Chapter 1
Simulating and Modeling Dual Market Segmentation Using PSA Framework

Jiamou Liu, Ziheng Wei and Quan Bai

Abstract Market segmentation refers to the analytical process of dividing a broad market into segments taking into account multiple factors such as consumer needs, interests and tastes; it has been considered one of the most important marketing strategies as it helps a business to identify hidden market trends, define target segments, and design marketing plans. Market segmentation may also be viewed as a computational challenge: Given the massive amount of data describing interactions between consumers and commodities, the task is to partition the set of consumers and commodities into subsets that corresponds to market segments – two consumers are in the same segments when they exhibit a similar purchasing pattern, while two products are in the same segments when they are purchased by a similar group of consumers. In this work, we focus on the definition and simulation of market segments. We employ the Propose-Select-Adjust (PSA) framework, introduced in an earlier work [9], to simulate the forming of market segments. Our approach is distributed and can be applied to large and dynamic market data set. The experimental results suggest that the proposed approach is a promising technique for supporting intelligent market segmentation.

Key words: Market segmentation, PSA framework, bipartite network

Jiamou Liu[1] · Ziheng Wei[2]
Department of Computer Science, The University of Auckland, Auckland, New Zealand, [1]e-mail: jiamou.liu@auckland.ac.nz [2]e-mail: zwei891@aucklanduni.ac.nz

Quan Bai
School of Engineering, Computer and Mathematical Sciences, Auckland University of Technology, Auckland, New Zealand
e-mail: quan.bai@aut.ac.nz

1.1 Introduction

"I Shop Therefore I Know That I Am", stated British sociologist Colin Campbell in a 2004 research article; this sentence has become a motto of modern consumerism: people increasingly justify their existence and identify themselves by the products they shop [1]. Indeed, the theory of "social groups" asserts that consumption is intimately tied to the question of "with whom do I belong" and the formation of collective identities – the so-called *social groups*; they capture notions such as trends, style and sub-cultures [13]. Analysing social groups helps us to decode the hidden patterns of consumer behaviours, differentiate sophisticated broad markets and identify trends and dynamics within them.

From a marketer's perspective, understanding the market amounts to understand the desire, preference and priorities of consumers; this task is called *market segmentation*. Market segmentation is one of the most important marketing strategies. Its aim is to divide a broad market into smaller groups, taking into account different factors such as customers' interests, and tastes, revealing important information regarding lifestyles, geographic differences, and other demographic and economical phenomena. With this information, a business defines its target customers, enabling more accurate orientation of marketing plans, allowing more efficient use of resources and hence increasing profits.

Market segmentation is a complex and data intensive task. In order to pinpoint the exact correlation between the consumption behaviours and consumers' life styles and trends, analysts need to process and catagorise a huge amount of market data such as shopping records. In the past 5 years, there has been a significant interest in computational approaches that support automated market segmentation and the problem has become a central problem in data mining [12, 11, 14, 2]. One common goal of these works is to build an intelligent tool which, through analysing large amounts of market data, computes an appropriate segmentation of the market.

Research on market segmentation exhibits a diverse landscape. For example, Miguis, et. al. recently use lifestyle information derived from market segmentation to analyse customer purchasing power [12]. In another work, the same authors also investigate the design of product promotion strategies based on market segmentation results [11]. Market segmentation has also been applied in customer loyalty management [14].

We point out here that market segmentation can be viewed from two perspectives: the *consumers*' perspective and the *commodities*' perspective. On one hand, market segmentation aims to categorise consumers into different social groups. For example, two consumers may be characterised as having a similar purchasing pattern and fall into the same social group. On the other hand, the process also aim to distinguish clusters of commodities. For example, two products may be seen as correlated as they are likely to be purchased by the same group of consumers. These two perspectively naturally influence each other, as the forming of a consumer group may help to identify a new commodity segment, and the introduction of a range of products may also enforce the emergence of a consumer group. The relation be-

tween consumers and commodities are interactive, highly dynamic and often subtle. It is therefore an important question to analyse interactions between consumers and commodities.

Based on the above observations, we propose the *dual market segmentation problem*. The underlying framework treats a market as a bipartite graph consisting of consumer nodes and commodity nodes, and interactions between them. The goal of the problem is to design a computational approach for segmenting – at the same time – both the consumers and commodities, taking into account these interactions.

In our previous work [10], we proposed a decentralised computational framework, Propose -Select -Adjust (PSA), for solving network problems. In this framework, each node of the graph acts as a individual computational unit which performs three procedures in cyclic order: Propose, Select and Adjust. The nodes are acting in an asynchronous manner while each node only has access to its local information. Through a series of case studies and experiments, we demonstrated the approach is a promising solution for simulating community formation, and hence detecting communities in a large dynamic network [9]. In this paper, we extend the PSA framework on the bipartite market network connecting consumers and commodities. We suggest that PSA can be applied to market segmentation. This approach has several advantages:

- Firstly, PSA is a technology for smart simulation of group behaviors in a network. Thus it simulates the formation of segments within the consumers and commodities. This simulation naturally captures the mutual influence between consumers and commodities, which is otherwise hard to define.
- Secondly, PSA as a distributed framework, can scale to handle large amount market data, while computation is kept local.
- Thirdly, as a PSA cell runs continuously, the PSA cells monitors the changing market, and detects emergence of market segments.

