows:

Feature-space-seed k-means (*FSSK-means)*: The labeled sample is used to initialize the k-means algorithm. Rather than initializing k-means randomly, the kth cluster is initialized with the mean of the kth partition of the labeled set. The labeled samples are only used for initialization, and they are not used in the following steps of the algorithm, such as re-estimating means and reassigning clusters in k-means clustering procedures. Taking into consideration on our specific application domain (text clustering), we utilized the classical similarity measurement cosine coefficient:

$$S(X, Y) = |X \cap Y|/(|X|^{1/2}|Y|^{1/2}) \tag{8}$$

where X and Y are two texts. *FSSK-means* is based on the semi-supervised k-means algorithms proposed by (Basu et al. 2002). *FSSK-means* inherits the labeled sample learning strategy, but adds the utilization of unlabeled samples with a strict feature space learning strategy (such as trust sample and *RIF* selection).

Feature-space-constrained k-means (*FSCK-means)*: The labeled sample set is also used to initialize the k-means algorithm as described for *FSCK-means*. However, in the subsequent procedures of *FSCK-means*, the labeled samples are employed in the clustering phase. This means that when the algorithm needs to re-estimate means and reassign clusters, the labeled samples join in the computation. In contrast, labeled samples in *FSSK-means* are only used in initialization. Equation (8) is also used in *FSCK-means* to measure the similarity.

2.2 Affinity propagation based FSL models

The core idea of affinity propagation (AP) is that it can be viewed as a method that searches for the minima of an energy function:

$$G(x) = -\sum_{i=1}^{N} s(d_i, c_k) \tag{9}$$

where N is the data number, k is the cluster index number, d_i is a sample involved in cluster k and c_k is the cluster center.

After the AP clustering processes, we consider the sample as a trust sample according to the above AP's objective function and Eq. (1):

$$d^* = \arg\min_{1 \le i,k \le N}[-(s(i,k) + s(k,k))] \tag{10}$$

where $s(i, k)$ is the similarity between sample d_i and its center c_k and $s(k, k)$ is the priori suitability of point c_k to serve as an exemplar. Similarly, Eq. (2) is used to search for the RIFs and Eq. (5) is used to update their weights.

Feature-space affinity propagation (*FSAP*) is performed on the basis of AP clustering. Labeled samples are directly used during the responsibility and availability messages transmission. It utilizes the cosine coefficient to measure the similarity between documents, and the self-similarity utilizes:

$$s(i,i) = \begin{cases} P(x) - \phi(Q(x) - P(x)) \\ +\infty, \quad N < i \le M \end{cases} \tag{11}$$

where N is the scale of the unlabeled data set, M indicates the scale of all the labeled and unlabeled samples, $P(x) = \min_{1 \le i,j \le N, i \ne j}\{s(i,j)\}, 1 < i \le N, Q(x) = \max_{1 \le i,j \le N, i \ne j}\{s(i,j)\}$, $1 < i \le N$ and ϕ is an adaptive factor. Moreover, in the FSL frame, we use Eq. (10) to select the trust samples. Then, with the rules of Eqs. (2) and (3), FSAP achieves the updated feature space.

Feature-space-seeds affinity propagation (*FSSAP*) *FSSAP* is derived from the combination of seeds affinity propagation (SAP) (Guan et al. 2011) and the FSL model. SAP related concepts are introduced: For document d_i, we denote F_i as the feature set of d_i, and SF_i as the significant feature set of d_i, including the most significant features–such as the words and key-phrases in title and abstract of d_i. Then for two document samples d_i and d_j, the co-feature set $CFS_{(i,j)}$, unilateral feature set $UFS_{(i,j)}$, and significant co-feature set $SCS_{(i,j)}$ are defined as follows:

$$CFS_{(i,j)} = \{f | f \in Fi \text{ and } f \in F_j\} \tag{12}$$

$$UFS_{(i,j)} = \{f | f \in Fi \text{ and } f \notin F_j\} \tag{13}$$

$$SCS_{(i,j)} = \{f | f \in Fi \text{ and } f \in SF_j\} \tag{14}$$

FSSAP inherits the tri-set similarity measurement from SAP:

