Sentiment and Preference Guided Social Recommendation

Yoke Yie Chen[1], Xavier Ferrer[2,3], Nirmalie Wiratunga[1], and Enric Plaza[2]

[1] IDEAS Research Institute, Robert Gordon University, Aberdeen, Scotland
{y.y.chen,n.wiratunga}@rgu.ac.uk
[2] Artificial Intelligence Research Institute (IIIA-CSIC)
Spanish National Research Council (CSIC)
Campus UAB, Bellaterra, Catalonia, Spain
[3] Universitat Autònoma de Barcelona
Bellaterra, Catalonia, Spain
{xferrer,enric}@iiia.csic.es

Abstract. Social recommender systems harness knowledge from social experiences, expertise and interactions. In this paper we focus on two such knowledge sources: sentiment-rich user generated reviews; and preferences from purchase summary statistics. We formalise the integration of these knowledge sources by mixing a novel aspect-based sentiment ranking with a preference ranking. We demonstrate the utility of our proposed formalism by conducting a comparative analysis on data extracted from Amazon.com. In particular we show that the performance of the proposed aspect based sentiment analysis algorithm is superior to existing aspect extraction algorithms and that combining this with preference knowledge leads to better recommendations.

Keywords: social recommender systems, sentiment analysis, aspect extraction, preference graph.

1 Introduction

Recommender systems have traditionally relied on improving their ranked lists by exploiting knowledge about user preferences [26], their information needs [11] or by exploiting similar behavior of other users [1]. The huge success of these systems in the retail sector is also its main driving force towards innovative and improved recommendation algorithms. Representation, similarity and ranking algorithms from the Case-Based Reasoning (CBR) community has naturally made a significant contribution to recommender systems research [18, 23]. The dawn of the social web creates many new opportunities for recommendation algorithms and so the emergence of social recommender systems [9, 12].

Consider a typical product recommendation scenario on Amazon (see Figure 1 and Figure 2). Here in addition to typical information about an artefact (e.g. Camera image and textual description), there is also information generated or derived from user interactions (e.g. reviews and what users actually buy after

viewing this camera). Specifically we observe that there is explicit knowledge in the form of user opinion and implicit knowledge in the form of user preferences. Here by preference we are referring to preference of users over viewed products. Opinion knowledge is often used to improve recommendations by incorporating the sentiment expressed in opinions to bias the retrieved list [9]. Similarly preference knowledge has also separately been applied to successfully influence recommendations [26]. Our focus in this paper is to harness social content by uniting both these knowledge sources to generate better recommendation rankings.

Our contribution is three-fold: firstly we demonstrate how the choice of sentiment analysis algorithms can impact the quality of recommendations; and secondly show how a page-rank type algorithm can be effortlessly incorporated to derive rankings on the basis of preference relationships; and finally provide a formalism to combine sentiment and preference knowledge. Our results confirm that aspect-based sentiment analysis is far superior to one that is agnostic of aspects. This is because purchase choices are based on comparison of artefacts; which implicitly or explicitly involves comparison of characteristics or aspects of these artefacts. Lastly, we introduce a novel algorithm to infer aspect importance weights guided by knowledge represented in a preference graph. Here the insight is that preference relationships modeled in this graph allow us to identify aspects that are likely to have influenced the users' purchase decision. In particular a users purchase decision hints at the aspects that are likely to have influenced their decision and as such be deemed more important.

The rest of the paper is organized as follows: In Section 2 we present the background research related to this work. Next in Section 3 we describe how preference graphs can be generated by using a case study from Amazon.com. The process of aspect extraction and weight learning for sentiment analysis is presented in Section 4. Finally, our evaluation results are presented in Section 5 followed by conclusions in Section 6.
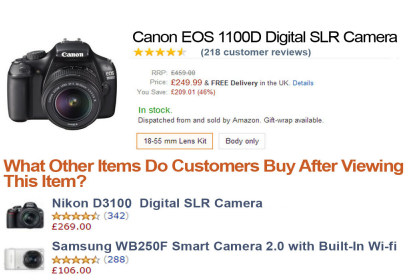

Fig. 1. Product information (explicit)

Fig. 2. Product reviews (implicit)

2   Related Work

There exist numerous applications of machine learning and CBR in the area of recommendation systems. Content-based filtering approaches exploit the past and current preferences of the user to build new recommendations. Early work in this area has analysed user logs and sessions [5,17]; reused similar user trails [2, 24]; and exploited click-through data [6] to improve recommendations.

Collaborative filtering [15] unlike content-based filtering emphasises the social dimensions of user similarities, preferences and experiences. A good example is Amazon.com, where user ratings and preferences are combined in different ways [16]. In Gupta and Chakraborti [13], graphs are used to model user decisions and combine estimated utilities in the context of preference based recommenders. In Vasudevan and Chakraborti [26], preference models of artefacts are induced based on the trails left by users when critiquing an artefact's aspect. In our work, we do not rely on a preference graph that explicitly defines which aspects are preferred, but instead infer this information from comparing the sentiment-rich content generated by users. Furthermore, unlike other preference based recommender systems which utilise user profile to generate user preferences, we obtained preference knowledge from user interactions. As such we provide a more flexible alternative to preference analysis.

