

PEOPLE I KNOW

Giovanni Semeraro
(Università di Bari, Italy
semeraro@di.uniba.it)

Pasquale Lops
(Università di Bari, Italy
lops@di.uniba.it)

Marco Degemmis
(Università di Bari, Italy
degemmis@di.uniba.it)

Abstract: The recent evolution of e-commerce and the astonishing growth of the Internet have increased the amount of information that scrupulous customers want to process before selecting items that meet their needs. Personalization has become an important strategy in Business to Consumer e-commerce, where knowledge about customers can be exploited in order to improve access to relevant products. This paper presents a machine learning-based approach to turn raw data about customers into knowledge about their interests. This knowledge is stored in personal profiles and is used to provide an intelligent search support.

Key Words: personalization, user profiling, intelligent search and retrieval, Bayesian learning, learning from labeled and unlabeled

Category: H.3.3, H.3.5

1 Introduction

In the era of Internet, a huge amount of data is available to everybody, in every place and at any moment. This is extremely useful and exciting, but it generates anxiety, especially in novice or occasional users. Finding relevant information about products is very time consuming for customers. The main challenge is to support them to facilitate searching in online product catalogues. A possible way to overcome this problem is the development of intelligent systems to provide personalized information services [Schafer et al., 2001]. Customers have started to require *personalized* support responding to their specific needs, in order to avoid the access to electronic shops that often appear like a warehouse, where you must know exactly what you want and where to find it. Effectively supporting user searching and browsing over such large repositories entails the problem of properly understanding user needs and filtering out not relevant items. The complexity of the problem might be lowered by the automatic construction of machine processable profiles that can be exploited to deliver personalized content to the user. This process is called *user modelling*. In our approach, machine learning techniques are used to infer the profiles that can be exploited to provide an intelligent search support when relevant information has to be retrieved.

2 Learning User Profiles to Know Customers' Preferences

2.1 Learning from Transactions

The process of learning customers' profiles is performed by the *Profile Extractor* [Semeraro et al., 2003], which employs supervised learning techniques to automatically discover users' preferences from transactional data recorded during past visits to the e-commerce web site. The Profile Extractor builds *coarse-grained* profiles containing the product categories the user is interested into. The problem of learning a user's preferences can be cast to the problem of inducing general concepts from examples labeled as members (or non-members) of the concepts. In this context, given a finite set of n categories of interest $C = \{c_1, c_2, \dots, c_n\}$, the task consists in learning the target concept T_i "users interested in the category c_i ". In the training phase, each user represents a positive example for each category he likes, and a negative example for each category he dislikes. We have chosen an operational description of the target concept T_i , using a collection of rules that match against the features that describe a user in order to decide if he is a member of T_i . The system has been tested on a set of data about customers accessing to a virtual bookshop. In this context, we have chosen the main book categories the product database of the virtual bookshop is subdivided into as categories of interest. Transactional data about customers (Users' History) are arranged into a set of unclassified instances (each instance represents a customer). The subset of the instances chosen to train the learning system has to be labeled by a domain expert, that classifies each instance as member or non-member of each book category. The training instances are processed by the Profile Extractor that induces a classification rule set for each book category. The learning algorithm is PART [Frank and Witten, 1998], which produces rules from pruned partial decision trees. The rule sets are used to predict whether a user is interested in each book category. Interesting and transactional data are put together to form the user profile, composed of two main frames: *factual* (personal and transactional data), and *behavioural* (preferred book categories ranked according to the degree of interest computed by the system).

2.2 Learning from Textual Descriptions

Our intention was to enrich the knowledge contained in the coarse-grained profiles, generated by the Profile Extractor, by exploiting other information sources such as product descriptions the user finds interesting. We adopted the naïve Bayes algorithm [Sebastiani, 2002] to classify the textual descriptions of the books because it has been shown to perform competitively with more complex algorithms in text classification applications [Pazzani and Billsus, 1997, Mooney and Roy, 2000]. Our prototype, the *ITem Recommender* (ITR), classifies books belonging to a specific category as interesting or uninteresting for a user. In our learning problem, each instance is represented by a set of slots. Each slot is a textual field corresponding to a feature of a book: *title*, *authors* and *textual annotation* (abstract). A pattern-matcher analyses the web pages containing the book descriptions and extracts the words to fill each slot (it also eliminates stopwords and applies stemming). The text in each slot is a bag of words (BOW) processed taking into account their occurrences in the original text. Given a set of classes $C = \{c_1, c_2, \dots, c_n\}$, the conditional probability of a class c_j given a