$$s(i,j) = \alpha \sum_{m1}^{|CFS|} n_j^m + \beta \sum_{q=1}^{|SCS|} n_{SF_j}^q - \gamma \sum_{p=1}^{|UFS|} n_i^q \tag{15}$$

where $|\cdot|$ indicates the scale of a set; n_j^m and n_i^q have a frequency of f_q in SF_j, and α, β and γ are adaptive factors that have been extensively discussed in Guan et al. (2011). The

labeled samples are first merged into compact seeds based on the labeled information before putting labeled and unlabeled samples together. Then, the compact seeds are entered into the message-passing of AP. While in the learning process, it should be noticed that the features are divided into a normal feature set and a significant feature set. Denote FC_k as the normal feature set of cluster k, and SFC_k as the significant feature set of cluster k. Then after obtaining the trust sample d^* of cluster k, if a feature of Eq. (2) is satisfied, it should be added into FC_k. Moreover, if it satisfies

$$m^i_{SF_k} \geq \sum_{p=1}^{SFC_k} m^p_{SF_k} / |SFC_k| \tag{16}$$

then it should be added into SFC_k, where $m^i_{SF_k}$ represents the frequency of feature f_i in SFC_k. Comparing FSAP and FSSAP, the most different strategy between them is that the latter chooses tri-set similarity measurement and compact seeds.

To compare the introduced ten clustering algorithms, Table 2 depicts all the different strategies. The main distinction among between semi-supervised learning and FSL laid on the different assumptions. The former only focus on the performance improvement. On the contrary, the latter not only consider the performance but also optimize the feature space. It is a bi-objective optimization.

3 Results and discussion

3.1 Datasets and evaluation

All the classical clustering, semi-supervised learning and new proposed FSL algorithms are applied to two benchmark text datasets: Reuters-21578 (Reuters) and 20 Newsgroups (20NG). They are maintained in the UC Irvine Machine Learning Repository and are widely used (Asuncion and Newman 2007; Bekkerman et al. 2003). Moreover, for text data clustering and classification, the high-dimensional and sparse matrix computation is a typical problem (Guan et al. 2011). FSL model transfers the original feature space into compact ones which can solve this problem.

Table 2 Different learning strategies for related algorithms

	Tri-Set similarity	Semi-supervised	FSL
k-means	×	×	×
AP(CC)	×	×	×
SK-means	×	✓	×
CK-means	×	✓	×
SAP (CC)	×	✓	×
SAP	✓	✓	×
FSSK-means	×	✓	✓
FSCK-means	×	✓	✓
FSAP	×	✓	✓
FSSAP	✓	✓	✓

The publicly available Reuters dataset is a widely used benchmark text mining data set which is pre-classified manually (Lewis 2004). This class information is eliminated before learning, and is used to evaluate the performance of each algorithm at the end. The original Reuters data consist of 22 files (for a total of 21,578 documents) and contains special tags like < *TITLE* >,< *TOPICS* >, and < *DATE* > among others, which are the text information and introduction. We firstly cut the files into a series of single texts and strips the documents from the special tags. Then, those documents which belong to at least one topic are selected. To avoid the imbalanced data problem in Reuters, the top 10 classes are selected as other researches (Estabrooks et al. 2004; Guan et al. 2011).

20NG is also a widely used benchmark data set. It is collected by Ken Lang and contains 19997 texts from 20 news groups (Rennie 2008). The original 20NG contains a large number of headers information (such as: Newsgroups , Subject , and Date) in each document. These headers are deleted before the experiments to avoid introducing label information.

The pre-processing includes text extraction, stop words removal and word frequency computation for each document, the data sets were changed into the superset form in Step 0 of Fig. 1. The labeled samples used in all of the algorithms are randomly selected without any prior knowledge. The count of the unlabeled data is 400. To pursue the compact feature space, for Reuters, the count of the labeled data is from 10 to 400 (2.5 to 50%); for 20NG, the count is from 20 to 400 (5 to 50%).

