View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by AIS Electronic Library (AISeL)

Association for Information Systems
AIS Electronic Library (AISeL)

ICIS 2009 Proceedings

International Conference on Information Systems (ICIS)

2009

Positive Example Learning for Content-Based Recommendations: A Cost-Sensitive Learning-Based Approach

Yen-Hsien Lee
National Chiayi University, yhlee@mail.ncyu.edu.tw

Paul Jen-Hwa Hu
University of Utah, actph@business.utah.edu

Tsang-Hsiang Cheng
Southern Taiwan University of Technology, cts@mail.stut.edu.tw

Ya-Fang Hsieh
National Chiayi University, s0961396@mail.ncyu.edu.tw

Follow this and additional works at: http://aisel.aisnet.org/icis2009

Recommended Citation

Lee, Yen-Hsien; Hu, Paul Jen-Hwa; Cheng, Tsang-Hsiang; and Hsieh, Ya-Fang, "Positive Example Learning for Content-Based Recommendations: A Cost-Sensitive Learning-Based Approach" (2009). *ICIS 2009 Proceedings*. 188.
http://aisel.aisnet.org/icis2009/188

This material is brought to you by the International Conference on Information Systems (ICIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in ICIS 2009 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

POSITIVE EXAMPLE LEARNING FOR CONTENT-BASED RECOMMENDATIONS: A COST-SENSITIVE LEARNING-BASED APPROACH

Completed Research Paper

Yen-Hsien Lee
National Chiayi University
Chiayi, Taiwan, R.O.C
yhlee@mail.ncyu.edu.tw

Paul Jen-Hwa Hu
University of Utah
Salt Lake City, Utah, USA
actph@business.utah.edu

Tsang-Hsiang Cheng
Southern Taiwan University
Tainan, Taiwan, R.O.C.
cts@mail.stut.edu.tw

Ya-Fang Hsieh
National Chiayi University
Chiayi, Taiwan, R.O.C
s0961396@mail.ncyu.edu.tw

Abstract

Existing supervised learning techniques can support product recommendations but are ineffective in scenarios characterized by single-class learning; i.e., training samples consisted of some positive examples and a much greater number of unlabeled examples. To address the limitations inherent in existing single-class learning techniques, we develop COst-sensitive Learning-based Positive Example Learning (COLPEL), which constructs an automated classifier from a single-class training sample. Our method employs cost-proportionate rejection sampling to derive, from unlabeled examples, a subset likely to feature negative examples, according to the respective misclassification costs. COLPEL follows a committee machine strategy, thereby constructing a set of automated classifiers used together to reduce probable biases common to a single classifier. We use customers' book ratings from the Amazon.com Web site to evaluate COLPEL, with PNB and PEBL as benchmarks. Our results show that COLPEL outperforms both PNB and PEBL, as measured by its accuracy, positive F1 score, and negative F1 score.

Keywords: Content-based Recommendation Systems, Cost-sensitive Learning, Single-class Learning, Positive Example-based Learning, Committee Machine

Introduction

Online retailing is one of the fastest growing segments in business-to-consumer e-commerce, offering an exciting global virtual channel for marketing and exchanges. Around the world, customers are increasingly accustomed to searching for products and making purchases online. According to the U.S. Department of Commerce, online retailing amounted to $81 billion in sales in 2005, with a compounded annual growth rate of 29% between 2000 and 2004 in the United States (Perez 2006). According to e-tailing.com (2007), U.S. consumers spent $109 billion on merchandise online in 2006, and a recent report by the U.S. Department of Commerce shows that online retail sales recorded $34.4 billion in the third quarter of 2008, a 5.7% increase from the same quarter in the previous year (Winters et al. 2008). Similar growth appears various countries, both developed and developing. For example, according to the Market Intelligence and Consulting Institute in Taiwan, online retail racked up $4.14 billion in 2008, representing an impressive 36% growth rate compared with 2007 (TrendGo 2007).

Vendors have far fewer “store” capacity constraints on their Web sites than their conventional (physical) store outlets. They provide customers with enormous product choices and in-depth information. But the wide range of products and voluminous product information often lead to information overload and can make customers’ product selection difficult, adversely affecting their shopping experiences (Haubl and Trifts 2000; Hostler et al. 2005; Lee and Lee 2004). To alleviate customers’ cognitive processing (commonly required in online retailing) and enhance their loyalty and value, many vendors develop and deploy automated recommender systems to make appropriate product recommendations to customers. In general, such systems analyze important information about different products (e.g., attributes) and customers (e.g., demographics, self-disclosed information, product search and purchase behaviors) to identify a subset of products presumably of interest to the focal customer and make recommendations accordingly, often in the form of a customized or personalized suggestion (Senecal and Nantel 2004; Wei et al. 2002; Xiao and Benbasat 2007). A good case in point is the instant recommendation function by Amazon, which makes book recommendations to customers according to their preferences (e.g., previous browsing, purchases) and the contents of the recommended book.