document d_i is calculated according to the Bayes' theorem. In our problem, we have two classes: c_+ represents the positive class (user-likes), and c_- the negative one (user-dislikes). Since instances are represented as a vector of three documents (BOWs), the conditional probability of a category c_j given an instance d_i is computed using the formula:

$$P(c_j | d_i) = \frac{P(c_j)}{P(d_i)} \prod_{m=1}^{|S|} \prod_{k=1}^{|b_{im}|} P(t_k | c_j, s_m)^{n_{kim}} \quad (1)$$

where S is the set of slots, b_{im} is the BOW in the slot s_m of the instance d_i , n_{kim} is the number of occurrences of the token t_k in b_{im} . To calculate (1), we need to estimate the probability terms $P(c_j)$ and $P(t_k | c_j, s_m)$ from the training data. Thus each instance is weighted using a discrete rating r_i (1-10) provided by a user:

$$\omega_i^+ = \frac{r_i - 1}{9} \quad \omega_i^- = 1 - \omega_i^+ \quad (2)$$

and the weights ω_i^+ and ω_i^- are used for estimating the two probability terms:

$$\hat{P}(c_j) = \frac{\sum_{i=1}^{|TR|} \omega_i^j}{|TR|} \quad (3) \quad \hat{P}(t_k | c_j, s_m) = \frac{\sum_{i=1}^{|TR|} \omega_i^j n_{kim}}{\sum_{i=1}^{|TR|} \omega_i^j |b_{im}|} \quad (4)$$

In (4), the denominator denotes the total weighted length of the slot s_m in the class c_j . This approach allows for the refinement of the coarse-grained profiles by including a probabilistic model that is able to describe a customer's preferences for each book category the system was trained on. The final outcome is a *fine-grained* profile. Summarising, the complete process for generating profiles exploits machine learning techniques to turn raw data about customers into knowledge stored in personal profiles [Fig. 1]. An experimental evaluation showing the promise of the approach is reported in [Degemmis et al., 2003].

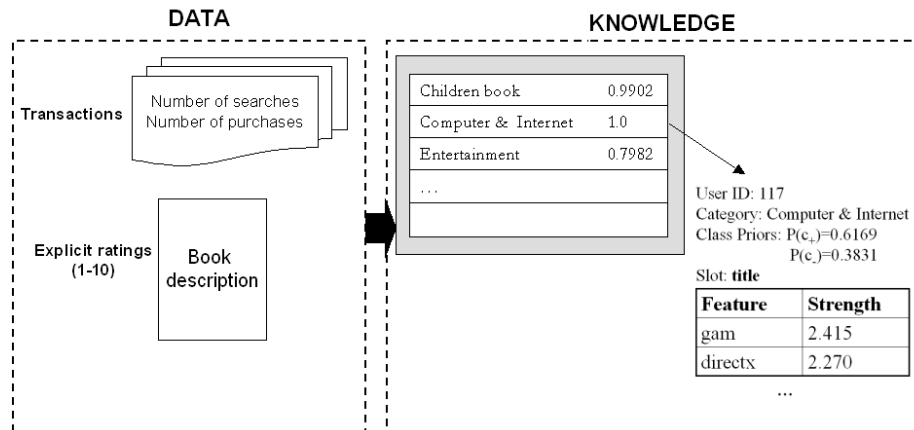


Figure 1: From data to knowledge using machine learning

3 Including Unlabeled Data Using EM

The process of learning from the textual descriptions has the drawback of requiring many examples labeled with ratings. This pre-classification task is often expensive and boring for a user. As a consequence, the approach is unfeasible in real world applications due to the difficulty to obtain from users a considerable number of rated items to learn an accurate classifier. A possible solution comes from the use of unlabeled examples in the learning phase. The idea is to use an algorithm [Fig. 2] that learns from few labeled examples and a large pool of unlabeled ones, combining the *Expectation-Maximization (EM)* technique [Dempster et al., 1977] with the naïve Bayes classifier, as in [Nigam et al., 2000]. Our prototype, called EMIR, integrates ITR with the EM algorithm, and has been used to evaluate the approach in the area of recommender systems, where the problem of obtaining user ratings is of primary importance.