To evaluate the performance of clustering, two types of measures were applied, namely F-measure and entropy, which has been widely used in information retrieval. They are used to compare the generated result with the set of categories created by experts. The F-measure is a harmonic combination of the precision and recall values. The larger the F-measure is, the better is the clustering performance. Entropy provides a measure of the uniformity or purity of a cluster. In other words, it can tell us how homogeneous a cluster is. The smaller the Entropy is, the better the clustering performance.

3.2 FSL vs. semi-supervised algorithms

To examine the effectiveness of the proposed model, several existing algorithms were implemented for comparison, the blue line represents k-means algorithm, the green line is SK-means, the red line CK-means, the cyan line is FSSK-means

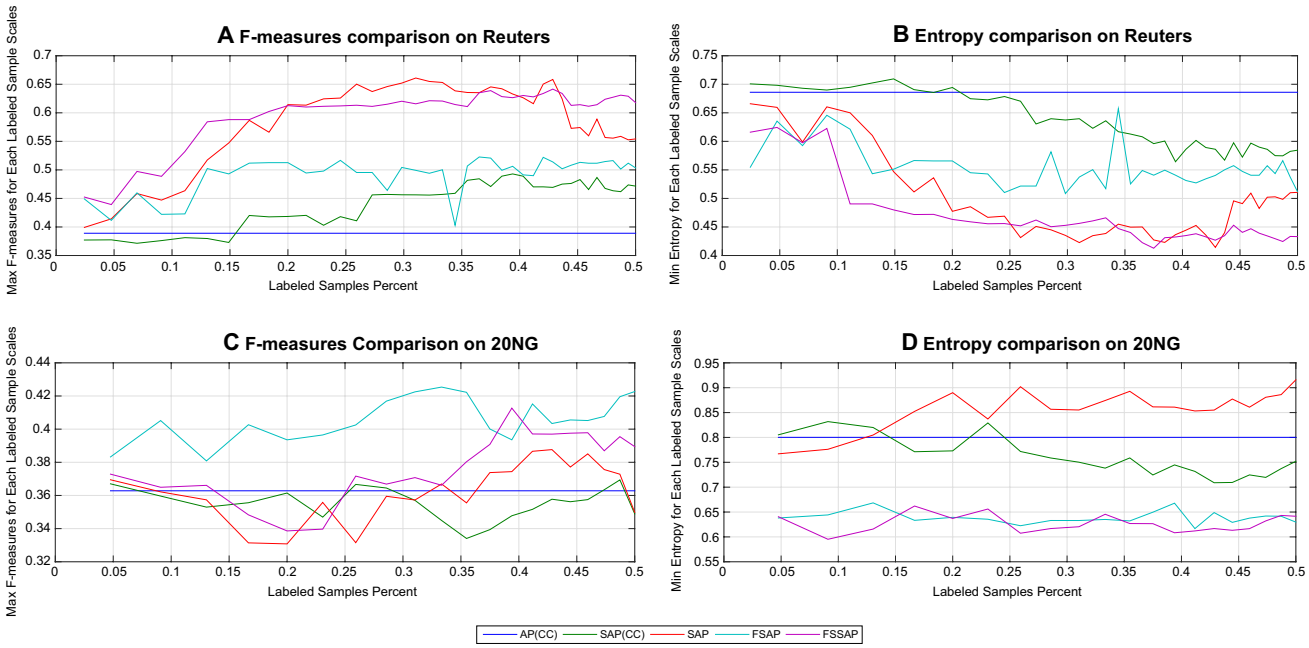

Fig. 3 Comparison results of AP based methods

and the purple line is FSCK-means. The x axes indicate the percentage of labeled samples in the datasets, the y axes of Fig. 2a, c are the maximum F-measure for each labeled sample scales, and the y axes of Fig. 2b, d are the minimum Entropy for each labeled sample scales. From Fig. 2, we can clearly see that the FSL learning curves (FSCK-means in purple line and FSSK-means in cyan line) are the highest F-measure and lowest entropy. They beat the other three algorithms on different labeled sample scales.