We support our claims above with experimental results on both synthesised and real data.

The rest of the paper is organised as follows. Section 1.2 formulates the dual market segmentation problem using a bipartite graph model. Section 1.3 describes the PSA framework and extend it to the bipartite dual network of consumers and commodities. Our new framework provides two perspectives: The *local perspective* segments consumers and commodities based on their links. The *global perspective*, on the other hand, captures the interactions between the segments identified in the local perspective. Our experimental results are presented and discussed in Section 1.4. Some related works are reviewed in Section 1.5. Finally the paper is concluded in Section 1.6.

1.2 Problem Formulation

A *bipartite graph* is a pair $(V_0 \cup V_1, E)$ where V_0 and V_1 are two sets of nodes with $V_0 \cap V_1 = \varnothing$, and the edge relation E is a subset of the Cartesian product $V_0 \times V_1$. We use bipartite graph to abstractly model transactions in a market. In particular, we may view V_0 as a set of *consumers*, and V_1 as a set of *commodities* in a market; an edge $(v, p) \in V_0 \times V_1$ denote that consumer v purchases the commodity p. Hence this bipartite graph captures a collection of purchasing records of all consumers in V_0. We point out that the terms "consumers" and "commodities" are used in a broad sense: these are two abstract concepts which captures any interacting objects in a market-like context.

This bipartite graph model involves a *dual network*: On one hand, from the perspective of a consumers, its outgoing edges indicate its *purchased commodities* – intuitively, two consumers have similar purchasing behaviours if they purchase similar commodities. On the other hand, from the perspective of a commodity, its incoming edges indicate its *consumer base* – intuitively, two commodities have similar consumer base, if they are purchased by similar groups of customers.

The main problem of the paper is to compute a segmentation of this dual network. For consumers, a segmentation amounts to dividing the market into groups of consumers with similar purchasing patterns which are affected by consumers' tastes, life styles, desires, etc. For commodities, a segmentation amounts to aggregation of correlated products on the market. Formally, we model market segmentation as follows:

Definition 1 (Dual segmentation). Let $G = (V_0 \cup V_1, E)$ be a bipartite graph. A *dual segmentation* is a pair $(\sim_0, \sim_1)$ where $\sim_i$ is an equivalence relation on V_i for $i \in \{0, 1\}$. An $\sim_i$-equivalence class is called a *V_i-segment*.

For any consumer $v \in V_0$, the V_0-segment of v contains all consumers $u \in V_0$ such that $v \sim_0 u$; for any commodity $x \in V_1$, the V_1-segment of x contains all products $y \in V_1$ such that $x \sim_1 y$. We observe that, in the context of a market, the two components of a dual segmentation are interrelated. For example, consumers with a similar lifestyle tend to purchase a similar group of products, which in turn causes correlation among these products. Therefore, our goal is to provide a uniform approach for computing a dual segmentation, in which the segments of consumers influence segments of commodities, and vice versa.

This approach builds the two networks involved in the market: the consumer network, and the commodity network. A *similarity function* on a set S is a function $\omega : S \times S \to \mathbb{R}$. The key ingredients of our approach are two similarity functions, ω_0, ω_1, that are defined on the two sets V_0 and V_1, respectively. Our similarity functions ω_0, ω_1 are based on *Jaccard similarity*, which is one of the most commonly used similarity measure for sets. Given a bipartite graph $G = (V_0 \cup V_1, E)$, for $u \in V_0$, $v \in V_1$, define

$$E(u) = \{v' \in V_1 \mid (u, v') \in E\} \text{ and } E^{-1}(v) = \{u' \in V_0 \mid (u', v) \in E\}.$$

The similarity $\omega_0(v_1, v_2)$ between $v_1, v_2 \in V_0$ is the Jaccard similarity

$$\omega_0(v_1, v_2) = \frac{|E(v_1) \cap E(v_2)|}{|E(v_1) \cup E(v_2)|} \tag{1.1}$$

The similarity $\omega_1(v_1, v_2)$ between $v_1, v_2 \in V_1$ is the Jaccard similarity

$$\omega_1(v_1, v_2) = \frac{|E^{-1}(v_1) \cap E^{-1}(v_2)|}{|E^{-1}(v_1) \cup E^{-1}(v_2)|} \tag{1.2}$$

Using these two similarity functions, two networks can be constructed:

Definition 2 (V_i-networks). Let $G = (V_0 \cup V_1, E)$ be a bipartite graph, and ω_0, ω_1 be two similarity functions on V_0 and V_1 respectively. For any $i \in \{0, 1\}$, the *V_i-network* is an undirected graph (V_i, E_i) where

$$E_i = \left\{(u_1, u_2) \in V_i^2 \mid \omega_i(u_1, u_2) \geq \alpha_i\right\}$$

where the constants α_i is called the *local similarity threshold*.

We call V_0-network and V_0-segments as *consumer network* and *consumer segments*, and call V_1-network and V_1-segments as *commodity network* and *commodity segments*, respectively.

