Extended Asynchronous SN P Systems for Solving Sentiment Clustering of Customer Reviews in E-commerce Websites

XUE BAI
Shandong Normal University
School of Management Science and Engineering
88 East Wenhua Road, 250014 Jinan
CHINA
sdnubaixue@163.com

XIYU LIU
Shandong Normal University
School of Management Science and Engineering
88 East Wenhua Road, 250014 Jinan
CHINA
sdxyliu@163.com

Abstract: Customer reviews of goods sold on shopping sites are pivotal factors to affect the potential purchasing behavior. However, some traditional classification methods are too simplistic to take buyers' feelings and emotions into full consideration. In this paper, an extended asynchronous spiking neural P system with local synchronization and weighted synapses is proposed to achieve the sentiment clustering of Chinese customer reviews based on graph representation. Each sentiment word in a comment record is regarded as one unique node of a senti-graph, and the extended SN P system carries out the clustering algorithm by taking each senti-graph as a vertex in the 2D scheme and using the sentiment similarity between them to measure the edge weight. Neurons in the system are divided into four main sets and several subsets, with rules applied asynchronously among different sets but used in a synchronous manner within the same one. The computational complexity is limited to $O(n^2)$ in the worst case and optimized to $O(n)$ as the best. A case study shows its computing effectiveness and customer feedbacks clustered by emotion orientation could provide us better understanding on the customer feelings and product features.

Key–Words: Membrane computing, Spiking neural P system, Sentiment clustering, Graph representation, Customer review

1 Introduction

With the popularity of Internet and the convenience of online shopping, more and more transactions on commodities and services have been conducted through the E-commerce website. Appraisals given by clients after once consumption have become both fundamental ways for business companies to collect attitudes about their products and important references for potential customers who intend to put their preferred goods in the "shopping cart". As the main task of opinion mining, sentiment analysis aims to abstract the emotion orientation of documents and sentences, and provide more complex classifications rather than just positive or negative categories. In the sentiment clustering approach based on the graph representation, a text is treated as a digraph, with the node denoting the opinion word and the directed edge defining the connecting relation of two words.

Membrane computing is one of the recent branches of natural computing and quite a number of topics have been raised and studied the new proposed P system. As the main type of tissue-like P systems, spiking neural P systems (SN P systems, for short) were introduced in the aim of defining computing models inspired from the way neurons communicating with each other by means of electrical impulses, where there are synapses between each pair of connected neurons [1]. With only one kind of objects indicating the spike, specific rules are applied non-deterministically in each neuron and the whole system works with maximum parallelism.

In this paper, a variant of SN P systems is introduced, using an asynchronous mode at the global level and making local synchronization in subdivided sets. Synapses are endowed with integer weight and different thresholds of spikes are assigned on different neurons. The problem of classifying those online reviews should be transformed into the clustering of senti-graphs in the first place, and then the PAM algorithm based on this new representation is applied to divide these data elements into several sentiment clusters. A typical real-word case is used to elaborate the applicability and scalability of this P system. Some drawbacks waiting to be conquered are also discussed in our work.

2 Related Work

In general SN P systems, the synchronization plays a crucial role in controlling the computation, but Pan et al. [2, 3] considered non-synchronized systems as noteworthy with the fact that an enabled rule wasn't obligatorily used and the neuron could remain unfired in a given time unit. Since there can be several synapses between each pair of connected neurons, Wang et al. [1, 4] endowed synapses with integer weight to be the efficacy measure. Each neuron was designated a threshold denoted by a real number and the spikes would be sent out when its potential was equal to the threshold. Moreover, characteristics of the astrocyte control and neuron division have been introduced into the framework of SN P systems successively [5, 6], but few researches have focused on the combination of clustering algorithms and SN P systems.

Feng et al. [7, 8] grouped blog search results with a lexicon-based sentiment clustering algorithm instead of using the traditional topic-oriented technique, and used a Probabilistic Latent Semantic Analysis (PLSA) based approach to model the hidden sentiment factors. In [9], a SoB-graph model was adopted and the node was defined as a sentiment word marked with an identifier. Each edge was assigned with three tags and the sentiment similarity between two BSR-items was calculated by the distance between two SoB-graphs. The sentiment analysis of customer reviews has already been mentioned in [10, 11]. However, researchers should pay more attention to the client feedbacks on Chinese E-commerce websites, which is a huge emerging market with rapid expansion, and clustering algorithms associated with the graph representation are capable of achieving opinion mining on large data sets generated online.

3 Sentiment Clustering Based on Graph Representation

In the majority of Chinese shopping websites, customer reviews of goods and services are usually described by three words: positive, neutral, and negative, which represent three recommendation degrees given by users based on their personal experiences. Nevertheless, this partition method is ambiguous to show different attitudes of people in regards to the quality of products, the logistics distribution, and the after-sales service. In decision making, especially on cosmetics and clothes, others' opinions have a significant effect on the choice making of customers, which need to be subdivided to improve users' browsing efficiency and assist manufacturers to find their weaknesses compared with other competitors. In addition, clear classification of shopping appraisals based on emotion orientation will help to collect more client information and contribute to the customer relationship management (CRM) in the long run.

3.1 The graph-based framework of customer reviews

For the purpose of applying sentiment clustering to customer reviews associated with a variety of commodities, the first thing to do is treating every record as an independent entity, using Chinese Natural Language Processing Tool (ICTCLAS) to conduct segmentation on each sentence and extracting sentiment words by means of NTUSD, a kind of simplified Chinese sentiment lexicons constituted by 2810 positive words and 8276 negative words. Since each review may consist of several parts of speech words such as nouns, verbs, adjectives and so forth, we just confine ourselves to retaining the words with sentiment polarity and neglecting others. Given the extracted sentiment words of one comment record, we use a two-level graph representation to provide more structural information within each record and more mutual relationships between them.

Definition 1 A graph G is a weighted and undirected graph of the form $G = (V, E)$, $V = \{v_i | 1 \leq i \leq n\}$, and $E = V \times V = \{e_{ij} | 1 \leq i, j \leq n\}$. Each customer review is corresponded to a vertex v_i and the weight endowed on the edge denotes the sentiment similarity between two customer reviews.

Definition 2 A customer review is also modeled as a *senti-graph* (using SG for short), and each $SG_i (1 \leq i \leq n)$ is seen as a vertex in graph G. SG_i is an undirected complete graph with the following structure: $SG_i = \{V_i, E_i, \lambda, \varphi\}$ $(1 \leq i \leq n)$, $V_i = \{v_{ij}^p | 1 \leq i \leq n, 1 \leq j \leq n_i, p \in N\}$, and $E_i = V_i \times V_i = \{e_{jk}^q | 1 \leq j,\ k \leq n_i,\ q \in N\}$, where $\lambda = \{p | p \in N\}$ and $\varphi = \{q | q \in N\}$.

From the above definition, each node of a senti-graph represents a sentiment word in the customer review and n_i is the number of nodes remaining in SG_i. Therefore, v_{ij} indicates the jth node in the ith senti-graph. Note that a sentiment word could appear more than once in a review record, it's appropriate to mark them with different vertices with the fact that they have different locations and share the relationship with different neighbors. So $p \in N$ is used as a label to indicate the word frequency and $v_{ij}^1, v_{ij}^2, \ldots, v_{ij}^p$ means different vertices but the same sentiment word. Moreover, label p can be omitted if there are no repeated words in a record. Label $q (1 \leq q \leq n_i - 1)$ reflects the amount of words between two sentiment words and it equals to 1 when they are adjacent to each other. The structure of graph G and an example of a senti-graph is better seen in Figure (1) and the sentiment similarity between two customer reviews is measured by the closeness between two senti-graphs based on this new graph representation.

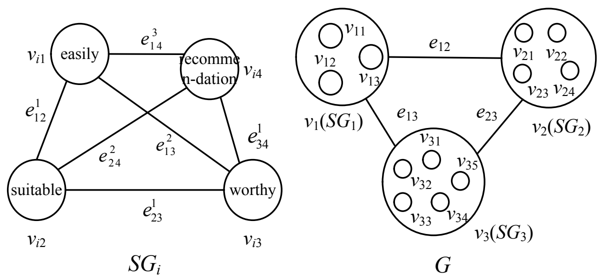

Figure 1: Example of a senti-graph and graph G

3.2 Clustering customer reviews using graph similarity

The similarity between two senti-graphs, defined as $s(SG_i, SG_j)$, is calculated by the following equation:

$$s(SG_i, SG_j) = \frac{mss(SG_i, SG_j)}{|SG_i| + |SG_j| + mss(SG_i, SG_j)} \quad (1 \le i < j \le n) \qquad (1)$$

where $mss(SG_i, SG_j)$ characterizes the maximum similarity between SG_i and SG_j. More precisely, $mss(SG_i, SG_j) = |identical\ nodes| + |identical\ edges|$, namely the sum of the number of all identical nodes and identical edges in two graphs. $|SG_i|$ measures the size of graph SG_i and equals to the equation $|SG_i| = |V_i| + |E_i|$. Obviously, $s(SG_i, SG_j) = s(SG_j, SG_i)$ on condition that $1 \le i < j \le n$, and we presume that $s(SG_i, SG_j) = 1$ when i is equivalent to j. Note that $s(SG_i, SG_j)$ changes in the interval of $[0, 1]$ and it reduces to zero with the value of $mss(SG_i, SG_j)$ becomes zero. With the same $|SG_i| + |SG_j|$, $s(SG_i, SG_j)$ gets larger when the maximum similarity of two senti-graphs increases and $s(SG_i, SG_j)$ turns to decline with the condition that the value of $|SG_i| + |SG_j|$ begins to rise and $mss(SG_i, SG_j)$ remains unchanged.