Social recommenders recognise the important role of sentiment analysis of user reviews [9]. Extracting sentiment from natural language constructs is a challenge. Lexicons are often used to ascertain the polarity (positive or negative) and strength of sentiment expressed at word-level (e.g. SentiWordNet [10]). However sophisticated methods are needed to aggregate these scores at the sentence, paragraph and document level to account for negation and other forms of sentiment modifiers [21]. Increasingly aggregation is organised at the aspect level, since the distribution of a user's sentiment is typically mixed and expressed over the aspects of the artefact (e.g. I love the *colour* but not too keen on *size*). Hu and Liu [14] propose an association mining driven approach to identify frequent nouns or noun phrases as aspects. Thereafter sentences are grouped by these aspects and sentiment scores assigned to each aspect group. Whilst there are many other statistical approaches to frequent noun extraction [3,14,19,22]; others argue that identifying semantic relationships in text provides significant improvements in aspect extraction [20]. Here we explore how semantic based extraction can be augmented by frequency counts.

3   Social Recommendation Process

An overview of our proposed process appears in Figure 3. The final outcome is a recommendation of products that are retrieved and ranked on their product rank score, *ProdRank*, with respect to a given query. Central to this ranking is the integration of sentiment scores derived from user reviews with dominant products inferred from the preference graph.

$$ProdRank(p_i) = \alpha * PrefRank(p_i) + (1 - \alpha) * SentiRank(p_i) \qquad (1)$$
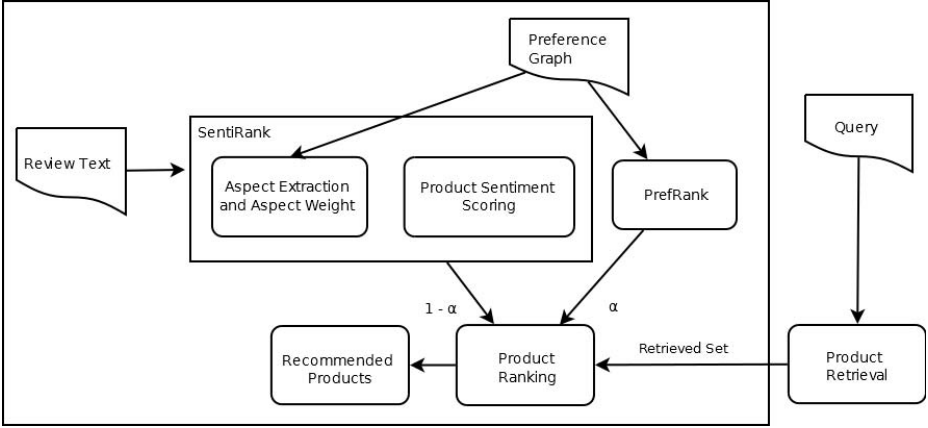
Fig. 3. Overview of social product recommender process

Where *PrefRank* assigns a score to product, p_i, by applying PageRank [4] to a preference graph; and *SentiRank* assigns a score based on aspect level sentiment analysis. The graph is generated from viewed and purchased product pairs; whilst the sentiment scores are derived from product reviews. We also advocate the use of weighted aspect level sentiment analysis and learn these weights by comparing the sentiment difference between node pairs in the graph (see Section 4). Here α is used as a mixture parameter to study the impact of preference-only ($\alpha = 1$), sentiment-only ($\alpha = 0$), or a mixture of both ($1 < \alpha < 0$).