In the rest of the paper, we describe a simulation of market segmentation in two steps: *local segmentation* and *global segmentation*. For local segmentation, we construct for each $i \in \{0, 1\}$, a set of disjoint V_i-segments $\mathscr{C}_i = \{C_{1,i}, ..., C_{\ell,i}\}$ where each $C_{j,i} \subseteq V_i$ and $\bigcup C_j \in \mathscr{C}_i = V_i$. Intuitively each $C_{j,0}$ represents a group of consumers who share some common interests. Thus each V_i-segment ideally should be a maximal clique (i.e., a complete subgraph) in the V_i-network. Since finding maximal cliques is a well-known NP-hard problem, we use the following notions to approximate a clique-like subgraph.

Definition 3 (Local core). In a graph $G = (V, E)$, $N(v)$, the *closed neighbourhood* of v, is a set $\{u \mid (u, v) \in E\} \cup \{v\}$ of all nodes that are adjacent to v. A *k-core* in a graph is an induced subgraph where all nodes have degree at least k. The *core number* of a node v is the largest $\kappa(v)$ such that $N(v)$ contains a $\kappa(v)$-core. For $v \in V$, the *local core* of v is the set

$$K(v) = \{u \in N(v) \mid |N(u) \cap N(v)| \geq \kappa(v) - 1\}$$

Intuitively, the local core of a node v is a subgraph that contains those nodes adjacent to v and are adjacent to at least $\kappa(v)$ many other nodes in the local core. The value of $\kappa(v)$ measure how "tightly-knitted" this subgraph is; when $\kappa(v)$ equals to the size of the local core minus 1, the local core is a clique.

The output of local segmentation consists of a set $\mathscr{C}_0$ of consumer segments and a set $\mathscr{C}_1$ of commodity segments. From the perspective of consumers, several consumer segments can share common interests towards the same commodity segments. From the perspective of commodities, several commodity segments are

deployed to the same consumer segments. This calls for possible combinations of these already computed segments. Therefore, in global segmentation, we take the edge set $E(C_1,C_2) = \{(u,v) \in C_0 \times C_1 \mid (u,v) \in E\}$, and construct a bipartite graph $G^H = (\mathscr{C}_0 \cup \mathscr{C}_1, E^H)$ where

$$E^H = \left\{ (C_1,C_2) \in \mathscr{C}_i \times \mathscr{C}_{1-i} \;\middle|\; \frac{|E(C_1,C_2)|}{|C_1|} \geq \beta_i, i \in \{0,1\} \right\}$$

where β_i is the *segmentation selection rate*, which determines the importance of the V_2-segment C_2 to the V_1-segment C_1.

Similarly to (1.1) and (1.2), we define *global similarity functions* ω_0^H and ω_1^H on $\mathscr{C}_0$ and $\mathscr{C}_1$, respectively. The graph G^H has corresponding dual networks: $\mathscr{C}_0$- and $\mathscr{C}_1$-network where

$$E_i^H = \{(C_1,C_2) \in \mathscr{C}_i^2 \mid \omega_i^H(C_1,C_2) \geq \gamma_i\}$$

where γ_i is the *global similarity threshold*. The similarity $\omega_i^H(C_1,C_2)$ between C_1, $C_2 \in \mathscr{C}_i$ where $i \in \{0,1\}$ is the Jaccard similarity.

1.3 The PSA Framework For Segmentation Simulation

1.3.1 General Description

The Propose-Select-Adjust (PSA) framework is a distributed computational framework for simulating a network of interconnected agents, called *cells*, which perform some homogeneous operations in an asynchronous manner. The framework was suggested by the authors in [10] to simulate community formation and detection in a distributed and dynamic setting.

In this paper, we extend the Propose-Select-Adjust (PSA) framework to simulate market segmentation based on a bipartite graph model. Intuitively, each node in the bipartite graph is a cell which carries out certain tasks independently. The PSA framework describes how to implement a single cell. The framework is inspired by the decision making process among a group of people: Imagine a group of individuals trying to decide on a partitioning of the group, where every member would belong to one and only one subgroup. The constraint is that each individual only sees local information about her own connections; no knowledge is shared among all members. Thus individuals can only make self-centred judgements and decide on the people that she would like to be with. Under this constraint, the following procedures can ensure the group arriving at a collective decision:

- Propose: Each person independently writes down a list of people whom she would like to join to form her own subgroup. She then sends an invitation to everyone on the list to form a group. Here we implicitly assume that a person makes an invitation to herself.

- Select: During the Propose phase, a person would receive a number of invitations from its neighbours. After all invitations are received, the person evaluates the quality of each proposal, then selects and accepts the best proposal.
- Adjust: Once a person accepts an invitation, she then updates her own decision according to the accepted proposal. After every individual finishes this step, the whole group would have been divided into a number of subgroups, and thus a clustering is formed.

More precisely, the crucial ingredients in the definition of a PSA cell consist of a set of proposals and a preference relation, which are defined as follows:

Definition 4 (Proposal and Preference). A *proposal* of $v \in V$ is a set $\mathscr{P}_v \subseteq N(v)$. Thus $2^{N(v)}$ denotes the set of all possible proposals of v. A *preference relation* $\preceq$ is a linear ordering defined on all finite graphs such that for any two graph G_1, G_2, $G_1 \prec G_2$ means G_1 is preferred over G_2.