The experiments show that our feature space learning algorithms outperform the classical k-means and semi-supervised k-means. The effectiveness of the FSL model can be seen in the distance between the classical semi-supervised clustering and feature space learning curves. We compute the mean value for each algorithm's learning curve in Fig. 2 and the numerical values of the maximum/minimum of these k-means based algorithms. They are showed in Table 3. From Table 3, we can see FSL based k-means algorithms are superior to the semi-supervised k-means clustering algorithms. FSCK-means achieved the highest F-measure (0.614 and 0.342 for Reuters and 20NG, respectively) and lowest Entropy (0.445 and 0.815 for Reuters and 20NG, respectively). For the mean value of each learning curve, it can be found that the FSL algorithms also achieved the highest mean F-measure (0.546 and 0.289 for Reuters and 20NG, respectively) and the lowest entropy (0.533 and 0.889 for Reuters and 20NG, respectively). All the two FSL algorithms are performed better than the ones without FSL strategies, e.g. FSSK-means gets 13.8% higher F-measure than SK-means and 73.6% higher than k-means (0.197) on 20NG data.

The comparison results of AP based algorithms are depicted in Fig. 3. In Fig. 3, the blue line represents AP(CC) algorithm, the green line is SAP(CC), the red line SAP, the cyan line is FSAP and the purple is FSSAP. The x axes

Table 3 Semi-supervised K-means VS FSL K-means

Data	Evaluation	SK-means	CK-means	FSSK-means	FSCK-means
Reuters	Max-F	0.517	0.600	0.542	**0.614**
	Min-E	0.563	0.494	0.547	**0.445**
	Mean-F	0.462	0.509	0.477	**0.546**
	Mean-E	0.619	0.576	0.611	**0.533**
20 Newsgroup	Max-F	0.289	0.336	0.329	**0.342**
	Min-E	0.870	0.825	0.841	**0.815**
	Mean-F	0.250	0.288	0.276	**0.289**
	Mean-E	0.936	0.890	0.906	**0.889**

The best results are in bold

indicate the percentage of labeled samples in the datasets, the y axes of Fig. 3a, c are the maximum F-measure for each labeled sample scales, and the y axes of Fig.3b, d are the minimum Entropy for each labeled sample scales. In Fig.3a, b, because SAP (red line) used the tri-set similarity, its learning curve is near FSSAP (purple line) on Reuters. However, on 20NG the gap between FSSAP and SAP is enlarged in Fig. 3c, d. In addition, on 20NG, FSAP achieved the highest F-measures in Fig. 3c. In summary, the FSL models FSAP and FSSAP broadly performed better than those without. Combined with the results of k-means based algorithms, it illustrates that the FSL model is not sensitive to the exact algorithms and can be used in a variety of clustering methods.

From the above experiments and results, it can be seen that the FSL model based algorithms could achieve better results than the traditional semi-supervised clustering algorithms on different data sets. It is because that by updating the feature space, the FSL model could modify the clustering strategy and embed information in similarity measurement simultaneously. The superior experimental results show that FSL can find out those important features and construct an informative feature space. Moreover, the FSL model can combine with different clustering algorithms not limited to the k-means and affinity propagation.

3.3 FSL vs. incremental semi-supervised algorithm

The incremental affinity propagation (IAP) is an existing AP based semi-supervised algorithm (Shi et al. 2009) which can be considered as SAP (CC) plus an incremental trust sample selection process, while FSAP is SAP (CC) combined with FSL. To test the only effect of the feature space learning procedure, the result of FSAP is compared with that of IAP in Shi et al. 2009. The F-measure and entropy comparison results are shown in Tables 4 and 5.

Tables 4 and 5 show that FSSAP with feature space learning receives the best values (the maximum F-measures and minimum entropies) in most cases, i.e. FSSAP performs better than IAP on five data scales (10, 100, 200, 300, and 400) on both F-measure and entropy. Most importantly, FSSAP gets 14.3% higher F-measure than IAP with a data scale of 100 and a 13.5% lower entropy score than IAP at 200.