Definition 3 Two nodes in senti-graphs SG_i and SG_j $(1 \le i < j \le n)$ are *identical nodes* if the similarity between the two corresponding sentiment words is larger than the threshold τ.

Draw inspiration from [12], some external knowledge is utilized to measure the sentiment similarity between two words with polarity. Considering the word resemblance, the value of τ in our paper is limited to 0.1, and two nodes are identical if and only if $s(v_{im}, v_{jl}) > 0.1$ (with the restriction that $1 \le i < j \le n$, $1 \le m \le n_i$, $1 \le l \le n_j$). Under the previous definition, $s(v_{im}, v_{jl})$ equals to 1 when v_{im} and v_{jl} match with the same sentiment word. Unlike traditional mathematical symbols, v_{im} in SG_i could be "identical" with multiple nodes in SG_j as long as the similarity between two sentiment words exceeds the threshold.

Definition 4 The edge $e^k_{ml} (1 \le m < l \le n_i, k \in N)$ and $e^t_{sr} (1 \le s < r \le n_j, t \in N)$ in SG_i and SG_j $(1 \le i < j \le n)$ are *identical edges* if one of the following conditions holds true: 1. v_{im} and v_{js}, v_{il} and v_{jr} are *identical nodes* with $k = t$; 2. v_{im} and v_{jr}, v_{il} and v_{js} are *identical nodes* with $k = t$. A sigleton edge in SG_i can also be "identical" with several edges in SG_j simultaneously and all the times should be counted to reveal the close connection between two senti-graphs.

Derived from the previous definition, we use the PAM clustering algorithm, a k-medoids based partitioning technique, to realise the sentiment classification of online customer reviews, and the basic algorithm is described as follow [13]:

Table 1: Sentiment clustering of customer reviews using the senti-graph representation

Inputs:	a set of n customer reviews, a parameter k defining the number of clusters
Outputs:	k centroids of clusters, and for each review the cluster it belongs to
Step 1	Segment review records and extract sentiment words by eliminating the others;
Step 2	Map each record into a senti-graph, calculate the similarity between each pair of them;
Step 3	Use n vertices in graph G to denote senti-graphs and choose $k (2 \le k \le n-1)$ of them as the initial centroids arbitrarily;
Step 4	Assign each data object to be in the cluster of its most similar centroid;
Step 5	Select a non-medoid object O' randomly and compute the total cost S';
Step 6	Replace one initial centroid with O' if $S' < S$ (the new centroid improves the total cost of the clustering result);
Step 7	Repeat Steps 4 to step 6 until the centroids do not change.

In light of this sentiment clustering algorithm, it pertains to designate those sentiment words, which contain in the center senti-graph, as the key words of this cluster to represent the overall appraisal of this product or service. Customers could see what other users think about the goods and view its strengths and weaknesses more intuitively.

4 Basic Configuration of Extended Asynchronous SN P Systems

Note that it's useful for readers to have some familiarity with basic prerequisites of the language theory. With respect to the alphabet V, V^* is the free monoid generated by V with the concatenation operation and V^+ denotes all nonempty strings over V^* expect λ, the empty string. Likewise, N^+ represents the natural numbers and $N = N_+ \cup \{0\}$. As SN P systems need only one kind of objects, we restrict V in $\{a\}$ and simply use a^*, a^+ instead of $\{a\}^*$ and $\{a\}^+$. As is mentioned in [14], an expression E over V is constructed starting from λ and symbols of V, using union, concatenation, and Kleene + as useful operations and needing parentheses for specifying the order. More precisely, each expression is associated with a language $L(E)$ and complied with the following representation: 1. $L(\lambda) = \{\lambda\}$ and $L(a) = \{a\}$ $(a \in V)$, 2. $L((E_1)^+) = L(E_1)^+$, 3. $L((E_1)\cup(E_2)) = L(E_1)\cup L(E_2)$, and 4. $L((E_1)(E_2)) = L(E_1)L(E_2)$.

Given the above notions, an extended asynchronous SN P system with local synchronization and weighted synapses (in short, an EASN P system) to achieve the sentiment clustering algorithm is a construct of the form:

$$\Pi = (O, \delta, \sigma, syn, \omega, in, out, H)$$

where:

- $O = \{a\}$ is an alphabet made up of only one object a, called a spike.
- $\delta = \{\theta_i|1 \leq i \leq n\} \cup \{\alpha_i|1 \leq i \leq n^2\} \cup \{\beta_i|1 \leq i \leq n^2\} \cup \{\gamma_i|1 \leq i \leq n^2\} \cup \{\mu_i|1 \leq i \leq 3\}$ is the family of neurons, with the meaning that θ_i $(1 \leq i \leq n)$ denote the n vertices in graph G, and α_i, β_i, γ_i $(1 \leq i \leq n^2)$ are assigned to indicate the edge weight $w_{11}, w_{21}, \ldots, w_{n1}, w_{12}, w_{22}, \ldots, w_{n2}, \ldots, w_{1n}, \ldots, w_{nn}$ sequentially. For pictorial illustration, a simplified model suiting for the clustering with four vertices is shown in Figure (2), which can be viewed as a toy example to figure out how neurons and synapses function as a whole system.

  Moreover, each neuron has the features of the form $\delta_i = (m_i, R_i, P_i)$ $(1 \leq i \leq 3n^2 + n + 3)$, where $m_i \geq 0$ shows the initial number of spikes contained in δ_i, and R_i is a finite set of spiking rules described as $(E/a^c \to a^d)$ or $(E'/a^c \to \lambda)$. Note that E and E' are regular expressions over $\{a\}$ and $L(E) \cap L(E') = \emptyset$. The rule R_i will be triggered only when E or E' is strictly satisfied and $c(1 \leq c \leq m_i)$ spikes are consumed by emitting $d(1 \leq d \leq c)$ spikes to each target neuron such that $(\delta_i, \delta_j) \in syn$. E can be omitted if $E = \{a^c\}$. $P_i \geq 0$ is designated as the potential threshold of δ_i to affect the available spikes in each step. The remaining spikes will vanish when the amount of them is smaller than P_i and the potential of δ_i turns to zero with no more reductions occur.
- $\sigma = \{\sigma_1, \sigma_2, \sigma_3, \sigma_4\}$ is the family of local sets constructed to achieve different procedures in the sentiment clustering algorithm. $\sigma_1 = \{\theta_i|1 \leq i \leq n\}$ constitutes the *Generation Module* and σ_1 is divided into n subsets with each neuron serving as one subset. $\sigma_2 = \{\alpha_i, \beta_i|1 \leq i \leq n^2\}$ is called the *Comparison Module* without further partition. The *Selection Module* equivalent to σ_3 consists of $\gamma_i(1 \leq i \leq n^2)$ and k of them (denoted by $\{\gamma_i|i = (j-1)n + j, 1 \leq j \leq n\}$) are combined into one subset with the remaining $n(n-1)$ neurons seen as a singleton subset respectively. The last *Determination Module* contains $\mu_i(1 \leq i \leq 3)$ and no more subdivisions are conducted on them.