Any implementation of the PSA system involves giving a precise definition of proposals for each cell, and a preference relation. In our bipartite graph market model, each consumer and each commodity will act as a PSA cell, which carries out computation based on its local information: For a consumer this information contains all commodities that consumer chooses; for a commodity, this information contains all consumers that choose this commodity.

Each cell performs three phases of computation repeatedly, which are roughly described below: In the Propose phase, every cell v prepares and announces a proposal $\mathscr{P}_v$ which is received by all nodes contained in the proposal. In particular, as the proposal $\mathscr{P}_v$ will automatically contain the node v itself, we also regard v as having received the proposal. Then in the Select phase, every cell v takes the collection of proposals it receives, and chooses a most preferred proposal according to the relation $\preceq$. Suppose v selects the proposal $\mathscr{P}_u$ proposed by u. In the Adjust phase, v observes the action of u and makes the two possible actions:

1. If u chooses its own proposal $\mathscr{P}_u$ (in this case, its own proposal is seen as the most preferred option), then v joins the group of u.
2. If u chooses the proposal $\mathscr{P}_w$ of some node $w \neq u$, then v does not join the group of u, and disregard the proposal from u.

After finishing the Adjust phase, every cell would go back to the Propose phase and restart the three phases again. This enables computation in a dynamic setting: Suppose the input data (in the form of edges) is dynamic, every cell would make spontaneous changes to recompute new proposals every time it reaches the Propose phase; then its neighbours will modify their selections and adjust their solutions correspondingly; such changes may cascade through the graph. Moreover, assuming the market data is stable (so no change occurs anymore), every PSA cell in this simulation will eventually reach a stable solution.

To implement a PSA system to solve the dual market segmentation problem on a bipartite graph, we must handle both the local and global segmentation steps. Therefore, we need one PSA system to simulate local segmentation of consumers

and commodities, and another PSA system to simulate global segmentation. Hence, we use two types of PSA cells: local PSA cells and global PSA cells.

1.3.2 *Local* PSA *Cells*

In a system $G = (V_0 \cup V_1, E)$, we compute the corresponding V_0-network (V_0, E_0) and V_1-network (V_1, E_1) as described above. In an undirected graph, a local core represents a set of nodes that are similar to each other. Therefore each cell $v \in V$ makes a proposal $\mathscr{P}_v = K(v)$. In the graph $G = (V, E)$, for any set $C \subseteq V$, we use $N(C)$ to denote the union $\bigcup_{u \in C} N(u)$. We utilise the following density measures:

- The *intra cluster density* of C is the percentage of the number of edges in C over all possible internal edges; in other words, we define

$$d_{intra}(C) = \frac{2 \times |E \restriction C|}{|C| \times (|C| - 1)} \tag{1.3}$$

- The *inter cluster density* is the percentage of the number of edges connecting C with an outside node over all possible links from C to outside nodes; in other words, we define

$$d_{inter}(C) = \frac{|(E \restriction N(C)) \setminus (E \restriction C)|}{|C| \times |N(C) \setminus C|} \tag{1.4}$$

Combining these two factors, we obtain the *utility* function d defined on a set C of nodes:

$$d(C) = d_{intra}(C) \times (\kappa(v) + 1) - d_{inter}(C) \times |C|$$

Note that intuitively, a set C has higher utility $d(C)$ if it is dense, has relatively large cardinality, and sparsely connected to nodes outside of it. To define the *preference* relation between proposals, we set $C_1 \preceq C_2$ whenever $d(C_1) \geq d(C_2)$.

As described above, every cell v makes a proposal $\mathscr{P}_v$ to its neighbours in the Propose phase. In this propose every cell would eventually receive a certain number of proposals. Each cell then anlayses all its received proposals and it chooses a most preferred proposal according to the preference relation μ defined above.

1.3.3 *Global* PSA *Cells*

After constructing the local segmentation, we obtain a new bipartite graph

$$G^H = \left(\mathscr{C}_0 \cup \mathscr{C}_1, E^H\right)$$

whose nodes are segments computed by the local PSA cells. We use another set of PSA cells to compute a global segmentation, where each cell stands for a local segment in $\mathscr{C}_0 \cup \mathscr{C}_1$. Similarly to the local perspective, the global perspective

also consists of a dual networks: $(\mathscr{C}_0, E_0^H)$ and $(\mathscr{C}_1, E_1^H)$. For each $C \in \mathscr{C}_i$ where $i \in \{0,1\}$, the proposal of C is $\mathscr{P}_C = \{C', C\}$ such that for all C_i $(C_i, C) \in E_i^H$ we have $\omega_i^H(\mathscr{C}', \mathscr{C}) \geq \omega_i^H(\mathscr{C}_i, \mathscr{C})$. Essentially, whenever two segments have maximum global Jaccard similarity, this process would form a larger segment by combining these two segments.