Collaborative and content-based filtering represent two crucial approaches for automated recommendation or user preference prediction systems (Basilico and Hofmann 2004). By and large, collaborative filtering exploits essential correlations between ratings across a population of users; in its most popular incarnation; it finds users most similar to the active user and thereby forms a weighted vote across these neighbors to predict unobserved ratings (Goldberg et al. 1992; Shardanand and Maes 1995). Content-based filtering, on the other hand, uses informative content descriptors to make recommendations, as applied in scenarios involving books, Web pages, or news items (Cheng and Hu 2007). Typically, content-based filtering assumes the existence of some important associations between or among products, measurable by their respective attributes, and analyzes these associations to recommend products relevant or appropriate to the focal customer’s need or preferences (Alspector et al. 1998). Customers generally consider a product interesting if it has attributes highly similar to or closely matched with their preferences (Balabanovic and Shoham 1997; Herlocker et al. 1999; Wei et al. 2002).

Automated product recommendation therefore can be considered as a classification problem; i.e., either the focal customer shows some interest in the product or otherwise. Accordingly, we can use an appropriate classification algorithm to construct, from a large sample of products (e.g., depicted by important attributes) and customers (e.g., described by demographics, behaviors, and/or preferences), an automated classifier for predicting whether a product, new or unlabeled, might be of interest to the customer. Previous research shows that automated classification is effective in recommendation scenarios involving structured data, such as music or movies (Ampazis 2008), as well as unstructured data, such as text documents, Web site URLs, and news (Krulwich and Burkey 1996; Lang 1995; Pazzani and Billsus 1997). The use of a conventional supervised learning-based classification technique to build an automated classifier is appealing, in that it could reveal important patterns embedded in a large training sample that consists of historical cases (instances) from distinct outcome classes. However, such techniques become ineffective in situations in which the training sample includes cases pertaining to only one outcome class. This situation underscores the challenge commonly known as single-class learning. In online retailing, it is not uncommon for a vendor to have some understanding of the products of interest to the focal customers (e.g., by tracking and analyzing products they browse, compare, purchase, or review); however, the vendor has difficulty gauging their potential interest in other products about which it has no search, browsing, or purchase behaviors to match to these customers.

Several techniques attempt to address this single-class learning challenge. For example, Tax and Duin (2004) develop support-vector data description (SVDD), which extends support vector machine (SVM) to accommodate a single-class training sample. This technique represents a substantial extension of SVM but seems to lack the efficiency and agility required by many recommendation scenarios, because it demands significant processing time. Yu et al. (2004) propose positive example-based learning (PEBL), which adopts an iterative strategy for prediction model building, using known positive examples as well as probable negative examples inferred from the unlabeled instances in the training sample. Denis et al. (2002) develop positive naïve Bayes (PNB) by adapting the naïve Bayes induction algorithm to positive example learning. The PNB estimates, from the positive and unlabeled examples, the a priori probability of each outcome class and thereby constructs an automated prediction model.

Although the initial evaluations of PNB and PEBL suggested their effectiveness (Denis et al. 2002; Yu et al. 2004), both techniques possess several inherent limitations that seriously impede or even prohibit their applications to automated recommendation. For example, the application of PNB assumes knowledge of the validity of the a priori probability of different outcome classes; this probability can significantly affect the performance of a naïve Bayes model (Sanchez-Hernandez et al. 2007; Wei et al. 2008). To estimate the a priori probability of the cases belonging to positive and negative outcome classes, PNB uses positive and unlabeled examples in the training sample, but in application to recommendation situations in online retailing, PNB appears likely to generate inaccurate a priori probability estimates, largely due to the difficulty of making reasonably accurate a priori probabilities from a training sample that consists of relatively few positive examples and a greater number of unlabelled examples.