  Spiking rules in different sets and subsets are applied asynchronously and the rules in the same set without subdivision are used in a synchronous manner. From the view of the whole system, there is only one set of rules can be executed in each time unit, and neurons in other sets have the chance to fire only when no enable rules in this set can be used. For subsets in each module, assume that at a given moment, one subset is picked up randomly and each neuron which can apply a rule should do it. After fully implementation of the spike firing, another subset is non-deterministically chosen to be active and the rules in it are able to be used.
- syn is a synapse dictionary among cells with the following components:

  $syn = \{(\theta_i, \theta_j)|1 \leq i, j \leq n, i \neq j\}$
  $\cup \{(\theta_i, \alpha_{ni}), (\theta_i, \alpha_{ni-1}), \ldots, (\theta_i, \alpha_{(i-1)n+2}),$
  $(\theta_i, \alpha_{(i-1)n+1})|1 \leq i \leq n\} \cup \{(\alpha_i, \beta_i)|1 \leq i \leq n^2\}$
  $\cup \{\{(\beta_i, \alpha_{i-jn}), (\beta_i, \alpha_{n+i-jn}), \ldots,$
  $(\beta_i, \alpha_{n(n-1)+i-jn})|0 \leq j \leq n-1, i = jn+1,$
  $jn+2, \ldots, jn+n\} - \{(\beta_i, \alpha_i)|1 \leq i \leq n^2\}\}$
  $\cup \{\{(\beta_i, \beta_{i-jn}), (\beta_i, \beta_{n+i-jn}), \ldots,$
  $(\beta_i, \beta_{n(n-1)+i-jn})|0 \leq j \leq n-1, i = jn+1,$
  $jn+2, \ldots, jn+n\} - \{(\beta_i, \beta_i)|1 \leq i \leq n^2\}\}$
  $\cup \{(\beta_i, \gamma_i)|1 \leq i \leq n^2\} \cup \{(\gamma_i, H)|1 \leq i \leq n^2\}$
  $\cup \{(\gamma_i, \mu_1)|1 \leq i \leq n^2\} \cup \{\{(\gamma_i, \gamma_{i-jn}),$
  $(\gamma_i, \gamma_{n+i-jn}), \ldots, (\gamma_i, \gamma_{n(n-1)+i-jn})|0 \leq j \leq$
  $n-1, i = jn+1, jn+2, \ldots, jn+n\} -$
  $\{(\gamma_i, \gamma_i)|1 \leq i \leq n^2\} - \{(\gamma_{(i-1)n+i}, \gamma_{(j-1)n+i})|$
  $1 \leq i, j \leq n\} + \{(\gamma_{(i-1)n+i}, \gamma_{(i-1)n+i})|1 \leq i \leq$
  $n\}\} \cup (\mu_1, \mu_3) \cup (\mu_2, \mu_3) \cup (H, \mu_2) \cup \{(\mu_3, \theta_i)|$
  $1 \leq i \leq n\}$.
- ω reveals the weight on synapses derived from the

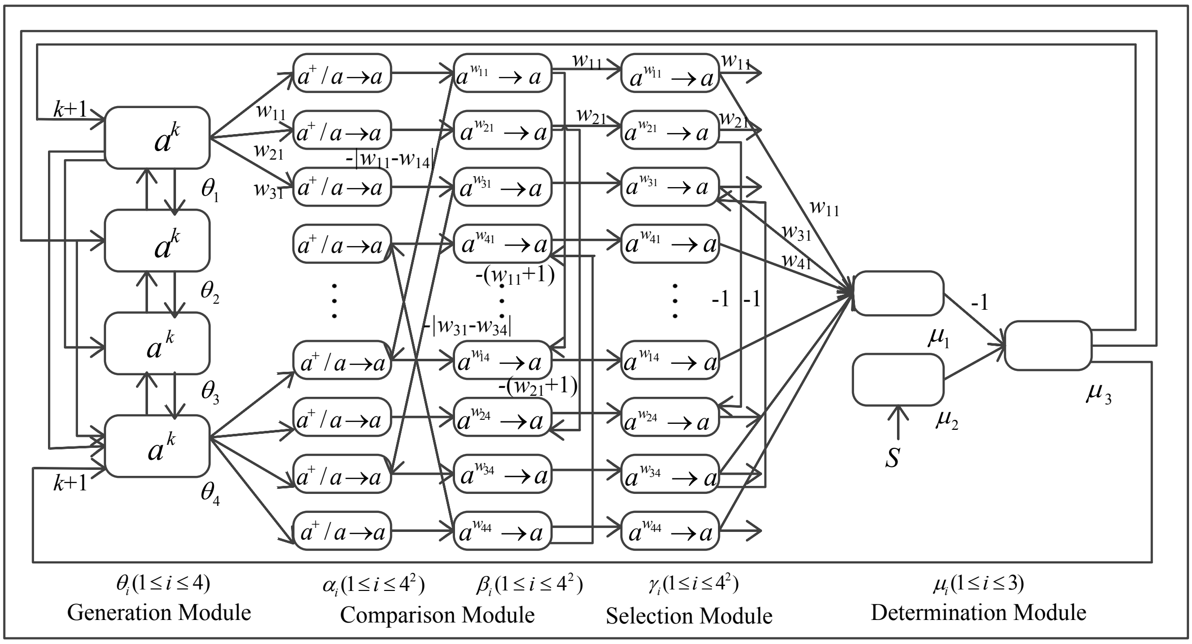

Figure 2: A simplified EASNP system for solving sentiment clustering

edge weights in graph G and the number of clusters. As mentioned in section 3.2, $s(SG_i, SG_j)$ is used to express the closeness between two review records and it's corresponding to w_{ij} endowed on e_{ij}. Nevertheless, it pertains to put $s(SG_i, SG_j)$ $(1 \le i \le j \le n)$ in descending order and select the sequence number to denote w_{ij} in order to be compatible with our system. After the transformation, it becomes apparent that: 1. $s(SG_i, SG_i) = w_{ii} = 1 (1 \le i \le n)$, 2. $w_{ij} = w_{ji} (1 \le i < j \le n)$, 3. $2 \le w_{ij} \le n(n-1)/2 + 1 (1 \le i, j \le n, i \ne j)$, and 4. $w_{ij} = w_{ml}$ if $s(SG_i, SG_j) = s(SG_m, SG_l)$ $(1 \le i, j, m, l \le n)$. The relationship between synapses and their weights are listed as below and the weight assigned by the value of 1 is omitted for simplification:

$$\omega = \begin{cases} w_{1i} & \{(\theta_i, \alpha_{(i-1)n+1}) | 1 \le i \le n\}, \\ & \{(\beta_{(i-1)n+1}, \gamma_{(i-1)n+1}) | 1 \le i \le n\}, \\ & \{(\gamma_{(i-1)n+1}, H) | 1 \le i \le n\}, \\ & \{(\gamma_{(i-1)n+1}, \mu_1) | 1 \le i \le n\} \in syn \\ w_{2i} & \{(\theta_i, \alpha_{(i-1)n+2}) | 1 \le i \le n\}, \\ & \{(\beta_{(i-1)n+2}, \gamma_{(i-1)n+2}) | 1 \le i \le n\}, \\ & \{(\gamma_{(i-1)n+2}, H) | 1 \le i \le n\}, \\ & \{(\gamma_{(i-1)n+2}, \mu_1) | 1 \le i \le n\} \in syn \\ & \quad \cdots\cdots \\ w_{ni} & \{(\theta_i, \alpha_{in}) | 1 \le i \le n\}, \\ & \{(\beta_{in}, \gamma_{in}) | 1 \le i \le n\}, \end{cases}$$

to be continued ...

$$\omega = \begin{cases} & \{(\gamma_{in}, H) | 1 \le i \le n\}, \\ & \{(\gamma_{in}, \mu_1) | 1 \le i \le n\} \in syn \\ k+1 & \{(\mu_3, \theta_i) | 1 \le i \le n\} \in syn \\ -1 & (\mu_1, \mu_3), \{\{(\gamma_i, \gamma_{i-jn}), (\gamma_i, \gamma_{n+i-jn}) \\ & , \ldots, (\gamma_i, \gamma_{n(n-1)+i-jn}) | 0 \le j \le n \\ & -1, i = jn+1, \ldots, jn+n\} - \\ & \{(\gamma_i, \gamma_i) | 1 \le i \le n^2\} - \{(\gamma_{(i-1)n+i}, \\ & \gamma_{(j-1)n+i}) | 1 \le i, j \le n\} + \\ & \{(\gamma_{(i-1)n+i}, \gamma_{(i-1)n+i}) | 1 \le i \le n\}\} \\ & \in syn \\ -(w_{1i}+1) & \{\{(\beta_{(i-1)n+1}, \beta_{jn+1}) | 1 \le i \le n, \\ & 0 \le j \le n-1\} - \{(\beta_{(i-1)n+1}, \\ & \beta_{(i-1)n+1}) | 1 \le i \le n\}\} \in syn \\ -(w_{2i}+1) & \{\{(\beta_{(i-1)n+2}, \beta_{jn+2}) | 1 \le i \le n, \\ & 0 \le j \le n-1\} - \{(\beta_{(i-1)n+2}, \\ & \beta_{(i-1)n+2}) | 1 \le i \le n\}\} \in syn \\ & \quad \cdots\cdots \\ -(w_{ni}+1) & \{\{(\beta_{in}, \beta_{jn+n}) | 1 \le i \le n, 0 \le \\ & j \le n-1\} - \{(\beta_{in}, \beta_{in}) | 1 \le i \\ & \le n\}\} \in syn \end{cases}$$

to be continued ...