3.1   Preference-Based Product Ranking

A preference relation between a pair of products denotes the preference of one product over the other through the analysis of viewed and purchased product relationships. To illustrate the generation of preference relation from Amazon dataset, consider a snapshot of Amazon product web page in Figure 1. Here *Canon EOS 1100D Digital SLR Camera* (*CanonSLR*) is the viewed product. At the bottom of this page, we can observed that *Nikon D3100 Digital SLR Camera* (*NikonSLR*) and *Samsung WB250F Smart Camera* (*SamsungSmart*) are the products that some of the users purchased after viewing *CanonSLR*. This list of purchased products hints at the preference of users. Using these information, we generate two preference relations in which *NikonSLR* is preferred over *CanonSLR* and *SamsungSmart* is preferred over *CanonSLR*. A preference graph, $G = (\mathcal{P}, \mathcal{E})$, is generated from such product relation (see Figure 4). The set of nodes, $p_i \in \mathcal{P}$, represent products, and the set of directed edges, $\mathcal{E}$, are preference relations, $p_j \succ p_i$, such that a directed edge from product p_i to p_j with $i \neq j$ represents that, for some users, p_j is preferred over product p_i. In some cases where $p_j \succ p_i$ and $p_i \succ p_j$, a bidirectional preference relation can be observed. For any p_i, we use $\mathcal{E}^i$ to denote in-coming and $\mathcal{E}_i$ for outgoing product sets.
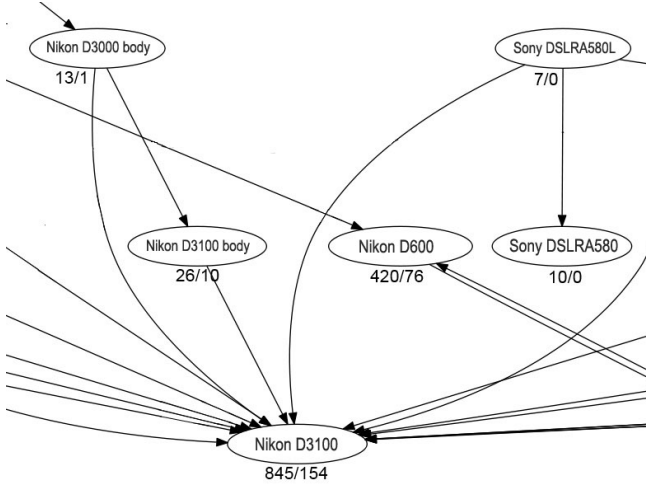
Fig. 4. Preference sub-graph for *Amazon Digital SLR Cameras*

Figure 4 illustrates a preference graph generated from a sample of Amazon data on *Digital SLR Camera*. The number of reviews/questions for a product is shown below each product node. Typically top ranked products have many more incoming edges, while less popular products tend to have more outgoing edges. It is not surprising that product *Nikon D3100* in Figure 4 is listed in Amazon's top 10 *Best Seller* ranking. Based on these observations, the higher the number of incoming edges (quantity) from preferred products (quality), the higher the preference rank, *PrefRank*, is for product, p_i.

$$PrefRank(p_i) = \sum_{p_j \in \mathcal{E}^i} \frac{PrefRank(p_j)}{|\mathcal{E}_j|} \qquad (2)$$

Where $\mathcal{E}^i$ is the set of all viewed products over which p_i is preferred and $\mathcal{E}_j$ is the set of products that are preferred after viewing p_i.

The main assumption here is similar to PageRank where the score captures the importance of pages by the analysis of the quantity and quality of edges to pages [4]. However we also observed that while our assumption is true with most studied products, it is not always the case that a product with higher *PrefRank* will also have a higher rank in Amazon's *Best Seller*. For example, Amazon's third ranked camera has a higher *PrefRank* score compared to the second ranked camera (see Table 1). Similar observations also hold when comparing *PrefRank* scores with user generated overall product ratings. This motivates the need to leverage on further dimensions of knowledge sources such as sentiment from product reviews for product recommendation.

Table 1. Summary of the averages of products organised by Amazon Best Seller rank

Amazon Bestseller Ranking	PREFERENCE				SENTIMENT		
	PrefRank	Incoming Edges	Outgoing edges	#Reviews per Day	*SentiRank* aspects	#Reviews	Product Rating
1	0.1240	26.0	1.0	1.136	0.087	1351.0	4.7
2	0.0037	2.0	2.0	0.770	0.106	242.0	4.7
3	0.0178	10.0	0.0	0.621	0.097	847.0	4.6
4-5	0.0053	1.5	0.5	0.589	0.102	129.0	4.8
6-10	0.0435	5.5	1.25	0.453	0.089	354.75	4.72
11-50	0.0048	2.071	0.892	0.506	0.080	407.35	4.45
51-100	0.0021	0.529	1.647	0.366	0.124	393.64	4.52
N/A	0.0015	0.439	1.469	0.143	0.092	175.34	4.61

4   Aspect Weighted Sentiment-Based Product Ranking

Users author reviews following the purchase of products. These contain user opinion in the form of positive and negative sentiment. Strength of sentiment expresses the intensity with which an opinion is stated with reference to a product [25]. We exploit this information as a means to rank our products, such that products ranked higher denote higher positive sentiment. *SentiRank* of a product, p_i, given a set of related reviews $\mathcal{R}^i$ is computed as follows:

$$SentiRank(p_i) = \frac{\sum_{k=1}^{|\mathcal{R}^i|} SentiScore(r_k)}{|\mathcal{R}^i|} \tag{3}$$

Here *SentiScore*s are generated by the SMARTSA system [21] for each $r_k \in \mathcal{R}^i$. The SMARTSA system obtains the sentiment score of sentiment-bearing words from SentiWordNet [10]. The score will be modified to take into consideration of negation terms and lexical valence shifters (e.g. intensifier and diminish terms). The negative and positive strength is expressed as a value in the range [-1:1].