Table 4 FSL AP vs IAP on F-measure

	10	50	100	200	300	400
IAP	0.456	0.465	0.449	0.459	0.465	0.465
FSAP	**0.503**	0.423	0.513	0.500	0.514	0.503
FSSAP	0.452	**0.532**	**0.612**	**0.621**	**0.642**	**0.618**

The best results are in bold

Table 5 FSL AP vs IAP on entropy

	10	50	100	200	300	400
IAP	0.698	0.594	0.594	0.598	0.598	0.596
FSAP	**0.555**	0.621	0.565	0.517	0.540	0.512
FSSAP	0.616	**0.490**	**0.463**	**0.466**	**0.427**	**0.433**

The best results are in bold

Table 6 Learned feature spaces

Algorithms	Clusters	Features' count	Example features
FSCK-means	1	2434	Aid = 3.37; share = 1.84; men = 1.72; ers = 1.54; dlr = 1.52
	2	2233	Pro = 3.93; pro = 3.93; aid = 3.71; mln = 3.07; est = 3.53; acre = 2.89
	3	3364	Aid = 3.37; aid = 6.02; ill = 3.52; pct = 3.11; pro = 3.08; pri = 2.70
	4	1109	Aid = 3.37; mln = 3.64; loss = 1.93 net = 1.70; pro = 1.55; dlr = 1.51
	5	2633	Aid = 3.37; pro = 3.66; aid = 3.13; ers = 2.49; men = 2.44; eat = 2.17
	6	2281	Aid = 3.37; ban = 3.79; rate = 3.66; bank = 3.47; pct = 3.11;aid = 2.56
	7	2652	Aid = 3.37; ban = 4.56; aid = 4.47; bank = 4.42; int = 2.65;dollar = 2.45
	8	2956	Aid = 3.37; aid = 3.66; ran = 2.67; ers = 2.47; men = 2.27; port = 2.19
	9	**3418**	Trade = 4.97; ill = 4.79; aid = 4.67; pro = 4.05; Japan = 3.85;
	10	2317	Ton = 3.40; aid = 3.23; tonnes = 2.88; eat = 2.85; wheat = 2.63;

Fig. 4 Visualization of feature space

The main difference between FSAP and IAP is that the former adopts the FSL model, which employs the more stringent feature space learning strategy. On the contrary, IAP simply adds unlabeled samples into clustering without selection and feature space promotion. Although IAP can obtain similar or even better results at the beginning, because of the noise accumulation and semantic shift, it quickly falls behind FSAP and FSSAP.

3.4 Learned feature spaces

During the clustering procedures, the FSL model also spans new feature spaces and updates feature values. We picked out the feature spaces of FSCK-means on Reuter data to illustrate the transformed feature spaces.

From Table 6, it can be seen that the feature space has been changed after performing FSCK-means clustering. At the start-up, all the documents should be represented as a vector based on the whole vector space of the data. Because most of the documents do not contain all the words used in the data, the vector space model often falls into high-dimensional and sparse. For example, in our case it consists of 8277 dimensions. However, with the FSL model, we selected no more than 3500 features (the largest one is 3418 for Cluster 9) that could represent all these documents and clusters. The newly generated feature spaces are only 13.4–41.3% of the original one. Moreover, during the clustering, the values

of the features are also accurately refined. For example, the "trade" value in cluster 9 in Table 6 is 4.97. This cluster coincides with the "trade" class in Reuters. With these well-refined feature spaces, more accurate classifiers can be constructed easily by similarity or distance computing.

3.5 Feature spaces visualization

Different from the abstract concepts of representation learnning, we make a visualization example for the learned feature space to make it visible and easy understand. In Fig. 4, the learned feature spaces of FSCK-means on Reuter data have been depicted. Instead of building up the whole vector spaces of the data and the high-dimensional and sparse matrix computing, with the FSL model, a much smaller feature space is constructed by those few labeled samples and the refinement of unlabeled samples features. At last, the learned feature spaces with low dimension could represent all the documents and clusters. Moreover, during the clustering, the values of the features are also more accurately depicted. With the well-refined feature spaces, the word clouds of the learned feature spaces for each cluster are drawn. The larger a feature value is, the bigger its logo. In addition, more accurate classifiers based on these learned feature spaces could be easily constructed by similarity or distance computing.