1.4 Experiment

1.4.1 Metrics for performance evaluation

We describe the basic setup for evaluating the proposed approach. Our data set represents a bipartite graph $(V_0 \cup V_1, E)$, to which we apply our PSA system. For any consumer $v \in V_0$, we use $\mathscr{C}_v$ to denote the set of commodities that v is attracted to. In other words,

$$\mathscr{C}_v = E(v) = \{u \in V_1 \mid \{v,u\} \in E\}. \tag{1.5}$$

For our experiments, we fix a percentage $p \in [0,1]$ and "hide" p percent of the edges in the data set. This results in a subgraph $(V_0 \cup V_1, E')$ where $E' \subseteq E$, called the *test data set*. The *known set* for the consumer v is $\mathscr{C}_v^{\mathsf{s}} = E'(v) = \{u \in V_1 \mid \{v,u\} \in E'\}$. Similarly, the known set for any commodity $u \in V_1$ is $\mathscr{C}_u^{\mathsf{s}} = E'^{-1}(u) = \{v \in V_0 \mid \{v,u\} \in E'\}$. For any $v \in V_0 \cup V_1$, the PSA system allocates the known set $\mathscr{C}_v^{\mathsf{s}}$ to the cell corresponding to u. The PSA system will produce, for any consumer $v \in V_0$ a segment $S(v) \subseteq V_0$ which contains v. We use $\mathscr{C}_v^{\mathsf{r}}$ to denote the set of commodities that are linked to all members of the segment $S(v)$. In other words,

$$\mathscr{C}_v^{\mathsf{r}} = \{u \mid \exists v' \in S(v) : \{u, v'\} \in E\}. \tag{1.6}$$

To validate our approach, we make the following assertion: In the ideal case, the market segments found by the approach should truthfully reflect consumers' interests in different products. This means, a desirable outcome of the market segmentation process, is that each consumer would have a strong interest in the products linked by all others in his own segment. This means that the desirable outcome is that $\mathscr{C}_v^{\mathsf{r}} = \mathscr{C}_v$. We will use this as a criterion for the validity of our PSA-based segmentation mechanism: We compare the set $\mathscr{C}_v^{\mathsf{r}}$ with the set $\mathscr{C}_v$ using three metrics: *precision*, *recall* and *successful deployment rate*. Precision and recall are two metric commonly used in data mining for demonstrating the performance of an automatic recommendation system [5].

- The precision of user v is defined as:

$$\mathsf{precision}(v) = \frac{|\mathscr{C}_v^{\mathsf{r}} \cap \mathscr{C}_v|}{|\mathscr{C}_v^{\mathsf{r}}|} \tag{1.7}$$

Intuitively, the value of $\mathsf{precision}(v)$ is the percentage of "correctly identified" commodities within the all "identified" commodities. It captures the amount of commodities chosen by a consumer within the computed segment of this consumer. Naturally, the higher its value, the higher the degree of relevance between v and the commodity segment $\mathscr{C}_v^{\mathsf{r}}$. In particular, the precision of a commodity segmentation should be higher than the its *random selection rate* $\frac{|\mathscr{C}_v|}{|V_1|}$ (V_1 is a set of all commodities), which is the value if $\mathscr{C}_v^{\mathsf{r}}$ is chosen randomly.

- The recall of v is defined as:

$$\mathsf{recall}(v) = \frac{|\mathscr{C}_v^{\mathsf{r}} \cap \mathscr{C}_v|}{|\mathscr{C}_v|} \tag{1.8}$$

The value of $\mathsf{recall}(v)$ is the percentage of "correctly identified" commodities within the total commodities "chosen by the consumer". Only considering precision may have some side effects. For example, in order to achieve high precision, a commodity may become conservative to include more commodities. Improving recall with segmentation of a commodity segmentation enforces it to include more attractive commodities. Similarly to precision, the *expected recall* is $\frac{|\mathscr{C}_v^t|}{|\mathscr{C}_v|}$.

- The successful deployment rate of v is:

$$\mathsf{sdr}(v) = \frac{|\mathscr{C}_v^r \cap (\mathscr{C}_v \setminus \mathscr{C}_v^t)|}{|\mathscr{C}_v \setminus \mathscr{C}_v^t|} \tag{1.9}$$

In other words, $\mathsf{sdr}(v)$ denotes the amount of products "unknown" to v that are linked to others in the same segments as v; This value represents how much "hidden" information can be "re-discovered" by the segmentation process. Hence a satisfying segmentation process should result in high values of $\mathsf{sdr}(v)$.

1.4.2 Data set

We take the Jester data set [4], which contains Internet users' ratings of a collection of jokes. In this setting, an online user is a consumer and a joke is the corresponding commodity. Segmentation of this data set should reveal users' different tastes on humour. We select one of the Jester data sets which contains 24,983 users and 101 jokes. Each user rates at least 36 jokes by giving an integer value ranging from -10 to 10. We further processed the data set by assuming that a user likes a joke if its rating is non-negative. We aim to use PSA system to identify segmentations of people with similar humour taste.

Our approach should be able to reveal whether a user would like a given joke. To test the validity of the approach, we pick a test data set from the original data set by keeping, for each user, only 80% of his rating of the jokes. We then apply our approach to process the test data set, which produces a number of segments. We test how much the results recovers the original rating (on all jokes).