A finer-grained analysis of reviews is achieved by computing sentiment at the aspect level. It allows the sentiment of product, p_i, to be associated with individual aspects $a_j \in \mathcal{A}^i$ where $\mathcal{A}^i \subseteq \mathcal{A}$. The aspects of a product are extracted by using Algorithm 1. Accordingly we have the following rewrite for *SentiRank*.

$$SentiRank(p_i, a_j) = \frac{\sum_{j=1}^{|\mathcal{A}^i|} AspectWeight(a_j) * AspectSentiScore(p_i, a_j)}{|\mathcal{A}^i|} \tag{4}$$

This new formalisation is a weighted summation of sentiment expressed at the aspect level. Once aspects are extracted the sentiment of a product's reviews can be expressed as a distribution over each aspect. Accordingly, the aspect-level sentiment score is:

$$AspectSentiScore(p_i, a_j) = \frac{\sum_{m=1}^{|\mathcal{R}_j^i|} SentiScore(r_m)}{|\mathcal{R}_j^i|} * (1 - Gini) \qquad (5)$$

Where $\mathcal{R}_j^i$ is a set of reviews for product p_i related to aspect a_j and $r_m \in \mathcal{R}_j^i$. We use the Gini coefficient [27] to assign higher sentiment scores to an aspect when there is consensus about the distribution of the sentiment and otherwise is penalised accordingly.

A product purchase choice is a preference made on the basis of one or more aspects. The notion of aspect importance arises when the same set of aspects contribute to similar purchase decisions. Using this same principle, aspects weights are derived by comparing the aspect sentiment score differences between purchased and viewed product pairs in which $(p_x, p_y) \in \{(p_x, p_y)\}_{x,y=1 \wedge x \neq y}^{t}$

$$AspectWeight(a_j) = \frac{\sum_{x=1}^{|\mathcal{P}|} \sum_{y=1}^{|\mathcal{P}|} \delta(a_j, p_x, p_y)}{|t \in \mathcal{E}|} \qquad (6)$$

where either $p_x \succ p_y$ or $p_y \succ p_x$ or both, and t is the set of product preference pairs containing aspect a_j. The preference difference between any pairs of products is computed as:

$$\delta(a_j, p_x, p_y) = |L_{min}(\mathcal{A}, \mathcal{E})| + \delta'(a_j, p_x, p_y) \qquad (7)$$

$$\delta'(a_j, p_x, p_y) = (AspectSentiScore(a_j, p_x) - AspectSentiScore(a_j, p_y)) \qquad (8)$$

Here $|L_{min}(\mathcal{A}, \mathcal{E})|$ is the lowest preference difference scores obtained over all the aspect and product preference pairs. This is required to avoid negative aspect weights while preserving the importance of the aspects.

4.1 Sentiment Aspect Extraction

Grammatical extraction rules [20] are used to identify a set of candidate aspect phrases from sentences. These rules operate on dependency relations in parsed sentences[1]. Figure 5 lists the rules that we have employed in this work. Here N is a noun, A an adjective, V a verb, h a head term, m a modifier and $\langle h, m \rangle$ is a candidate phrase. Examples of how these rules apply to sample sentences appear in Table 2. Consider the first example sentence which according to Algorithm 1 applies to rule three: $cop(good, is) + nsubj(good, lens) \rightarrow \langle lens, good \rangle$. Next, if a Noun Compound Modifier (nn) exists in the sentence, rules five and six apply; and in this example rule five applies resulting in the following candidate aspects: $(lens, good) + nn(lens, camera) \rightarrow \langle camera\ lens, good \rangle$. In this way given a set of reviews a set of candidate phrases are extracted. For each candidate, non noun (N) words are eliminated. Thereafter frequency of each candidate is calculated

---

[1] Sentences are parsed using the Stanford Dependency parser [7].

according to its N and NN phrase; retaining only those candidates above a frequency cut-off (e.g. greater than 1% of the maximum frequency occurrence of a noun).

DP = set of dependency pattern rules
$\{$

$\quad dp_1 : amod(N, A) \rightarrow \langle N, A \rangle,$

$\quad dp_2 : acomp(V, A) + nsubj(V, N) \rightarrow \langle N, A \rangle,$

$\quad dp_3 : cop(A, V) + nsubj(A, N) \rightarrow \langle N, A \rangle,$

$\quad dp_4 : dobj(V, N) + nsubj(V, N') \rightarrow \langle N, V \rangle,$

$\quad dp_5 : \langle h, m \rangle + nn(h, N) \rightarrow \langle N + h, m \rangle,$

$\quad dp_6 : \langle h, m \rangle + nn(N, h) \rightarrow \langle h + N, m \rangle$

$\}$

Fig. 5. Extraction rules

Algorithm 1. Aspect Selection by Dependency Patterns (FQDPRULES)