Input: D_i^l = items rated in a specific category by user U_i
 D_i^u = items not rated by the user U_i in the category

1. Train ITR using only labeled documents D_i^l
 Parameters are estimated as in Eq. (3) and (4). Obtain a classifier NB_0
2. Loop while classifier parameters change
(E-step): Use the current classifier NB_k to compute $P(c_+|d_j)$ for each d_j in D_i^u (see Eq. (1))
(M-step): Re-estimate classifiers parameters using Eq. (3) e (4) given the probabilistically assigned class for d_j

Figure 2: Algorithm for learning user profiles using EM

4 Experimental session

4.1 Experimental Data Set

For the experiment, 5 book categories were selected at the Web site of a virtual bookshop. For each book category, a set of book descriptions was obtained and stored in a local database. Table 1 gives the specific figures for each category in terms of number of books available, number of books with a textual annotation, and average length (number of words) of the textual annotation. Each user involved in the experiment was requested to choose one category of interest and to rate 100 books, providing discrete ratings between 1 and 10. In this way, a dataset for each category was obtained. From each dataset D_i , 70 examples were randomly selected and used as training set TR_i ; the rest of D_i was used as test set TS_i .

Dataset	# of Books	# of Books with abstract	Abstract length (avg. value)
Computing & Internet	5414	4190 (77%)	42.39
Fiction & literature	6099	3378 (55%)	35.54
Travel	3179	1541 (48%)	28.29
Business	5527	3668 (66%)	42.04
SF, horror & fantasy	667	484 (72%)	22.33
Total	20886	13261 (63%)	

Table 1: Database Information

4.2 Experiment Description and Evaluations Measures

An experiment was performed to compare the performance of EMIR with respect to ITR. Specifically, the main objective of the experiment was to verify whether EMIR could reach the same performance of ITR using a fewer number of labeled instances and exploiting a large pool of unlabeled examples (all books not rated by the user).

For each TR_i , we randomly selected n labeled instances to build two distinct classifiers, one running ITR and the other running EMIR. The former was induced just from the n labeled documents, while all the available unlabeled instances were also used for the latter. The training phase has been repeated 80 times by varying n in the set $\{5, 10, 20, 30, 40, 50, 60, 70\}$ and, for each one of the 8 values of n , by repeating 10 times the random choice of the n labeled instances for the TR_i .

Several metrics were used in the testing phase. Since the task was to identify or retrieve items preferred by users from a repository, traditional information retrieval measures were adopted, namely *Precision (Pr)*, *Recall (Re)* and *F-measure (F)* [Sebastiani, 2002]. Moreover, we also adopted the *Normalized Distance-based Performance Measure (NDPM)* [Yao, 1995] to evaluate the goodness of the items' ranking calculated according to a certain relevance measure. Specifically, *NDPM* was exploited to measure the distance between the ranking imposed by the user ratings and the ranking predicted by the system. Values range from 0 (agreement) to 1 (disagreement).

4.3 Analysis of Results

For the sake of simplicity, we present only results for the datasets "Computing and Internet" and "Travel". Fig. 3 depicts the effect of varying the amount of labeled data.