$$\omega = \begin{cases} -|w_{1i} - w_{1j}| \ \{\{(\beta_{(i-1)n+1}, \alpha_{(j-1)n+1})|1 \le i, \\ \qquad j \le n\} - \{(\beta_{(i-1)n+1}, \\ \qquad \alpha_{(i-1)n+1})|1 \le i \le n\}\} \in syn \\ -|w_{2i} - w_{2j}| \ \{\{(\beta_{(i-1)n+2}, \alpha_{(j-1)n+2})|1 \le i, \\ \qquad j \le n\} - \{(\beta_{(i-1)n+2}, \\ \qquad \alpha_{(i-1)n+2})|1 \le i \le n\}\} \in syn \\ \qquad \cdots\cdots \\ -|w_{ni} - w_{nj}| \ \{\{(\beta_{in}, \alpha_{jn})|1 \le i, j \le n\} - \\ \qquad \{(\beta_{in}, \alpha_{in})|1 \le i \le n\}\} \in syn \end{cases}$$

- in explains the initial configuration by designating m_i spikes to θ_i $(1 \le i \le n)$ and S spikes to the environment. Note that m_i spikes included in θ_i help to achieve the third step in the sentiment clustering algorithm and the preparation of S spikes contributes to completing the computing process in step 6.
- out shows output neurons consisting of γ_i $(1 \le i \le n^2)$ and μ_3, where the production of γ_i shows the computational result during each iteration and the spiking of μ_3 initializes the next round of clustering procedures.
- H indicates the environment and it's capable of receiving spikes sent from γ_i $(1 \le i \le n^2)$ and pushing them into μ_1 to carry out the comparison between the total cost S and S', which is generated in step 3 and 5 by different assignments of the cluster centroids.

Based on the previous illustration, some additional explanations of EASNP systems are required to list as below:

- As usual, if the rule $(E/a^c \to a^d)$ in δ_i is applicable in step t and $(\delta_i, \delta_j) \in syn$ $(1 \le i, j \le n, i \ne j)$, then a spike is sent out and replicated by the weight endowed on (δ_i, δ_j), entering into δ_j with no delay. In the case of the output neuron, spikes are also delivered to the environment immediately. Similarly, it happens in the same step that the rule $(E'/a^c \to \lambda)$ is used and c spikes are lost.
- Since only one (sub)set becomes active in each time unit, rules in a neuron are applied in the exhaustive mode: If δ_i produces a spike in step t, all of its available rules should be used in subsequent steps and all the reserving spikes need to be evolved as much as possible. With one of those rules selected in a nondeterministic way during each computation, the process of neuron updating stops when no applicable rules exist in it.
- The configuration of this system, regarded as C_i $(i = 1, 2, \dots)$, is described by the amount of spikes present in each neuron and the state of neurons which are active or not. A transition of Π is defined as $C_i \Rightarrow C_{i+1}$ and any sequence of them starting from C_1 is called a computation of Π. A computation is successful when it halts and no rules can be activated in any neuron. In the given case, the computation finishes when $S' \ge S$ — the total cost isn't improved after once circulation, and those spikes indicating the corresponding edge weights, fired by γ_i $(1 \le i \le n^2)$ in the last iteration, are viewed as the final clustering result generated by the EASN P system.
- The total cost S has the expression: $S = w_{1,i_1} + w_{2,i_2} +, \dots, +w_{n,i_n}$ $(1 \le i_j, j \le n)$, where k of w_{j,i_j} equal to w_{ii} $(1 \le i \le n)$ associated with the k centroids, and $(n-k)$ of w_{j,i_j} denote the weight between non-medoid object and its centroid in each cluster. It's easy to see that $S_{min} = k + 2(n-k) = 2n - k$, and all the weights between different vertices are set to two. S_{max} can be calculated using equation (2) with the fact that S achieves its maximum if and only if the remaining $(n-k)$ weights get the value of $n(n-1)/2$, $n(n-1)/2 - 2, \dots, n(n-1)/2 - 2(n-k-1)$ sequentially. Therefore, S changes in the interval of $[2n-k, (n^3-(3+k)n^2+5kn+2n-2k^2)/2]$ and the difference between S_{max} and S_{min} varies in $[0, (n^3-(3+k)n^2+5kn-2n-2k^2+2k)/2]$.

$$\begin{aligned} S_{max} &= k + n(n-1)/2 + n(n-1)/2 - 2 + \dots \\ &\quad + n(n-1)/2 - 2(n-k-1) \\ &= (n^3 - (3+k)n^2 + 5kn + 2n - 2k^2)/2 \end{aligned} \quad (2)$$

- In the first iteration of the algorithm, S_{max} spikes are prepared in the environment. When neuron μ_1 receives S spikes, S_{max} spikes are delivered to μ_2 simultaneously to carry out the comparison between S and S_{max}. If $S < S_{max}$, the computing process continues and S spikes will reach μ_2 in the second round to make the *Determination Module* start to work.

5 EASN P Systems for Solving Sentiment Clustering Problem

In this section, we will give some elaborations of the EASN P system to realize the sentiment clustering algorithm efficiently. For each $n \in N$, Table (2) indicates the threshold setting of δ_i $(1 \le i \le 3n^2+n+3)$, as well as the number of initial spikes and rules contained in them.

An overview of the computation needs to be introduced firstly. In order to finish each iteration of

Table 2: Threshold setting and the illustration of initial spikes and rules

Module	Generation Module	Comparison Module	Selection Module	Decision Module	
Neuron	θ_i $(1 \leq i \leq n)$	$\alpha_i \quad \beta_i$ $(1 \leq i \leq n^2)$	γ_i $(1 \leq i \leq n^2)$	μ_i $(1 \leq i \leq 2)$	μ_i $(i = 3)$
Threshold (P_i)	$P_i = k$ $(1 \leq i \leq n)$	$P_i = 1$ $(1 \leq i \leq n^2)$	$P_i = w_{i,1}\ (1 \leq i \leq n)$ $P_i = w_{i-n,2}\ (n+1 \leq i \leq 2n)$ $\ldots$ $P_i = w_{i-n^2+n,n}$ $(n^2 - n + 1 \leq i \leq n^2)$	$P_i = 1$ $(1 \leq i \leq 2)$	$P_3 = 1$
Initial Spike	k	0	0	S_{max}	0
Rule	$a^k \to a$ $a^{k+1} \to a$ $\ldots$ $a^{2k-1} \to a$ $a^{2k+1}/a \to \lambda$ $a^{4k}/a^{3k} \to \lambda$ a^{4k+1}/a^{2k+1} $\to \lambda$	$a^+/a \to a$ $a^{w_{i,1}} \to a$ $(1 \leq i \leq n)$ $a^{w_{i-n,2}} \to a$ $(n+1 \leq i \leq 2n)$ $\ldots$ $a^{w_{i-n^2+n,n}} \to a$ $(n^2 - n + 1 \leq i \leq n^2)$	$a^{w_{i,1}} \to a$ $(1 \leq i \leq n)$ $a^{w_{i-n,2}} \to a$ $(n+1 \leq i \leq 2n)$ $\ldots$ $a^{w_{i-n^2+n,n}} \to a$ $(n^2 - n + 1 \leq i \leq n^2)$	$a^{2n-k} \to a^{2n-k}$ $a^{2n-k+1} \to a^{2n-k+1}$ $a^{2n-k+2} \to a^{2n-k+2}$ $\ldots$ $a^{S_{max-1}} \to$ $a^{S_{max-1}}$ $a^{S_{max}} \to a^{S_{max}}$	$a \to a$ $a^2 \to a$ $a^3 \to a$ $\ldots$ $a^{S_{max} - S_{min}}$ $\to a$

the algorithm, the strategy consists of four stages and each of them corresponds to one module in the framework of EASN P systems. In the *Generation Stage*, θ_i $(1 \leq i \leq n)$ pick up k vertices v_{i_1}, v_{i_2}, $\ldots$, v_{i_k} $(1 \leq i_j \leq n, 1 \leq j \leq k)$ randomly to be as the initial centroids in the first iteration. In the next round, one of $(n-k)$ non-medoid objects is selected and defined as O' to replace a former center O. As to the *Comparison Stage*, w_{i,i_1}, w_{i,i_2}, $\ldots$, w_{i,i_k} are compared to each other and v_i will be designated to its most similar centroid with the minimum weight. If v_i owns the same minimum value with muti-medoids, then the *Selection Module* intends to choose one randomly. In the last stage, S_{max} will be replaced by S and the algorithm proceeds to the second circulation if $S < S_{max}$. When $S' \geq S$ in the later computation, the system halts and final clustering results come into being eventually. More details about the calculation process are listed as below:

Generation Stage: Since θ_i $(1 \leq i \leq n)$ is modeled as a singleton subset and it has the enable rule $(a^k \to a)$ in step 1, one neuron θ_{i_1} $(1 \leq i_1 \leq n)$ becomes active due to the asynchronous manner, sending a spike to θ_{i_2}, θ_{i_3}, $\ldots$, θ_{i_n} and $\alpha_{(i_1-1)n+1}$, $\alpha_{(i_1-1)n+2}, \ldots, \alpha_{i_1 n}$. After that, θ_{i_1} has no spikes left and the other neurons contained in this module own $(k+1)$ spikes with the executable rule $(a^{k+1} \to a)$. Besides that, $\alpha_{(i_1-1)n+1}$, $\ldots$, $\alpha_{i_1 n}$ receive w_{1,i_1}, $\ldots$, w_{n,i_1} spikes separately at the same time. In step 2, θ_{i_2} emits a spike to θ_{i_1}, θ_{i_3}, $\ldots$, θ_{i_n} and $\alpha_{(i_2-1)n+1}$, $\alpha_{(i_2-1)n+2}$, $\ldots$, $\alpha_{i_2 n}$, causing θ_{i_3}, $\ldots$, θ_{i_n} to get $(k+2)$ spikes but making θ_{i_1} reserve no spikes as the number of them is less than the potential threshold k $(2 \leq k \leq n-1)$. With the fact that only one module is allowed to work in each time unit, $\alpha_{(i_1-1)n+1}$, $\ldots$, $\alpha_{i_1 n}$ can't start to fire although the firing condition $(a^+/a \to a)$ is satisfied. From step 3 to k, θ_{i_3} to θ_{i_k} are non-deterministically chosen with only one at a time, making $\theta_{i_{k+1}}$ to θ_{i_n} retain $2k$ spikes and the selected k neurons have the potential of zero. The *Generation Module* stops in step k because no more rules can be further used and kn neurons of α_i $(1 \leq i \leq n^2)$, connected with θ_{i_1} to θ_{i_k} respectively, own the amount of spikes equal to the synapse weights and will be triggered in the next stage.

Comparison Stage: The first layer of this module consists of α_i $(1 \leq i \leq n^2)$ and kn of them begin to produce a spike simultaneously since step $(k+1)$ as all of them are included in one set without subdivision. In step $(k+1)$, kn neurons of β_i $(1 \leq i \leq n^2)$ located in the second layer receive a spike and $\beta_{(i_1-1)n+i_1}$, $\beta_{(i_2-1)n+i_2}$, $\ldots$, $\beta_{(i_k-1)n+i_k}$ are ready to fire in the next step. That is to say, the minimum weight of k centroids has been gained due to the fact that a centroid can't be assigned to other clusters besides the one it belongs to. In step $(k+2)$, $\alpha_{(i_1-1)n+i_1}$ to $\alpha_{(i_k-1)n+i_k}$ have no spikes to fire but the other $k(n-1)$ neurons in the first layer continue to execute the rule $(a^+/a \to a)$. At the same time, $\beta_{(i_1-1)n+i_1}$ send one spike to α_{i_1}, α_{n+i_1}, $\ldots$, $\alpha_{n(n-1)+i_1}$ (expect $\alpha_{(i_1-1)n+i_1}$) and β_{i_1}, β_{n+i_1}, $\ldots$, $\beta_{n(n-1)+i_1}$ (expect $\beta_{(i_1-1)n+i_1}$), actually generating $|w_{i_1,i_1} - w_{i_1,1}|$, $\ldots$, $|w_{i_1,i_1} - w_{i_1,n}|$ (expect $|w_{i_1,i_1} - w_{i_1,i_1}|$) and $|w_{i_1,i_1} + 1|$ spikes to be removed from the target neuron. w_{i_1,i_1} spikes are transported to $\gamma_{(i_1-1)n+i_1}$ immediately and the same operation also occurs on $\beta_{(i_2-1)n+i_2}$ to $\beta_{(i_k-1)n+i_k}$. As a result, the number of spikes preserved in α_{i_1}, α_{n+i_1}, $\ldots$, $\alpha_{n(n-1)+i_1}$, $\ldots$, α_{i_k}, α_{n+i_k}, $\ldots$, $\alpha_{n(n-1)+i_k}$ and $\beta_{i_1}, \ldots, \beta_{n(n-1)+i_1}, \ldots, \beta_{i_k}, \ldots, \beta_{n(n-1)+i_k}$ decreases to zero and the neurons change into inactive with no capable rules. In the following steps, the minimum weight between the rest $(n - k)$ non-medoid objects and their most similar centers is selected one after another and the *Comparison Stage* halts in step $(k + w_{i_p,i_q} + 1)$ $(i_p \in\{i_{k+1}, i_{k+2}, \ldots, i_n\}$, $i_q \in \{i_1, i_2, \ldots, i_k\})$, where w_{i_p,i_q} is the maximum of n minimum weights associated with n objects. Using the method suggested above, if v_i $(1 \leq i \leq n)$ has the equal minimum weight with more than one centroid, such as v_{i_j} and v_{i_l} $(i_j, i_l \in \{i_1, \ldots, i_k\}$, $i_j \neq i_l)$, then w_{i,i_j} $(w_{i,i_j} = w_{i,i_l})$ spikes are delivered to $\gamma_{(i_j-1)n+i}$ and $\gamma_{(i_l-1)n+i}$ for further application.

Selection Stage: As the module is constituted by $(n^2 - n + 1)$ subsets mentioned in section 4, in step $(k + w_{i_p,i_q} + 2)$, one of them is chosen to apply the rule whereas neurons in other sets keep unfired. Since the order of activating different sets doesn't make an effect on the computing result, we presume that the set consisting of $\gamma_{(i_1-1)n+i_1}$, $\gamma_{(i_2-1)n+i_2}$, $\cdots$, $\gamma_{(i_k-1)n+i_k}$ starts to work in the first place and each of them introduce w_{i_j,i_j} $(i_j \in \{i_1, \ldots, i_k\})$ spikes to μ_1 and H. μ_1 obtains the spikes immediately and accumulates them for spiking preparation. Spikes received by the environment are regarded as a part of the clustering result computed by this iteration. In step $(k + w_{i_p,i_q} + 3)$, $\gamma_{(i_1-1)n+i_1}$, $\ldots$, $\gamma_{(i_k-1)n+i_k}$ have no rules to use and one of $\gamma_{(i_l-1)n+i_j}$ $(i_j \in \{i_{k+1}, \ldots, i_n\}$, $i_l \in \{i_1, \ldots, i_k\})$ conveys w_{i_j,i_l} spikes to μ_1 and H the same as before. Moreover, one spike should be moved from γ_{i_j}, γ_{n+i_j} $\ldots$, $\gamma_{(n-1)n+i_j}$ (expect $\gamma_{(i_l-1)n+i_j}$), because synapses start from $\gamma_{(i_l-1)n+i_j}$ have the weight of -1. If they have spikes gained from the computation in the last stage, then these spikes will vanish as the number of them is less than the potential threshold. If not, the potential of them remains to zero and no more rules can be applied either. In this way, as for each v_i with $w_{i,i_j} = w_{i,i_l}$ $(1 \leq i \leq n$, $i_j, i_l \in \{i_1, \ldots, i_k\}$, $i_j \neq i_l)$, only one of them with will be non-deterministically chosen and calculated in the total cost of S. The following $(n - k - 1)$ steps proceeds in the same way as explained in step $(k + w_{i_p,i_q} + 3)$ and all the n minimum weights associated with n vertices are summed up in μ_1 after step $(w_{i_p,i_q} + n + 2)$. Also in that step, S_{max} spikes are sent to μ_2 from the environment, preparing for the comparison with S in the next stage.

Determination Stage: In step $(w_{i_p,i_q} + n + 3)$, the rule $(a^{S_{max}} \to a^{S_{max}})$ delivers S_{max} spikes to μ_3. Simultaneously, $(a^S \to a^S)$ can be used in μ_1 and cause S spikes lost in μ_3 due to the negative weight. Note that two possible cases come into being: 1. If $S < S_{max}$, $(a^{S_{max}-S} \to a)$ included in μ_3 can be applied and one spike is duplicated by $(k + 1)$ in step $(w_{i_p,i_q} + n + 4)$, making the number of spikes in θ_{i_1} to θ_{i_k} increase to $(k + 1)$ and the other neurons consist of $(3k + 1)$ spikes. 2. If $S = S_{max}$, the computation finishes and w_{1,i_1}, w_{2,i_2}, $\ldots$, w_{n,i_n} spikes fired by $\gamma_{(i_1-1)n+i_1}$, $\gamma_{(i_2-1)n+i_2}$, $\cdots$, $\gamma_{(i_n-1)n+i_n}$ in H are seen as the clustering result in the end (with the meaning that each $v_i(1 \leq i \leq n)$ has been classified into one cluster successfully although this solution isn't the optimal one). From the above explanation of the computing process, it's not difficult to find that a better clustering result can be gained if and only if $S < S_{max}$ holds true and a new round of iteration will begin in step $(w_{i_p,i_q} + n + 5)$.