4 Conclusion

In this paper, we proposed a feature space learning model and four FSL algorithms. Inspired by Zipf's law and words bursts, FSL model employs risk control strategies to avoid untrusted samples and filter the features. By constructing a more powerful feature space, the four clustering algorithms perform better than the classical clustering (MacQueen et al. 1967; Frey and Dueck 2007), semi-supervised clustering (Basu et al. 2002; Guan et al. 2011) and even incremental semi-supervised algorithms (Shi et al. 2009), e.g. on F-measure, FSCK-means is 73.6% higher than k-means and FSSAP gets 14.3% higher than IAP. Experimental results on the benchmark datasets demonstrate that the FSL model can dynamically promote learning performance and construct better understandable feature spaces. However, to pursue feature space and clusters simultaneously, the computation complexity of FSL model is higher than classical clustering and semi-supervised clustering models. Further model innovations may be needed to addressing this remaining limitation.

Acknowledgements The authors are grateful for the support of the National Natural Science Foundation of China (Nos. 61572228, 61472158, 61300147, 61602207), United States National Institutes of Health (NIH) Academic Research Enhancement Award (No. 1R15GM114739), the Science Technology Development Project from Jilin Province (No. 20160101247JC), Zhuhai Premier-Discipline Enhancement Scheme and Guangdong Premier Key-Discipline Enhancement Scheme.The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

References

Asuncion A, Newman D (2007) UCI machine learning repository [The data are available as Reuters-21578 Text Categorization Collection Data Set and Twenty Newsgroups Data Set]. https://archive.ics.uci.edu/ml/index.php. Accessed 8 May 2018

Barabasi AL (2005) The origin of bursts and heavy tails in human dynamics. Nature 435(7039):207–211. https://doi.org/10.1038/nature03459

Basu S, Banerjee A, Mooney RJ (2002) Semi-supervised clustering by seeding. In: Proceedings of the nineteenth international conference on machine learning, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, ICML '02, pp 27–34

Bekkerman R, El-Yaniv R, Tishby N, Winter Y (2003) Distributional word clusters vs. words for text categorization. J Mach Learn Res 3(Mar):1183–1208

Bengio Y, Courville A, Vincent P (2013) Representation learning: a review and new perspectives. IEEE Trans Pattern Anal Mach Intell 35(8):1798–1828. https://doi.org/10.1109/TPAMI.2013.50

Bobadilla J, Ortega F, Hernando A, Gutiérrez A (2013) Recommender systems survey. Knowl Based Syst 46:109–132. https://doi.org/10.1016/j.knosys.2013.03.012

Comaniciu D, Meer P (2002) Mean shift: a robust approach toward feature space analysis. IEEE Trans Pattern Anal Mach Intell 24(5):603–619. https://doi.org/10.1109/34.1000236

Estabrooks A, Jo T, Japkowicz N (2004) A multiple resampling method for learning from imbalanced data sets. Comput Intell 20(1):18–36

Esteva A, Kuprel B, Novoa RA, Ko J, Swetter SM, Blau HM, Thrun S (2017) Dermatologist-level classification of skin cancer with deep neural networks. Nature 542(7639):115–118

Frakes WB, Baeza-Yates R (1992) Information retrieval: data structures and algorithms, vol 331. Prentice Hall, Englewood Cliffs, New Jersey

Frey BJ, Dueck D (2007) Clustering by passing messages between data points. Science 315(5814):972–976. https://doi.org/10.1126/science.1136800

Griffiths TL, Steyvers M (2004) Finding scientific topics. Proc Natl Acad Sci 101(Supplement 1):5228–5235. https://doi.org/10.1073/pnas.0307752101