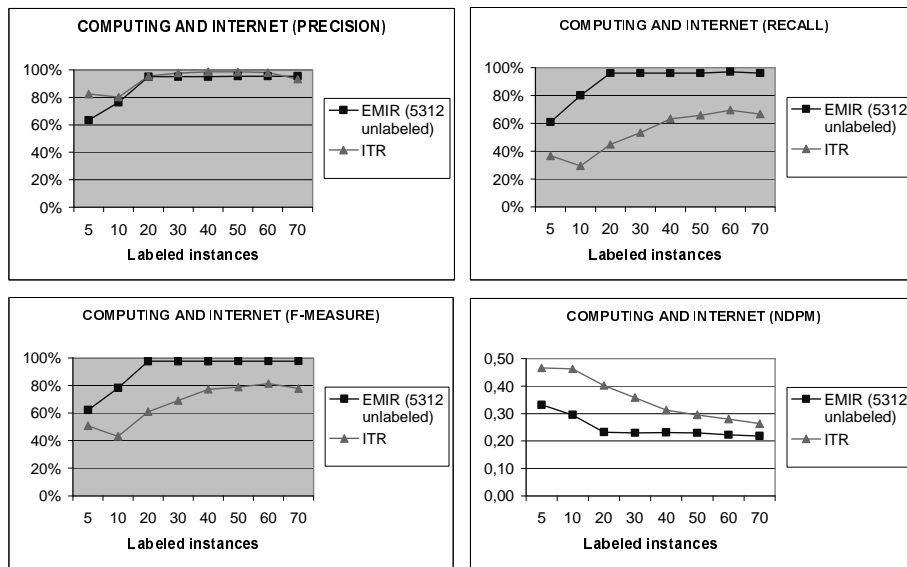


Figure 3: Performance measures of ITR and EMIR on the “Computing and Internet” dataset.

The number of unlabeled instances is constant, while the number of labeled instances in the horizontal axis varies. The values of measures are calculated by averaging the results of the 10 trials as described in Section 4.2. The results show that the use of EMIR improves Recall, F-Measure, and NDPM. For example, the highest Recall value reached by ITR is 69% (using 60 labeled instances), while EMIR reaches 96% of Recall using only 20 labeled instances. Again, EMIR dominates ITR as regards NDPM. As regards Precision, it seems that adding unlabeled data to a small amount of labeled data hurts performance. These results will be analyzed further.

We now move on to the “Travel” dataset. Results in Fig. 4 show that EMIR reaches the highest value of Precision using 20 labeled data, while ITR needs 50 labeled instances to obtain the same result. EMIR exceeds 80% Recall using 30 labeled instances, while ITR never achieves this result.

To sum up, EMIR outperforms ITR, as shown by values of F-Measure (that combines Precision and Recall). Finally, values of NDPM seem similar when the number of labeled instances is less than 30, but, surprisingly, ITR outperforms EMIR when the number of labeled instances is greater than 30. We think that this is probably due to the distribution of ratings given by the user. In fact, the average rating for the “Computing and Internet” dataset is 6.55 and 70% of training documents is positive (rating from 6 to 10), while for the “Travel” dataset the average rating is 6.33 and 66% of training documents is positive. Probably this is not the only reason for such a behavior; therefore we will further investigate on that.

In order to verify if the obtained results are statistical significant, we adopted the non-parametric Wilcoxon signed rank test (Orkin and Drogin 1990), since the number of independent trials (i.e., datasets) is relatively low and does not justify the application of a parametric test. The test was adopted in order to evaluate the difference in effectiveness of the two systems according to the performance metrics described in Section 4.2. The test compared values obtained by ITR when trained using 70 labeled instances, with values obtained by EMIR trained with a set of n labeled instances, n varying from 5 to 70.