Jester data set is an *imbalanced data set* because the number of consumers is much larger than the number of commodities. To understand the performance of our approach on *balanced data sets*, i.e., those where the number of consumers and number of commodities are both large, we design an experiment which involves a synthetic data set. In generating this synthetic data set, we use the following parameters:

- the number of consumers and commodities (σ_1, σ_2)
- the probability of selecting a segmentation in the dual network (σ_3)
- the probability of generating an edge to a node in a selected segmentation of the dual network (σ_4)
- the probability of generating an edge to a node in a segmentation which is not selected of the dual network (σ_5)

If two segmentations on the different sides of a dual network select each other, then the probability of generating edges between nodes in these two segmentations is high. Otherwise the probability should be low. Note that one segmentation has to select at least one segmentation and it also has a probability to select more than one segmentation. Our synthetic data set indeed makes sure that certain relationships are established between segmentations. In the experiment, we give all the edges to our system. Instead, our system needs to find all the segmentations and selected segmentations in the dual network. Our settings for a synthetic data set as follow:

- $\sigma_1 = \sigma_2 = 500$, $\sigma_3 = 0.5$, $\sigma_4 = 0.8$ $\sigma_5 = 0.2$
- $\alpha_0 = \alpha_1 = 0.7$, $\beta_0 = \beta_1 = 0.5$, $\beta_0 = \beta_1 = 0.6$

1.4.3 Experiment Environment

The PSA framework is designed to be implemented in a distributed environment where cells functions autonomously. Due to physical limitation, we perform computation in a single workstation, but implement a number of parallel processes, each simulating a group of cells. In our simulation, each cell only has to process a small amount of local information surrounding it. So, this parallel framework greatly improves computation time. We implement PSA system using the Java programming language as a multi-threaded application. To improve efficiency, we evenly distribute all the cells in a data set into four parallel threads. Each thread is responsible to execute PSA cells sequentially. This process results in faster computation comparing to sequentially executing of all PSA cells in series. The detailed specification of our experiment environment is: Windows 7 operating system, Intel Core i5-4570 quad-core CPU 3.2GHz and 16GB RAM.

1.4.4 Experimental Results and Discussion

We conduct four experiments on the Jester data set. We use PSA system to identify segmentations of users with similar tastes for jokes. In Experiment 1, we take 10 disjoint test data sets consisting of 1000 consumers (each set contains a disjoint set of consumers) and 101 commodities each. For any consumer, we retain 80% commodities it links to in the data set. We set $\alpha_0 = 0.35$, $\beta_0 = 0.3$, $\gamma_0 = 0.5$ and $\alpha_1 = 0.45$, $\beta_1 = 0.7$, $\gamma_1 = 0.2$. The results are shown in Figure 1.1.

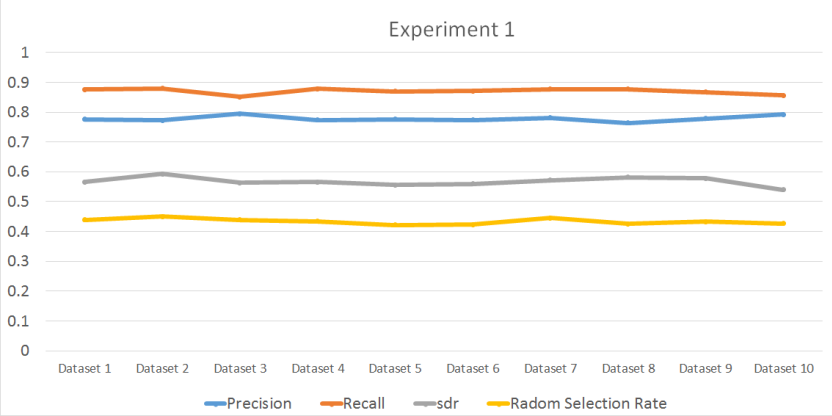

Fig. 1.1 Results of 10 disjoint test data sets with 20% hidden links.

The Jester data set is imbalanced as the number of users greatly exceeds that of times. So we have to configure different parameters for each dual network. The random selection rate result shows that, in general, our approach has more than 40% chance to correctly compute the segment of a consumer than making random guesses. The result shows that our approach improves the precision to 80%. In addition, it finds almost 60% of hidden commodities.

In *Experiment 2*, our goal is to analyse how the data set quality influences the results. We take consumers in *Dataset 1* from *Experiment 1* and generate 10 test data sets by varying the amount of links that are hidden from the test data set. The results show that with a higher percentage of deleted links, it is more difficult to compute correct segments. The results of *Experiment 2* are shown in Figure 1.2.