1: **INPUT:** S = sentences
2: **for all** s_j **in** S **do**
3: $\quad g = grammaticalRelations(s_j)$
4: $\quad candidateAspects = \{\}$
5: $\quad$ **for** $dp_i \in DP$ and $1 \le i \le 4$ **do**
6: $\quad\quad$ **if** $g.matches(dp_i) \wedge g.contains(nn)$ **then**
7: $\quad\quad\quad candidateAspects. \leftarrow g.apply(\{dp_5, dp_6\})$ $\qquad \triangleright$ Apply rules dp_5, dp_6
8: $\quad\quad$ **end if**
9: $\quad$ **end for**
10: $\quad aspects \leftarrow candidateAspects.select(N, NN)$ $\qquad \triangleright$ select nouns and compound nouns
11: **end for**
12: $filterByFrequency(aspects)$ $\qquad \triangleright$ ignore low frequency aspects
13: **return** $aspects$

4.2 Aspect Weight Extraction

Learning of aspect weights relies on preference graph and aspect level sentiment knowledge. Figure 6 and 7 illustrates the notion of preference difference calculations using a trivial three node preference graph. In Figure 6, the relation $p_3(+0.1) \succ p_1(+0.8)$ denotes product p_3 is preferred over p_1 and they have an aspect sentiment score of +0.1 and +0.8 respectively for aspect *lens* (see Equation 5). Corresponding preference difference scores are also shown in Table 3 for two aspects. Here *lens* and *screen* have a normalised aspect weights of 0.03 and 0.97 respectively. Therefore, we suggest that aspect *screen* is more important than aspect *lens*.

Table 2. Definition of grammatical relations

Example	Grammatical relations	Dependencies
The camera lens is good	nsubj (Nominal Subject)	nsubj(good, lens)
	cop (Copula)	cop(good, is)
	nn (Noun Compound Modifier)	nn(lens, camera)
The screen is bright with nice colors	amod (Adjectival Modifier)	amod(colors, nice)
She looks amazing	acomp (Adjectival Complement)	acomp(looks, amazing)
I like the camera lens	dobj (Direct Object)	dobj(like, lens)
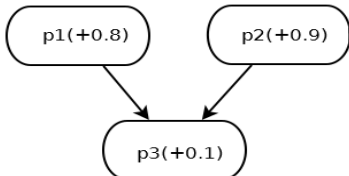
Fig. 6. Sub-graph *lens* aspect
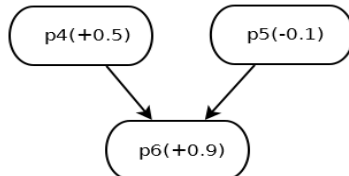
Fig. 7. Sub-graph *screen* aspect

Table 3. Aspect preference scores

Aspects	Preference Relations	Preference Difference Scores	Normalised Aspect Weights
lens	$p_3 \succ p_1$ $p_3 \succ p_2$	$= 0.1$ $= 0.0$	$\frac{0.1}{3.1} = 0.03$
screen	$p_6 \succ p_4$ $p_6 \succ p_5$	$= 1.2$ $= 1.8$	$\frac{3.0}{3.1} = 0.97$

5 Evaluation

In this section we evaluate our proposed integration of sentiment and preference knowledge applied to product recommendation. Since the quality of extracted aspects in sentiment analysis will have a direct impact on the quality of recommendations; we first conduct a pilot study to evaluate the quality of aspects extracted by our algorithm with the state-of-the-art. Thereafter, we evaluate how well the recommendation system works in practice on Amazon.com data using two derived benchmark rankings. Here we are keen to study the complimentary roles of sentiment and preference knowledge and expect to discover a synergistic relation between them.

5.1   Comparative Pilot Study - Aspect Extraction Analysis

We use a public dataset on product reviews containing manually marked-up product aspects [8, 14]. For this study we use phone category products with at least hundred reviews. Precision and recall is used to compare manually labeled aspects with extracted ones using the following alternative extraction algorithms:

– FQITEMS uses shallow NLP to identify single nouns as candidate aspects that are then pruned using a frequency cut-off threshold.
– FQPOS uses Part-of-Speech(POS) extraction patterns that are then pruned using sentiment informed frequency cut-off threshold [9]
– DPRULES uses the dependency extraction rules in Figure 5 [20].
– FQDPRULES same as DPRULES but prunes candidate aspects using a frequency cut-off (See Algorithm 1).

Table 4. Results for aspect extraction pilot study

Approach	Precision	Recall	F-score
FQITEMS	0.60	0.34	0.43
FQPOS	0.71	0.11	0.20
DPRULES	0.28	0.66	0.39
FQDPRULES	0.76	0.25	0.37

Precision of all frequency based extraction approaches are significantly better compared to DPRULES (see Table 4). We also confirm that FQPOS improves over FQITEMS. As expected best results are achieved with FQDPRULES when deep NLP semantics is combined with frequency prunning. Here we observe a 26% and 7% improvement in precision over FQITEMS and FQPOS respectively. Recall trends suggests that FQDPRULES must have many false negatives and so missed extraction opportunities compared to FQITEMS and FQPOS. However a lower precision is more damaging as it is likely to introduce aspect sparsity problems which have detrimental effect on sentiment difference computations. Therefore on the basis of these results we use FQDPRULES to extract product aspects for sentiment analysis in the social recommender experiments in the next sections.