Since θ_{i_1} to θ_{i_k} are capable of applying the rule $(a^{k+1} \to a)$, one of them is activated, distributing a spike to the rest $(n - 1)$ neurons in the *Generation Module*, as well as n neurons located in the *Comparison Module* after duplicated by the synapse weight. After that, the second selected neuron enables to use $(a^{k+2} \to a)$ and changes the amount of spikes contained in $\theta_{i_{k+1}}, \ldots, \theta_{i_n}$ into $(3k + 3)$. Functioning in the asynchronous mode, $(k - 1)$ neurons in $\{\theta_{i_1}, \ldots, \theta_{i_k}\}$ spike the second time from step $(w_{i_p,i_q} + n + 5)$ to step $(w_{i_p,i_q} + n + k + 3)$ and the potential of them becomes zero again. In step $(w_{i_p,i_q} + n + k + 4)$, with $4k$ spikes remaining in $\theta_{i_{k+1}}, \ldots, \theta_{i_n}$, its essential to pick one associated with a new centroid O' and the neuron could execute $(a^{4k}/a^{3k} \to \lambda)$ and $(a^k \to a)$ sequentially. In the next $(n - k)$ steps, two forgetting rules $(a^{4k+1}/a^{2k+1} \to \lambda)$ and $(a^{2k+1}/a \to \lambda)$ are applicable in those neurons indicating the new $(n - k)$ non-medoid objects, leaving $2k$ spikes contained in them just like the former iteration. The *Generation Stage* stops working in step $(w_{i_p,i_q} + 2n + 5)$, and it turns to the *Comparison Module* and *Selection Module* to produce a new clustering pattern of n vertices. The computation process continues until $S' \geq S$ with the clustering result clearly reflected in H, and Table (3) shows the resource summary of EASN P systems constructed in this work.

It is important to point out that the duration of each iteration is limited to $O(n^2)$, while the computation complexity of the original PAM clustering algorithm is $O(k(n - k)^2)$. In the best case, only three steps are cost in the *Comparison Stage*, with the restriction that w_{ij} equals to 2 for each v_i and v_j $(i \neq j)$

Table 3: Summary of resources used in the EASN P system

Number	Generation Module	Comparison Module	Selection Module	Decision Module	Sum
Neuron	n	$2n^2$	n^2	3	$3n^2+n+3$
Threshold (P_i)	k	1	$w_{i,1}, w_{i-n,2},$ $\ldots,$ $w_{i-n(n-1),n}$	1	/
Initial Spike	kn	0	0	$(n^3-(3+k)n^2+5kn$ $+2n-2k^2)/2$	$(n^3-(3+k)n^2+7kn$ $+2n-2k^2)/2$
Rule	$(k+3)n$	$2n^2$	n^2	$(3n^3-(9+3k)n^2+15kn$ $-6n-6k^2+6k+4)/2$	$(3n^3-(3+3k)n^2+17kn$ $-6k^2+6k+4)/2$
Set	n	1	$n(n-1)+1$	1	n^2+3
Step	k or $n+1$	$min:3$ $max:$ $n(n-1)/2+1$	$n-k+1$	2	$min:n+6$ $max:$ $(n^2+3n-2k+10)/2$

Table 4: Three properties of senti-graphs derived from 11 customer reviews

$s(SG_i,SG_j)$ w_{ij}	1	2	3	4	……	9	10	11
1	1	0.488	0.333	0.156		0.160	0.129	0.090
(6)	1	4	15	38		37	41	43
2	0.488	1	0.375	0.200		0.143	0.077	0.087
(15)	4	1	11	33		39	46	45
3	0.333	0.375	1	0.279	……	0.219	0.162	0.107
(10)	15	11	1	23		30	36	42
4	0.156	0.200	0.279	1		0.400	0.160	0.182
(21)	38	33	23	1		10	37	34
				……				
8	0.222	0.211	0.324	0.368		0.430	0.308	0.375
(15)	29	31	16	12		7	18	11
9	0.160	0.143	0.219	0.400		1	0.320	0.375
(15)	37	39	30	10		1	17	11
10	0.129	0.077	0.162	0.160	……	0.320	1	0.400
(21)	41	46	36	37		17	1	10
11	0.090	0.087	0.107	0.182		0.375	0.400	1
(15)	43	45	42	34		11	10	1

in graph G. In the worst case, $w_{i_p,i_q}=n(n-1)/2$ and $(n(n-1)/2+1)$ steps are required before completing the function of the *Comparison Module*.

6 Experiments and Results Analysis

In this section, a real-word case is presented to illustrate how EASN P systems deal with the sentiment clustering and classify customer reviews effectively. Table (4) indicates 11 records of the CLINIQUE Moisturizing Gel, a popular skin care product sold on JUMEI.COM, and they are uniquely identified by a number between 1 and 11. The preprocessing steps need to segment these Chinese records into separate words by ICTCLAS and extract those with sentiment polarity using NTUSD. According to the computing method mentioned in section 3, $|SG_i|$ $(1 \leq i \leq 11)$ is calculated and listed in the first column with parentheses. $s(SG_i,SG_j)$ $(1 \leq i,j \leq 11)$ is written in first line of each review respectively, with w_{ij} shown below and used in the computing process of EASN P systems. The core structure of an EASN P system is presented in Figure (3) and the function of four modules will be specified graphically.

Table 5: One possible computation of the EASNP system based on sentiment clustering algorithm

Steps in the $1st$ iteration	Neuron	Spike (Rule)	
1	θ_2	$3(a^3 \to a)$	
2	θ_3	$4(a^4 \to a)$	
3	θ_8	$5(a^5 \to a)$	
4	$\alpha_{12}, \alpha_{13}, \ldots, \alpha_{33}, \alpha_{78}, \ldots, \alpha_{88}$	$4(a^+/a \to a), 1(a^+/a \to a), \ldots, 42(a^+/a \to a), 29(a^+/a \to a), \ldots, 11(a^+/a \to a)$	
5	$\alpha_{12}, \alpha_{14}, \ldots, \alpha_{24}, \alpha_{26}, \ldots, \alpha_{33}, \alpha_{78}, \ldots, \alpha_{84}, \alpha_{86}, \alpha_{87}, \alpha_{88}, \beta_{13}, \beta_{25}, \beta_{85}$	$3(a^+/a \to a), 10(a^+/a \to a), \ldots, 10(a^+/a \to a), 22 (a^+/a \to a), \ldots, 41(a^+/a \to a), 28(a^+/a \to a), \ldots, 12 (a^+/a \to a), 6(a^+/a \to a), 17(a^+/a \to a), 10(a^+/a \to a), 1(a \to a), 1(a \to a), 1(a \to a)$	
6	$\alpha_{12}, \alpha_{15}, \alpha_{17}, \alpha_{18}, \alpha_{20}, \ldots, \alpha_{23}, \alpha_{26}, \ldots, \alpha_{29}, \alpha_{31}, \ldots, \alpha_{33}, \alpha_{78}, \alpha_{81}, \ldots, \alpha_{84}, \alpha_{86}, \alpha_{87}, \alpha_{88}, \beta_{16}$	$2(a^+/a \to a), 31(a^+/a \to a), 3(a^+/a \to a), 17(a^+/a \to a), 37(a^+/a \to a), \ldots, 13(a^+/a \to a), 21(a^+/a \to a), \ldots, 12(a^+/a \to a), 28(a^+/a \to a), \ldots, 40(a^+/a \to a), 27 (a^+/a \to a), 10(a^+/a \to a), \ldots, 11(a^+/a \to a), 5(a^+/a \to a), 16(a^+/a \to a), 9(a^+/a \to a), 2(a^2 \to a)$	
7	$\alpha_{12}, \alpha_{15}, \alpha_{17}, \alpha_{18}, \alpha_{20}, \ldots, \alpha_{23}, \alpha_{26}, \alpha_{28}, \alpha_{29}, \alpha_{31}, \ldots, \alpha_{33}, \alpha_{78}, \alpha_{81}, \alpha_{83}, \alpha_{84}, \alpha_{86}, \alpha_{87}, \alpha_{88}$	$1(a^+/a \to a), 30(a^+/a \to a), 2(a^+/a \to a), 16(a^+/a \to a), 36(a^+/a \to a), \ldots, 12(a^+/a \to a), 20(a^+/a \to a), 3 (a^+/a \to a), 11(a^+/a \to a), 27(a^+/a \to a), \ldots, 39(a^+/a \to a), 26(a^+/a \to a), 9(a^+/a \to a), 12(a^+/a \to a), 10 (a^+/a \to a)$ 22, $4(a^+/a \to a), 15(a^+/a \to a), 8(a^+/a \to a)$	
		……	
22	$\alpha_{21}, \alpha_{32}, \beta_{87}$	$28(a^+/a \to a), 18(a^+/a \to a), 18(a^{18} \to a)$	
23	$\gamma_{13}, \gamma_{25}, \gamma_{85}$	$1(a \to a), 1(a \to a), 1(a \to a)$	
24	γ_{16}	$2(a^2 \to a)$	
25	γ_{87}	$18(a^{18} \to a)$	
26	γ_{86}	$7(a^7 \to a)$	
27	γ_{12}	$4(a^4 \to a)$	
28	γ_{88}	$11(a^{11} \to a)$	
29	γ_{17}	$5(a^5 \to a)$	
30	γ_{81}	$12(a^{12} \to a)$	
31	γ_{84}	$13(a^{13} \to a)$	
32	μ_1, μ_2	$75(a^{75} \to a), 387(a^{387} \to a)$	
33	μ_3	$312(a^{312} \to a)$	
Weights	$w_{12}, w_{22}, w_{33}, w_{48}, w_{52}, w_{62}, w_{78}, w_{88}, w_{98}, w_{10,8}, w_{11,8}$	Medoids (vertices)	$v_2(v_1, v_5, v_6), v_3, v_8(v_4, v_7, v_9, v_{10}, v_{11})$
		……	
Weights	$w_{12}, w_{22}, w_{36}, w_{48}, w_{52}, w_{66}, w_{76}, w_{88}, w_{98}, w_{10,8}, w_{11,8}$	Medoids (vertices)	$v_2(v_1, v_5), v_6(v_3, v_7), v_8 (v_4, v_9, v_{10}, v_{11})$
		……	
Weights	$w_{16}, w_{26}, w_{36}, w_{48}, w_{56}, w_{66}, w_{76}, w_{88}, w_{98}, w_{10,11}, w_{11,11}$	Medoids (vertices)	$v_6(v_1, v_2, v_3, v_5, v_7), v_8 (v_4, v_9), v_{11}(v_{10})$