Guan R, Shi X, Marchese M, Yang C, Liang Y (2011) Text clustering with seeds affinity propagation. IEEE Trans Knowl Data Eng 23(4):627–637. https://doi.org/10.1109/TKDE.2010.144

Hinton GE, Salakhutdinov RR (2006) Reducing the dimensionality of data with neural networks. Science 313(5786):504–507

Huang X, Yin C, Dadras S, Cheng Y, Bai L (2018) Adaptive rapid defect identification in ECPT based on K-means and automatic

segmentation algorithm. J Ambient Intell Hum Comput. https://doi.org/10.1007/s12652-017-0671-5
Jain AK, Murty MN, Flynn PJ (1999) Data clustering: a review. ACM Comput Surv 31(3):264–323. https://doi.org/10.1145/331499.331504
Jiang M, Liang Y, Feng X, Fan X, Pei Z, Xue Y, Guan R (2016) Text classification based on deep belief network and softmax regression. Neural Comput Appl 29(1):61–70
Karpathy A, Johnson J, Fei-Fei L (2015) Visualizing and understanding recurrent networks. arXiv preprint arXiv:150602078. Accessed 8 May 2018
Kleinberg J (2002) Bursty and hierarchical structure in streams. ACM Press, p 91. https://doi.org/10.1145/775047.775061
Leichter I (2012) Mean shift trackers with cross-bin metrics. IEEE Trans Pattern Anal Mach Intell 34(4):695–706. https://doi.org/10.1109/TPAMI.2011.167
Lewis DD (2004) Reuters-21578 test collection, Reuters21578 [The data are available as Reuters-21578]. http://www.daviddlewis.com/resources/testcollections/reuters21578. Accessed 8 May 2018
Li Y, Zhou Z (2015) Towards making unlabeled data never hurt. IEEE Trans Pattern Anal Mach Intell 37(1):175–188. https://doi.org/10.1109/TPAMI.2014.2299812
MacQueen J et al (1967) Some methods for classification and analysis of multivariate observations. In: Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, Oakland, CA, USA, vol 1, pp 281–297
Rennie J (2008) Home page for 20 newsgroups data set [The data are available as 20 Newsgroups]. http://qwone.com/~jason/20Newsgroups/. Accessed 8 May 2018
Semetko H, Valkenburg P (2000) Framing European politics: a content analysis of press and television news. J Commun 50(2):93–109. https://doi.org/10.1111/j.1460-2466.2000.tb02843.x
Mn Serrano, Flammini A, Menczer F (2009) Modeling statistical properties of written text. PLoS One 4(4):e5372. https://doi.org/10.1371/journal.pone.0005372
Shi X, Guan, RC, Wang, LP, Pei, ZL, Liang, YC (2009) An incremental affinity propagation algorithm and its applications for text clustering. IEEE 2914–2919. https://doi.org/10.1109/IJCNN.2009.5178973
Tang W, Xiong H, Zhong S, Wu J (2007) Enhancing semi-supervised clustering: a feature projection perspective. In: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, pp 707–716
Wang Y, Chen S, Zhou Z (2012) New semi-supervised classification method based on modified cluster assumption. IEEE Trans Neural Netw Learn Syst 23(5):689–702. https://doi.org/10.1109/TNNLS.2012.2186825
Wu M, Li X, Liu C, Liu M, Zhao N, Wang J, Wan X, Rao Z, Zhu L (2018) Robust global motion estimation for video security based on improved k-means clustering. J Ambient Intell Humaniz Comput. https://doi.org/10.1007/s12652-017-0660-8
Xue H, Chen S, Yang Q (2011) Structural regularized support vector machine: a framework for structural large margin classifier. IEEE Trans Neural Netw 22(4):573–587. https://doi.org/10.1109/TNN.2011.2108315
Zhang XY, Yin F, Zhang YM, Liu CL, Bengio Y (2018) Drawing and recognizing Chinese characters with recurrent neural network. IEEE Trans Pattern Anal Mach Intell 40(4):849–862
Zipf GK (1949) Human behavior and the principle of least effort, vol xi. Addison-Wesley Press, Oxford