5.2   Amazon Dataset

We crawled 2264 Amazon products during April 2014. From this we use the *Digital SLR Camera* category containing more than 20,000 user generated reviews. Since we are not focusing on the cold-start problem, newer products and those without many user reviews are removed. Here we use 1st January 2008 and less than 15 reviews as the prunning factor for products. Finally, any synonymous products are united leaving us data for 80 products (see Table 5).

The FQDPRULES algorithm extracted 981 unique aspects and on average 128 different aspects for each product. Importantly more than 50% of the products shared at least 70 different aspects, while 30% shared more than 90 aspects on average. The fact that there are many shared aspects is reassuring for product comparison.

Table 5. Amazon Digital SLR Camera dataset

#Products	2,264
#Products (filtered)	80
#Reviews (filtered)	21,034
#Aspects Mean (Std. Dev.)	128.66 (43.84)
#Different aspects	981

5.3 Ranking Strategies

The retrieval set of a query product consists of products that share a similar number of aspects. This retrieval set is ranked using the following sentiment-based recommendation strategies in which the formalisations were presented in Section 4:

- BASE: recommend using sentiment in reviews (see Equation 3);
- ASPECTG: recommend using aspect sentiment analysis (see Equation 5);
- ASPECT: same as ASPECTG but without the *Gini coefficient* weighting in Equation 5; and
- ASPECTG*: same as ASPECTG but with the additional aspect weighting component in Equation 4.

We also present each of the above strategies in combination with preference knowledge (see Equation 2). Finally, the impact of increasing α values is analysed according to Equation 1 to study the relationship between sentiment and preference knowledge.

5.4 Evaluation Metrics

In the absence of a manual qualitative estimate of recommendation or access to user specific purchase trails, we derived approximations from the Amazon data we had crawled. For this purpose, using a *leave-one-out* methodology, the average gain in rank position of recommended products over the left-out query product is computed relative to a benchmark product ranking.

$$\%RankGain = \frac{\sum_{i=1}^{n=3} benchmark(P_q) - benchmark(P_i)}{n * |\mathcal{P} - 1|} \tag{9}$$

where n is the size of the retrieved set and *benchmark* returns the position on the benchmark list. The greater the gain over the query product the better. Suppose the query product is ranked 40th on the benchmark list of 81 unique products $\mathcal{P}$, and the recommended product is ranked 20th on this list, then the recommended product will have a relative benchmark rank improvement of 25%.

We generate two benchmark lists according to the following dimensions:

- *Star-Rating*: Use the Amazon star rating. However, this benchmark is static and tends to be less reliable as it does not necessarily represent the current trends about products. For example, the LCD display on a camera may have been a good reason for a high star rating in the past whilst now almost every camera possesses it.
- *Popular*: Derived from Amazon's reviews, questions and timeline data.

$$Popular(p) = \frac{nReviews + nQuestions}{days\_online} \tag{10}$$

where *nReviews* and *nQuestions* refer to the number of reviews and questions of a product respectively, and *days_online* is the number of days the product has been on Amazon's website. We found that this formula has some correlation with the top 100 Amazon Best Seller ranking (*Spearman correlation* of -0.4381). Unlike Amazon's benchmark this allows us to experiment with query products that may not necessarily be in the Top 100[2].

5.5   Results

The graphs in Figures 8 and 9 show for increasing numbers of shared aspects the performance of each algorithm in terms of the average *%RankGain* on benchmarks Popular and Star-Rating. In general ASPECT and ASPECTG perform best, with both strategies recommending products with an average rank gain of close to 15% compared to BASE with no aspect extraction. It is worth pointing out that the improvement with applying the *Gini coefficient* is approximately 5% more on average when compared to not using it.
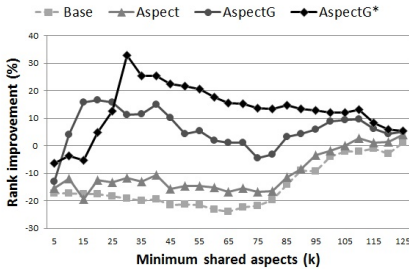

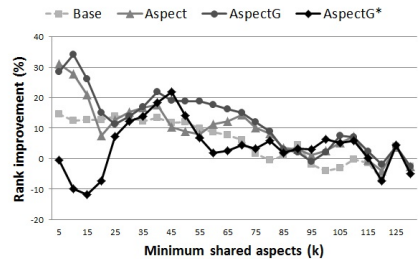**Fig. 8.** Benchmark Popular   **Fig. 9.** Benchmark Star-Rating

Results when using aspect weights with ASPECTG* is mixed and seems dependent on the benchmark used. For instance it does very poorly on the static Star-Rating benchmark yet is the winning algorithm on the Popular benchmark. In fact it achieves up to 30% gain (with 30 shared aspects or more with the query product) on the Popular benchmark. One explanation for its poor performance on Star-Rating might be explained by the fact that user preferences