Precision describes how less mistakes that a system makes. When we reduce the percentage of given commodities, our system starts to lose capability of finding correlated segmentations within current parameter configuration. Hence, it becomes conservative to make any segmentations but only deploys given commodities. This experiment only shows influences caused by quality of data sets. We keep the pa-

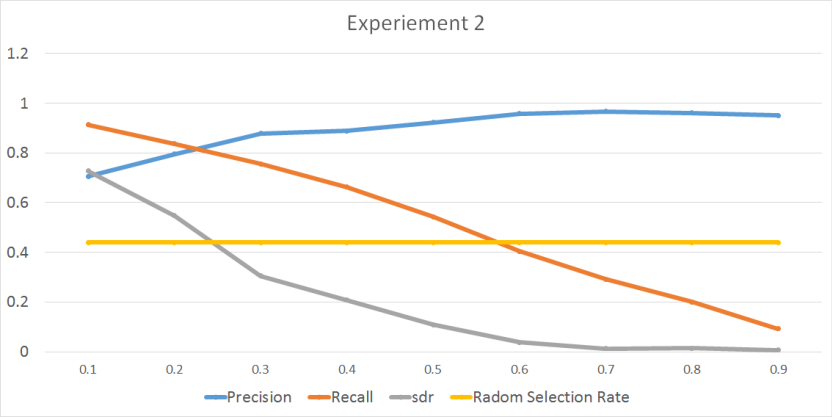

Fig. 1.2 Results of 10 test data sets with varying hidden links

rameter configuration from *Experiment 1*. In practice, we should reconfigure system parameters while dealing with different data sets.

In *Experiment 3*, we generate 20 test data sets with size varying from 100 to 2000. We keep the same configuration as in *Experiment 1*. The results are in Figure 1.3. As shown from the figures, the performance of our approach isn't affected much by the size of the data sets. In *Experiment 3*, fluctuation of sdr is stronger than other metrics. However, by increasing the data set size, sdr becomes flatter. Prediction of sdr is hard. As shown in *Experiment 2*, a data set should not include many unknown information otherwise segmentations will be inaccurate. Correlating segmentations in a dual network becomes harder when one side of the dual network has much less segments than the other side.

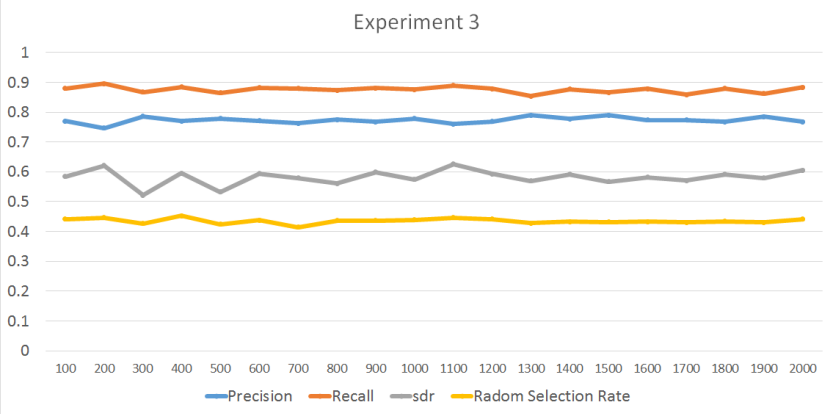

Fig. 1.3 Results of 20 test data sets with different size

In *Experiment 4*, we compare our solution with some other existing approaches for market segmentation over the same data sets. The problem is that some approaches such as k-means, x-means and hierarchical clustering require predetermined numbers of segmentations, which different from our approach where no prior knowledge to the number of segments is needed. Therefore we only compare our solution to algorithms that can compute with an unknown number of segments. The expectation maximization algorithm in WEKA uses cross validation to determine the number segmentations in a data set. Our PSA system determines the number of segmentations on acceptances of different proposals. The results of *Experiment 4* are given in Figure 1.4. EM clustering has high recall and sdr because it deploys all commodities to every consumer segmentation. It is only slightly better if a system randomly guesses the result. The advantage of PSA system is to dynamically segmenting consumers according to commodity segmentation. A proposal captures local information regarding to what requires in a segmentation.

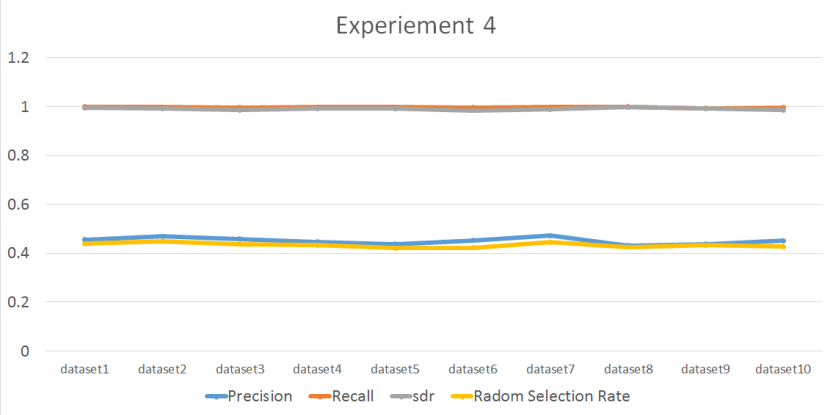

Fig. 1.4 Computing the results of our PSA-based approach with other approaches

Due to the mentioned inbalanceness of Jester data set, we further validate our approach in a balanced network where the number of consumers roughly equals to the number of commodities. To make the synthetic data more realistic, we includes certain noise data, so that certain consumers or commodities in a designated segment have an usual low number of links. Such noise does not affect the experimental results as our system still finds more hidden segments while keeping a high value of precision and recall. Our approach achieves an overall good performance on this synthesised data:

precision	recall	sdr	random selection rate
0.87466	0.939716	0.981968	0.655504

1.5 Related works

Huang et al. implements a market segmentation mechanism based on support vector machines (SVM) [6]. The authors conduct experiments on a drink company data set and demonstrate how SVM may outperform in terms of forming better clusterings than other techniques such as the *k*-means algorithm. In [14], Wang adopts another clustering techinque, namely robust fuzzy clustering. He improves the original algorithm by pre-processing data set so that noise data can be eliminated. This results in certain improvements in terms of quality of clusters. Migueis et al. use a variable clustering algorithm to determine consumer segmentations from purchase history dataset in [12]. The authors assign consumers to each segments to illustrate purchasing power of customers. Deng et al. analyse market segmentation in mobile e-commerce of a Japanese chain restaurant in [2]. They propose a hybrid clustering framework that combines *k*-means algorithm, self-organising feature mapping and particle swarm optimisation. They identified over 70% observed customer segmentations; and each individual approach in the hybrid approach performs less accurate. Kuo et. al investigate different combinations of a set of algorithms in [7]. The authors conduct experiments on data collected from survey about web user segmentations and evaluate their results through a business marketing analysis.

1.6 Conclusion and Future Work

This paper proposes a new approach that is based on the PSA-framework to simulate and solve the dual market segmentation problem. Our experimental results shows that on real data sets our approach may find reasonable segments while improving from existing approaches. Furthermore, as the approach is implemented in a distributed framework, it may be used on markets where data is maintained dynamically and detect the changing patterns of market segments.

As future works, there are several possible ways to extend our current work:

- Firstly, one could explore different models for implementing the PSA-system in our approach. In particular, one could use different similarity functions, utility functions and proposal functions and test their impact on the performance of the method.
- Secondly, more tests could be performed on real-world and artificial data. In particular, we would need data sets that contains a rather sophisticated market that contains a large number of consumers and a large number of products of different ranges. Instead of using data mining techniques such as cross validation, we could compare the identified market segments with real world ground-truth.
- Thirdly, the challenging aspect for our current approach is that we have to manually configure parameters such as different threshold in our method. To facilitate real automated market segmentation, it would be necessary to implement a mechanism that allows automated parameters configuration to best fit the data sets.

References

1. Campbell, C. (2004), "I Shop Therefore I Know That I Am: The Metaphysical Foundations of Modern Consumerism", in Elusive Consumption, Karin Ekstrom and Helen Brembeck (eds.), Oxford: Berg, 10–21.
2. Deng, X. Y., Jin, C., Higuchi, Y., & Han, J. C. (2011). An efficient hybrid clustering algorithm for consumer segmentation in mobile e-commerce. ICIC Express Letters, 5(4B), 1411–1416.
3. Dutta, S., Bhattacharya, S., & Guin, K. K. (2015, January). Data Mining in Market Segmentation: A Literature Review and Suggestions. In Proceedings of Fourth International Conference on Soft Computing for Problem Solving (pp. 87–98). Springer India.
4. Goldberg, K., Roeder, T., Gupta, D., & Perkins, C. (2001). Eigentaste: A constant time collaborative filtering algorithm. Information Retrieval, 4(2), 133-151.
5. Davis, J., & Goadrich, M. (2006). The relationship between Precision-Recall and ROC curves. In Proceedings of the 23rd international conference on Machine learning (pp. 233-240). ACM.
6. Huang, J. J., Tzeng, G. H., & Ong, C. S. (2007). Marketing segmentation using support vector clustering. Expert systems with applications, 32(2), 313–317.
7. Kuo, R. J., Chang, K., & Chien, S. Y. (2004). Integration of self-organizing feature maps and genetic-algorithm-based clustering method for market segmentation. Journal of Organizational Computing and Electronic Commerce, 14(1), 43–60.
8. Li, Z., Wei, Z., Jia, W., & Sun, M. (2013, July): Daily life event segmentation for lifestyle evaluation based on multi-sensor data recorded by a wearable device. In Engineering in Medicine and Biology Society (EMBC), 2013 35th Annual International Conference of the IEEE (pp. 2858–2861). IEEE.
9. Liu, J., & Wei, Z. (2014). From a Local to a Global Perspective of Community Detection in Networks. In PRICAI 2014: Trends in Artificial Intelligence (pp. 1036-1049). Springer International Publishing.
10. Liu, J., & Wei, Z. (2014, December). Community Detection Based on Graph Dynamical Systems with Asynchronous Runs. In Computing and Networking (CANDAR), 2014 Second International Symposium on (pp. 463–469). IEEE.
11. Miguéis, V. L., Camanho, A. S., & e Cunha, J. F. (2011). Mining consumer loyalty card programs: The improvement of service levels enabled by innovative segmentation and promotions design. In Exploring Services Science (pp. 83–97). Springer Berlin Heidelberg.
12. Miguéis, V. L., Camanho, A. S., & e Cunha, J. F. (2012). consumer data mining for lifestyle segmentation. Expert Systems with Applications, 39(10), 9359–9366.
13. Tuner, J. C. (1987). Redicovering the social group: a self-categorization theory. Oxford: Blackwell (pp. 42–67).
14. Wang, C. H. (2010): Apply robust segmentation to the service industry using kernel induced fuzzy clustering techniques. Expert Systems with Applications, 37(12), 8395–8400.