brought forward...

Steps in the $4th$ iteration	Neuron	Spike (Rule)	
112	θ_6	$4(a^4 \to a)$	
113	θ_{11}	$5(a^5 \to a)$	
114	θ_9	$12(a^{12}/a^9 \to \lambda)$	
115	θ_9	$3(a^3 \to a)$	
116	θ_8	$7(a^7/a \to \lambda)$	
117	θ_1	$13(a^{13}/a^7 \to \lambda)$	
118	θ_7	$13(a^{13}/a^7 \to \lambda)$	
119	θ_5	$13(a^{13}/a^7 \to \lambda)$	
120	θ_3	$13(a^{13}/a^7 \to \lambda)$	
121	θ_{10}	$13(a^{13}/a^7 \to \lambda)$	
122	θ_2	$13(a^{13}/a^7 \to \lambda)$	
123	θ_4	$13(a^{13}/a^7 \to \lambda)$	
124	$\alpha_{56}, \ldots, \alpha_{66}, \alpha_{89}, \ldots, \alpha_{99}, \alpha_{111}, \ldots, \alpha_{121}$	$8(a^+/a \to a), \ldots, 35(a^+/a \to a), 37(a^+/a \to a), \ldots, 11 (a^+/a \to a), 43(a^+/a \to a), \ldots, 1(a^+/a \to a)$	
125	$\alpha_{56}, \ldots, \alpha_{60}, \alpha_{62}, \ldots, \alpha_{66}, \alpha_{89}, \ldots, \alpha_{96}, \alpha_{98}, \alpha_{99}, \alpha_{111}, \ldots, \alpha_{120}, \beta_{61}, \beta_{97}, \beta_{121}$	$7(a^+/a \to a), \ldots, 8(a^+/a \to a), 8(a^+/a \to a), \ldots, 34 (a^+/a \to a), 36(a^+/a \to a), \ldots, 6(a^+/a \to a), 16(a^+/a \to a), 10(a^+/a \to a), 42(a^+/a \to a), \ldots, 9(a^+/a \to a), 1(a \to a), 1(a \to a), 1(a \to a)$	
126	$\alpha_{56}, \ldots, \alpha_{60}, \alpha_{62}, \alpha_{63}, \alpha_{65}, \alpha_{89}, \ldots, \alpha_{93}, \alpha_{95}, \alpha_{96}, \alpha_{98}, \alpha_{111}, \ldots, \alpha_{115}, \alpha_{117}, \alpha_{118}, \alpha_{120}$	$6(a^+/a \to a), \ldots, 7(a^+/a \to a), 7(a^+/a \to a), 13(a^+/a \to a), 30(a^+/a \to a), 35(a^+/a \to a), \ldots, 38(a^+/a \to a), 22(a^+/a \to a), 5(a^+/a \to a), 15(a^+/a \to a), 41(a^+/a \to a), \ldots, 45(a^+/a \to a), 38(a^+/a \to a), 9(a^+/a \to a), 8 (a^+/a \to a)$	
		$\ldots\ldots$	
134	$\alpha_{59}, \alpha_{65}, \alpha_{98}, \alpha_{114}, \beta_{92}, \beta_{120}$	$15(a^+/a \to a), 22(a^+/a \to a), 7(a^+/a \to a), 24(a^+/a \to a), 10(a^{10} \to a), 10(a^{10} \to a)$	
135	γ_{58}	$6(a^6 \to a)$	
136	γ_{120}	$10(a^{10} \to a)$	
137	γ_{92}	$10(a^{10} \to a)$	
138	$\gamma_{61}, \gamma_{97}, \gamma_{121}$	$1(a \to a), 1(a \to a), 1(a \to a)$	
139	γ_{57}	$5(a^5 \to a)$	
140	γ_{60}	$9(a^9 \to a)$	
141	γ_{56}	$8(a^8 \to a)$	
142	γ_{96}	$7(a^7 \to a)$	
143	γ_{62}	$9(a^9 \to a)$	
144	μ_1, μ_2	$67(a^{67} \to a), 69(a^{69} \to a)$	
145	μ_3	$2(a^2 \to a)$	
Weights	$w_{16}, w_{26}, w_{36}, w_{49}, w_{56}, w_{66}, w_{76}, w_{89}, w_{99}, w_{10,11}, w_{11,11}$	Medoids (vertices)	$v_6(v_1, v_2, v_3, v_5, v_7), v_9 (v_4, v_8), v_{11}(v_{10})$
		$\ldots\ldots$	
Weights	$w_{16}, w_{26}, w_{36}, w_{49}, w_{56}, w_{66}, w_{76}, w_{89}, w_{99}, w_{10,10}, w_{11,10}$	Medoids (vertices)	$v_6(v_1, v_2, v_3, v_5, v_7), v_9 (v_4, v_8), v_{10}(v_{11})$