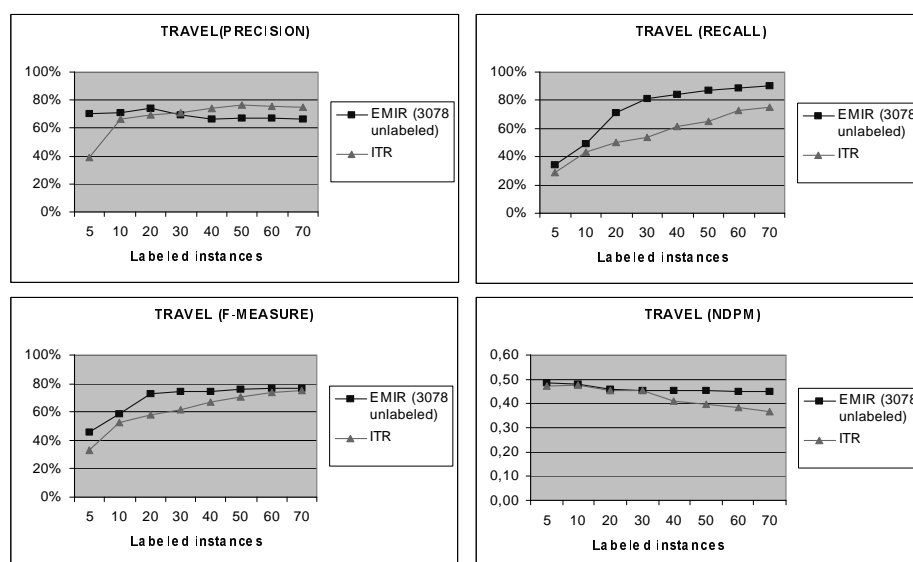


Figure 4: Performance measures of ITR and EMIR on the "Travel" dataset.

Dataset	Precision		Recall		F-Measure		NDPM	
	ITR(70)	EMIR(5)	ITR(70)	EMIR(5)	ITR(70)	EMIR(5)	ITR(70)	EMIR(5)
Computing & Internet	0,933	0,635	0,667	0,610	0,778	0,622	0,263	0,332
Business	0,667	0,418	0,706	0,641	0,686	0,506	0,438	0,357
Travel	0,750	0,703	0,750	0,340	0,750	0,458	0,367	0,485
SF, fantasy & horror	0,875	0,722	0,333	0,348	0,483	0,469	0,436	0,545
Fiction & literature	0,400	0,295	0,100	0,285	0,160	0,290	0,640	0,398
Avg.	0,725	0,554	0,511	0,445	0,571	0,469	0,429	0,423
W=	15		5		11		-1	

Table 2: Results of the comparison between ITR (70 labeled) and EMIR (5 labeled)

Dataset	Precision		Recall		F-Measure		NDPM	
	ITR(70)	EMIR(20)	ITR(70)	EMIR(20)	ITR(70)	EMIR(20)	ITR(70)	EMIR(20)
Computing & Internet	0,933	0,950	0,667	0,960	0,778	0,975	0,263	0,233
Business	0,667	0,662	0,706	0,818	0,686	0,732	0,438	0,383
Travel	0,750	0,737	0,750	0,715	0,750	0,726	0,367	0,458
SF, fantasy & horror	0,875	0,746	0,333	0,500	0,483	0,599	0,436	0,463
Fiction & literature	0,400	0,422	0,100	0,200	0,160	0,271	0,640	0,420
Avg.	0,725	0,703	0,511	0,639	0,571	0,660	0,429	0,391
W=	1		-13		-13		5	

Table 3: Results of the comparison between ITR (70 labeled) and EMIR (20 labeled)

Dataset	Precision		Recall		F-Measure		NDPM	
	ITR(70)	EMIR(30)	ITR(70)	EMIR(30)	ITR(70)	EMIR(30)	ITR(70)	EMIR(30)
Computing & Internet	0,933	0,950	0,667	0,960	0,778	0,975	0,263	0,230
Business	0,667	0,659	0,706	0,847	0,686	0,741	0,438	0,378
Travel	0,750	0,695	0,750	0,810	0,750	0,748	0,367	0,456
SF, fantasy & horror	0,875	0,776	0,333	0,538	0,483	0,636	0,436	0,459
Fiction & literature	0,400	0,741	0,100	0,160	0,160	0,263	0,640	0,428
Avg.	0,725	0,764	0,511	0,663	0,571	0,672	0,429	0,390
W=	1		-15		-13		5	

Table 4: Results of the comparison between ITR (70 labeled) and EMIR (30 labeled)

Dataset	Precision		Recall		F-Measure		NDPM	
	ITR(70)	EMIR(40)	ITR(70)	EMIR(40)	ITR(70)	EMIR(40)	ITR(70)	EMIR(40)
Computing & Internet	0,933	0,950	0,667	0,960	0,778	0,975	0,263	0,231
Business	0,667	0,659	0,706	0,900	0,686	0,761	0,438	0,363
Travel	0,750	0,663	0,750	0,840	0,750	0,741	0,367	0,454
SF, fantasy & horror	0,875	0,746	0,333	0,514	0,483	0,609	0,436	0,450
Fiction & literature	0,400	0,959	0,100	0,225	0,160	0,364	0,640	0,441
Avg.	0,725	0,796	0,511	0,688	0,571	0,690	0,429	0,388
W=	1		-15		-13		5	

Table 5: Results of the comparison between ITR (70 labeled) and EMIR (40 labeled)

Differences in performance are statistically significant ($p < 0.05$) only for *Precision*, in favor of ITR. This means that it is possible to obtain reliable recommendations using only 5 labeled instances, even if this leads to a loss in precision. The same results are obtained setting $n=10$.