---

[2] http://www.amazon.co.uk/Best-Sellers-Electronics-Digital-SLR-Cameras

Table 6. Preferred aspects for Digital SLR Cameras in Amazon

	Top 5 Popular Aspects			Bottom 5 Popular Aspects	
Aspect	**Weight** (normalised)	**Frequency** (No. of products)	**Aspect**	**Weight** (normalised)	**Frequency** (No. of products)
shutter	0.00229	61	deal	0	67
photography	0.00181	70	lcd	$8.56 \cdot 10^{-5}$	59
point	0.00179	70	control	$2.06 \cdot 10^{-4}$	66
system	0.00176	54	day	$2.92 \cdot 10^{-4}$	70
video	0.00166	68	iso	$3.53 \cdot 10^{-4}$	67

about camera aspects may have changed during the period of 2008-2014. Since aspect weights are learnt on the basis of preference knowledge it may well be that similar information is implicitly influencing the Popular benchmark.

Table 6 shows the top 5 most and less preferred aspects together with their weights and frequency of appearance in products. Note that the sum of aspects weights is 1. Here the *shutter* aspect seems to be very important for Amazon users whilst *deal*, found in 67 different products, is about 30 times less appreciated. It is interesting to note that aspect frequencies are equally distributed, ranging from 59 to 70 in our set formed by 80 products, however the weight distribution provides a finer-grained differentiation.
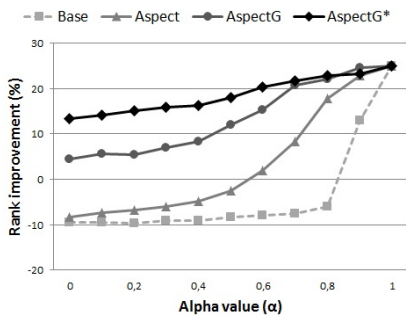
Fig. 10. *ProdRank* on Popular
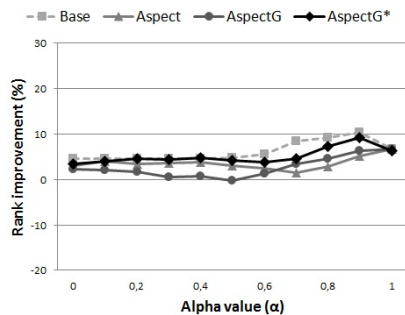
Fig. 11. *ProdRank* on Star-Rating

Next we study the mixing parameter α using 90 shared aspects in Figure 10 and Figure 11; where high values of α mean *PrefRank* having a greater influence on *ProdRank* compared to that of *SentiRank*. In general we observe that increasing values of α lead to increasing gain in benchmark positions. This is most notable on the Popular benchmark, where it achieves a maximum of 24% with all four strategies; whilst showing only a very modest gain of 3% with $\alpha = 0.9$ on the Star-Rating benchmark.

6   Conclusion

Social recommenders have created new opportunities for CBR research based on harnessing and reusing people's online experiences. The volume and variety of social media calls for innovative extraction, representation, retrieval and knowledge integration algorithms.

In this paper we formalised the extraction and integration of two social media sources: sentiment knowledge in product reviews; and preferences from purchase summary statistics. The benefits are demonstrated in a realistic recommendation setting using benchmarks generated from social media. We show that higher precision in aspect extraction was achieved when grammatical rules are combined with support statistics. Importantly there was over 50% shared aspects on average between any product pair, providing for non-sparse aspect-level representations. We confirm that preference knowledge can be conveniently exploited using the PageRank algorithm, and demonstrated the benefits of inferring aspect weights from preference graphs.

In the absence of ground truth data, generation of benchmark data becomes important for recommender research in general. Whilst there are a variety of social data dimensions that can be used to derive benchmark metrics, further work is needed to understand the interactions between these dimensions. Our results show that the combination of sentiment and preference knowledge are promising, but further work is needed to study closely the role of each on different domains and benchmarks. Finally it would also be interesting to explore how aspect importance weights are likely to evolve with context and time.