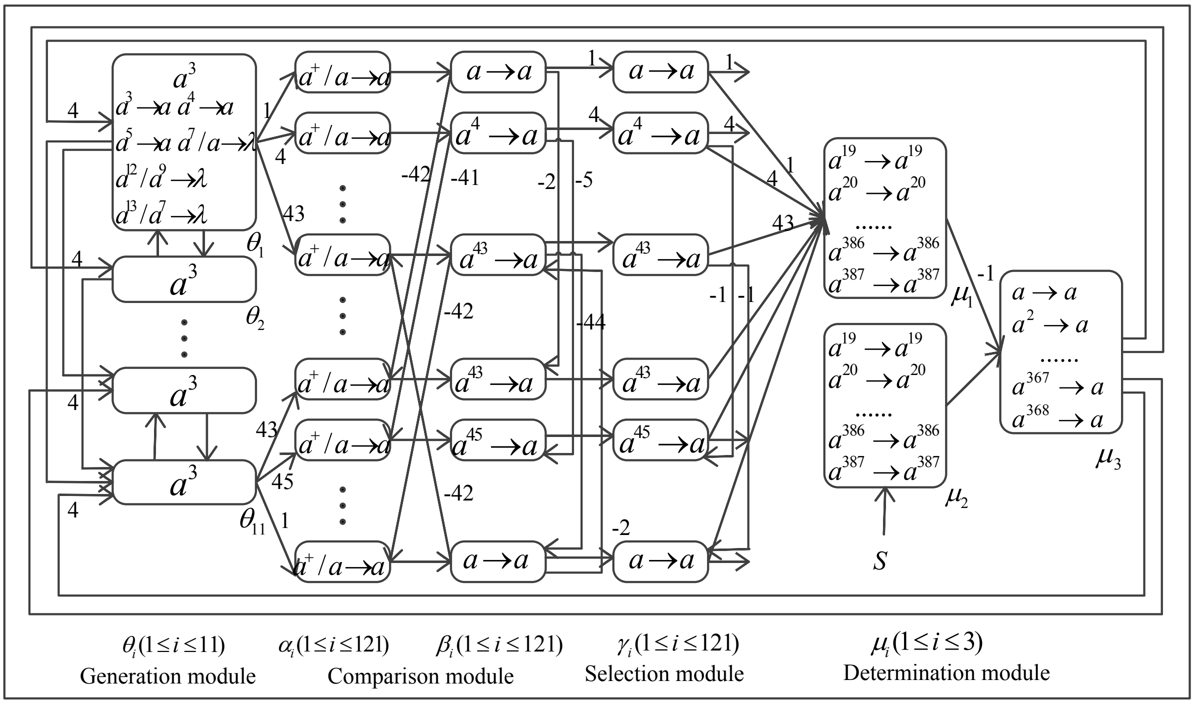

Figure 3: The EASNP system generated according to the given data

Table 6: Sentiment Clustering results of 11 Customer reviews

Cluster	Key words	Other sentiment words
1	suitable, promote, balance, avoid, waste	lovely, very good, comfortable, the best one, oily, effectively, prevent, allergy, specially, enjoy, without, easily, worthy, recommendation, like, burden
2	comfortable, a little bit, old, abrasion	without, with no idea, whether or not, useful, relatively, mild, suitable, worry, expect
3	many, indistinct, allergy, poor, inconformity	none, worthy, oily, requirement, to one's surprise, disappointment

Table (5) denotes one possible computation including five iterations and stopping when the termination condition is satisfied. As for the user-provided parameter, the number of clusters is assigned by three and it could be larger with the expansion of initial solution space. In each step, the spiking neuron is showed in the second column and the third row reveals the number of reserving spikes with available rules written in brackets. The clustering result is listed in the end of each iteration and all the vertices in parentheses have been designated to a centroid of one specific cluster. In the given case, the algorithm halts in step 179 with the condition that $S' = S = 67$. As we can see Figure (4) drawn by Matlab R2010a, v_6, v_9, v_{11} produced in the $4th$ iteration are considered to be the best centroids associated with three groups and 11 customer reviews are divided into different clusters based on their sentiment similarity finally.

Since vertex 1, 2, 3, 5, 6, 7 are classified into cluster 1, sentiment words contained in the $6th$ sentigraph are regarded as the key words of this cluster and the words representing other reviews are sorted in accordance with the order by descending w_{i6} $(i = 1, 2, 3, 5, 7)$. Same operations are conducted on cluster 2 and cluster 3, which consist of vertex 4, 8, 9 and vertex 10, 11 respectively. The clustering result about 39 sentiment words is shown in Table (6) and customers are able to browse comment records with different emphases and obtain valuable information of this popular product with the help of these representative sentiment words listed in front of each review group. If hundreds of reviews are classified according to this sentiment analysis method, it will bring great convenience to shoppers and manufacturers, shortening the time of opinion extraction and speeding up the marketing research on the product appraisal given by online consumers.

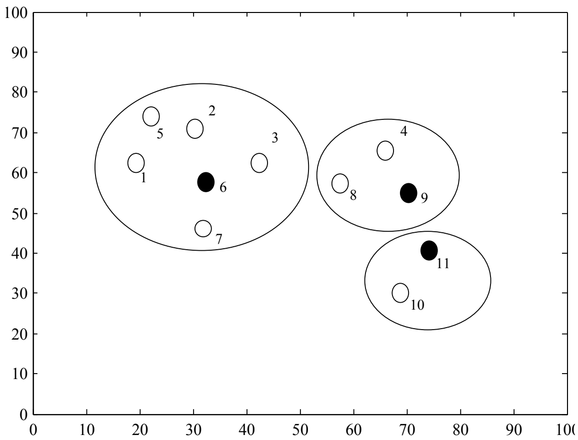

Figure 4: Clustering results gained from the computer simulation

7 Final Remarks

In this paper, a variant of SN P systems, an EASN P system is introduced, using weighted synapses and potential threshold to realize the sentiment clustering of Chinese customer reviews based on the graph representation. Form the theoretical point of view, the computational complexity of our systems is limited to $O(n^2)$, whereas the traditional PAM clustering algorithm is $O(k(n-k)^2)$. The whole system is divided into four sets, functioning in an asynchronous manner to achieve the four main procedures in the sentiment clustering algorithm. Spiking rules are exhaustively applied in each active neuron and weighted synapses working in corporation with the potential threshold make different effects on the spike transmission and neuron evolution. Its highly conceivable that modeling each customer review as a senti-graph contributes to keeping its inherent structure and we perform the sentiment clustering algorithm on these senti-graphs to find out their emotion orientations instead of the topic relevance between them. Different from classifying the reviews into positive and negative categories, the goal of sentiment clustering is to make full use of the complexity of online shoppers' attitudes and emotions, providing a convenient and fast way for potential clients to filter out useful suggestions the users expect to highlight and collect the product information they actually need.

Nevertheless, there are also some drawbacks worth investigating further. The number of clusters needs to be predefined and more than n^3 rules and initial spikes should be prepared for the EASN P system. In our future work, it seems promising to use this SN P system to solve other clustering problems based on the graph representation and apply the way of sentiment analysis to more practical applications both online and offline.

Acknowledgements: The research was supported by National Natural Science Foundation of China (No. 61170038), Humanities and Social Sciences Project of Ministry of Education, China (NO. 12YJA630152), Natural Science Foundation of Shandong Province, China (NO. ZR2011FM001), Social Science Foundation of Shandong Province, China (NO. 11CGLJ22).

References:

[1] L. Q. Pan, X. X. Zeng, X. Y. Zhang and Y. Jiang, Spiking neural P systems with weighted synapses, *Neural Processing Letters* 35, 2012, pp. 13–27.

[2] T. Song, L. Q. Pan and G. Păun, Asynchronous spiking neural P systems with local synchronization, *Information Sciences* 219, 2013, pp. 197–207.

[3] M. Cavaliere, O. H. Ibarra, G. Păun, O. Egecioglu, M. Ionescu and S. Woodworth, Asynchronous spiking neural P systems, *Theoretical Computer Science* 410, 2009, pp. 2352–2364.

[4] J. Wang, H. J. Hoogeboom, L. Q. Pan and G. Păun, Spiking neural P systems with weights and thresholds, *The 10th Workshop on Membrane Computing (WMC10)*, Research Group on Natural Computing, Sevilla University 2009

[5] J. Wang, H. J. Hoogeboom and L. Q. Pan, Spiking neural P systems with neuron division, *Membrane Computing* 6501, 2011, pp. 361–376.

[6] L. F. Macías-Ramos and M. J. Pérez-Jiménez, Spiking neural P systems with functional astrocytes, *The 13th International Conference on Membrane Computing (CMC13)*, Springer–Verlag, Berlin–Heidelberg 2012

[7] S. Feng, D. L. Wang, G. Yu, C. Yang and N. Yang, Sentiment clustering: a novel method to explore in the blogosphere, *Advances in Data and Web Management* 5446, 2009, pp. 332–344.

[8] S. Feng, D. L. Wang, G. Yu, C. Yang and N. Yang, Chinese blog clustering by hidden sentiment factors, *Advanced Data Mining and Applications* 5678, 2009, pp. 140–151.

[9] S. Feng, J. Pang, D. L. Wang, G. Yu, F. Yang and D. P. Xu, A novel approach for clustering sentiments in Chinese blogs based on graph similarity, *Computers and Mathematics with Applications* 62, 2011, pp. 2770–2778.

[10] D. Gräbner, M. Zanker, G. Fliedl and M. Fuchs, Classification of customer reviews based on sentiment analysis, *Information and Communication Technologies in Tourism 2012*, Springer –Verlag, Berlin–Heidelberg 2012

[11] A. Khan, B. Baharudin and K. Khan, Sentiment classification from online customer reviews using lexical contextual sentence structure, *Software Engineering and Computer* 179, 2011, pp. 317–331.

[12] J. L. Tian and W. Zhao, Words similarity algorithm based on Tongyici Cilin in semantic web adaptive learning system, *Journal of Jilin University (Information Science Edition)* 28, 2010, pp. 602–608.

[13] Y. Z. Zhao, X. Y. Liu and J. H. Qu, The K-medoids clustering algorithm by a class of P system, *Journal of Information and Computational Science* 9, 2012, pp. 1–10.

[14] M. Ionescu, G. Păun and T. Yokomori, Spiking neural P systems, *Fundamenta Informaticae* 71, 2006, pp. 279–308.