Tables 3,4 and 5 report the summary of results with $n=20, 30, 40$ respectively.

Notice that with $n=20$ there is no difference ($p < 0.05$) between the systems. With $n \geq 30$, the results show that EMIR performs at least as well as ITR. Indeed, we observe that Recall calculated by EMIR is significantly higher than the one calculated by ITR and this difference is statistically significant. These results indicate that the number of labeled instances to obtain a reliable classifier for item recommending can be effectively reduced using a large amount of unlabeled data.

5 Conclusions

In this paper we discussed an approach to infer users' profiles for item recommending. We presented a process that turns transactional (browsing and purchasing history) and unstructured data (textual product descriptions) into knowledge about customers' preferences using supervised machine learning techniques. In particular, a naïve Bayes classifier was used to construct a model of user's interests based on user ratings. However, obtaining training labels is often expensive, thus we proposed also an approach based on the combination of EM and naïve Bayes classifier, that can exploit unlabeled documents in the training process. Experiments provide evidence that the quality of recommendations is improved.

References

- [Degemmis et al., 2003] Degemmis, M., Lops, P., Semeraro, G., Abbattista, F.: "Extraction of User Profiles by Discovering Preferences through Machine Learning"; Proc. Intelligent Information Systems: New Trends in Intelligent Information Processing and Web Mining, Advances in Soft Computing, Springer, Berlin (2003) (to appear).
- [Dempster et al., 1977] Dempster, M. M., Laird, N. M., Jain, D. B.: "Maximum Likelihood from Incomplete Data via the EM Algorithm"; Journal of Royal Statistical Society, Series B, 39 (1977), 1–38.
- [Frank and Witten, 1998] Frank, E., Witten, I. H.: "Generating Accurate Rule Sets without Global Optimisation"; Proc. International Conference on Machine Learning, Morgan Kaufmann, Menlo Park (1998), 144–151.
- [Mooney and Roy, 2000] Mooney, R. J., Roy, L.: "Content-based Book Recommending using Learning for Text Categorization", Proc. ACM Conference on Digital Libraries, ACM Press, New York (2000), 195-204.
- [Nigam et al., 2000] Nigam, K., McCallum, A. K., Thrun, S., Mitchell, T. M.: "Text Classification from Labeled and Unlabeled Documents using EM"; Machine Learning, 39, 2/3 (2000), 103–134.
- [Orkin and Drogin, 1990] Orkin M., Drogin R.: "Vital Statistics"; McGraw-Hill, New York (1990).
- [Pazzani and Billsus, 1997] Pazzani, M., Billsus, D.: "Learning and Revising User Profiles: The Identification of Interesting Web Sites"; Machine Learning, 27, 3 (1997), 313–331.
- [Schafer et al., 2001] Schafer, J., Konstan, J., Riedl, J.: "E-Commerce Recommendation Applications"; Data Mining and Knowledge Discovery, 5, 1/2 (2001), 115–153.
- [Sebastiani, 2002] Sebastiani, F.: "Machine Learning in Automated Text Categorization"; ACM Computing Surveys, 34, 1 (2002), 1–47.
- [Semeraro et al, 2003] Semeraro, G., Abbattista, F., Degemmis, M., Licchelli, O., Lops, P., Zambetta, F.: "Agents, Personalisation, and Intelligent Applications"; R. Corchuelo, A. Ruiz Cortés and R. Wrembel (Eds.), Technologies Supporting Business Solutions, Part IV: Data Analysis and Knowledge Discovery, Chapter 7, Nova Sciences Books and Journals, (2003), 163-186 (to appear).
- [Yao, 1995] Yao, Y. Y.: "Measuring Retrieval Effectiveness Based on User Preference of Documents"; Journal of the American Society for Information Science, 46, 2, (1995), 133–145.