References

1. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE Transactions on Knowledge and Data Engineering 17 (2005)
2. Billsus, D., Pazzani, M.: A hybrid user model for news story classification. Courses and Lectures - Inter. Center for Mechanical Sciences 99 (1999)
3. Blair-Goldensohn, S., Hannan, K., McDonald, R., Neylon, T., Reis, G., Reynar, J.: Building a sentiment summarizer for local service reviews. In: WWW Workshop on NLP in the Information Explosion Era, p. 14 (2008)
4. Brin, S., Page, L.: The anatomy of a large-scale hypertextual web search engine. Computer Networks and ISDN Systems 30(1), 107–117 (1998)
5. Burke, R.: Hybrid recommender systems: Survey and experiments. User Modeling and User-Adapted Interaction, 331–370 (2002)
6. Cao, B., Shen, D., Wang, K., Yang, Q.: Clickthrough log analysis by collaborative ranking. In: AAAI Conf. on Artificial Intelligence (2010)

7. De Marneffe, M., MacCartney, B., Manning, C., et al.: Generating typed dependency parses from phrase structure parses. In: Proc. of Language Resources and Evaluation Conference, pp. 449–454 (2006)
8. Ding, X., Liu, B., Yu, P.S.: A holistic lexicon-based approach to opinion mining. In: Proc. Int. Conf. on Web Search and Data Mining (2008)
9. Dong, R., Schaal, M., O'Mahony, M.P., McCarthy, K., Smyth, B.: Opinionated product recommendation. In: Delany, S.J., Ontañón, S. (eds.) ICCBR 2013. LNCS, vol. 7969, pp. 44–58. Springer, Heidelberg (2013)
10. Esuli, A., Sebastiani, F.: Sentiwordnet: A publicly available lexical resource for opinion mining. In: Proc. Language Resources and Evaluation Conference, pp. 417–422 (2006)
11. Gauch, S., Speretta, M., Chandramouli, A., Micarelli, A.: User profiles for personalized information access. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) Adaptive Web 2007. LNCS, vol. 4321, pp. 54–89. Springer, Heidelberg (2007)
12. Groh, G., Birnkammerer, S., Köllhofer, V.: Social recommender systems. In: Pazos Arias, J.J., Fernández Vilas, A., Díaz Redondo, R.P. (eds.) Recommender Systems for the Social Web. ISRL, vol. 32, pp. 3–42. Springer, Heidelberg (2012)
13. Gupta, S., Chakraborti, S.: UtilSim: Iteratively helping users discover their preferences. In: Huemer, C., Lops, P. (eds.) EC-Web 2013. LNBIP, vol. 152, pp. 113–124. Springer, Heidelberg (2013)
14. Hu, M., Liu, B.: Mining and summarising customer reviews. In: Proc. of ACM SIGKDD Inter. Conf. on Knowledge Discovery and Data Mining, KDD 2004, pp. 168–177 (2004)
15. Kim, S., Pantel, P., Chklovski, T., Pennacchiotti, M.: Automatically assessing review helpfulness. In: Proc. Conf. on Empirical Methods in Natural Language Processing, pp. 423–430 (2006)
16. Linden, G., Smith, B., York, J.: Amazon.com recommendations: Item-to-item collaborative filtering. IEEE Internet Computing 7(1), 76–80 (2003)
17. McCarthy, K., Salem, Y., Smyth, B.: Experience-based critiquing: Reusing critiquing experiences to improve conversational recommendation. In: Bichindaritz, I., Montani, S. (eds.) ICCBR 2010. LNCS, vol. 6176, pp. 480–494. Springer, Heidelberg (2010)
18. McGinty, L., Smyth, B.: Collaborative case-based reasoning: Applications in personalised route planning. In: Aha, D.W., Watson, I. (eds.) ICCBR 2001. LNCS (LNAI), vol. 2080, pp. 362–376. Springer, Heidelberg (2001)
19. Moghaddam, S., Ester, M.: Opinion digger: An unsupervised opinion miner from unstructured product reviews. In: Proc. Inter. Conf. on Information and Knowledge Management, CIKM 2010 (2010)
20. Moghaddam, S., Ester, M.: On the design of lda models for aspect-based opinion mining. In: Proc. Inter. Conf. on Information and Knowledge Management, CIKM 2012 (2012)
21. Muhammad, A., Wiratunga, N., Lothian, R., Glassey, R.: Contextual sentiment analysis in social media using high-coverage lexicon. In: Research and Development in Intelligent Systems, pp. 79–93 (2013)
22. Popescu, A., Etzioni, O.: Extracting product features and opinions from reviews. In: Natural Language Processing and Text Mining, pp. 9–28. Springer, London (2007)
23. Quijano-Sánchez, L., Bridge, D., Díaz-Agudo, B., Recio-García, J.A.: Case-based aggregation of preferences for group recommenders. In: Agudo, B.D., Watson, I. (eds.) ICCBR 2012. LNCS, vol. 7466, pp. 327–341. Springer, Heidelberg (2012)

24. Singla, A., White, R., Huang, J.: Studying trailfinding algorithms for enhanced web search. In: Proc. ACM SIGIR Conf. on Information Retrieval, pp. 443–450 (2010)
25. Turney, P.: Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In: Proc. Annual Meeting on Association for Computational Linguistics, pp. 417–424 (2002)
26. Vasudevan, S., Chakraborti, S.: Mining user trails in critiquing based recommenders. In: Proc. Inter. Conf. on World Wide Web Companion, pp. 777–780 (2014)
27. Yitzhaki, S.: Relative deprivation and the Gini coefficient. The Quarterly Journal of Economics, 321–324 (1979)