The PEBL technique extends an existing learning-based classification technique and is relatively easy to implement in comparison to PNB. It identifies strong negative examples from the unlabelled examples (i.e., those not containing important features of positive examples), then uses both positive examples and strong negative examples to construct a rough classifier to predict the probable negative examples among the remaining unlabeled examples. The resulting (predicted) negative examples then appear in the training sample, from which a refined classifier can be constructed and used again to predict probable negative examples from the remaining unlabeled examples. This process continues until no more negative examples can be identified from the unlabeled examples. The effectiveness of PEBL thus greatly depends on the representativeness of the initial array of strong negative examples (Wei et al. 2008); using a less representative array of strong negative examples initiates a downward cycle that continually decreases the predictive power of the constructed classifier. Furthermore, the iterative processing of PEBL demands a significant amount of time and computational processing, which limits its applicability to automated recommendations.

To address the limitations of these existing approaches to single-class learning, we propose a COst-sensitive Learning-based Positive Example Learning (COLPEL) approach for constructing an automated classifier from a single-class training sample. For this method, our approach employs cost-proportionate rejection sampling (von Neumann 1951; Zadrozny et al. 2003) to derive from the unlabeled examples a subset likely to represent the negative examples, on the basis of their respective misclassification cost estimates. We develop a cost measure to examine the misclassification cost associated with each unlabeled example. In addition, COLPEL adopts a committee machine strategy; i.e., constructing a set of automated classifiers to reduce the probable biases common to the use of a single classifier. To empirically evaluate the effectiveness of COLPEL, we collect customers' book ratings from the Amazon.com Web site and use the predictions of PNB and PEBL for benchmark purposes.

The remainder of this paper is organized as follows: In Section 2, we review relevant prior research pertaining to content-based recommendation and single-class learning to highlight our motivation. In Section 3, we describe the proposed COLPEL approach, followed by details of our evaluation design and key results in Section 4. We conclude with a summary and discussions of the study's contributions and some future research directions in Section 5.

Literature Review

In this section, we review extant literature regarding content-based filtering for recommendations and single-class learning; in so doing, we highlight our research motivation.

Content-Based Filtering for Recommendation

Conceptually, automated systems make recommendations to a customer by sifting through a large collection of products to identify those that seem most relevant or of interest to that customer. To thrive in the highly competitive online retailing marketplace, most vendors offer customized or personalized services, often in the form of effective recommendations (Dewan et al. 2000; Murthi and Sarkar 2003). The ease of collecting, integrating, and analyzing vast amounts of data about products, customers, and their behaviors with respect to specific products has generated significant interest in personalized services and therefore fostered the practice of automated recommendations (Adomavicius and Tuzhilin 2005; Alspector et al. 1998; Wei et al. 2002). Many online vendors offer their own versions of personalized recommendation services (e.g., Amazon, Genius, StyleFeeder, Netflix).

Content-based filtering represents a common approach to support personalized recommendations; it constructs individual customer profiles to be used for identifying products that have attributes or characteristics highly similar to those affiliated with a specific profile; then it makes product recommendations accordingly (Balabanovic and Shoham 1997; Herlocker et al. 1999). This approach can be supported by existing statistical techniques, as well as inductive learning algorithms that construct an automated classifier to embrace important patterns in the customer's preference profile and thereby recommend appropriate products to that customer. Prior research shows that product recommendations using content-based filtering are generally effective (Krulwich and Burkey 1996; Lang 1995; Pazzani and Billsus 1997).

The general process of content-based filtering for automated recommendation includes the following phases: feature extraction and selection (i.e., extracting and selecting key features for all products under consideration), representation (i.e., representing each product using the feature set selected in the previous phase), user profile learning (i.e., automatically discovering and adaptively updating each user's preference profile model on the basis of a training sample), and recommendation generation (i.e., recommending specific products by examining the respective product features and user profile models (Adomavicius and Tuzhilin 2005; Wei et al. 2002).

In the feature extraction and selection phase, a set of features gets extracted from the analysis of all products under examination (Dumais et al. 1998; Ng et al. 1997). Feature selection then proceeds by selecting a subset of the extracted features to represent these products (Piramuthu 1998). Feature selection improves learning efficiency and effectiveness (Dumais et al. 1998; Kittler 1975; Modrzejewski 1993; Siedlecki and Sklansky 1989). A feature selection measurement score gets assigned to each feature, and the k features with the highest scores are selected to represent all products (i.e., top k features); a prespecified number of features (i.e., k) is included in the final feature set. Several evaluation functions have been proposed for feature selection, including within-document term frequency (TF), within-document term frequency × inverse document frequency (TF×IDF), correlation coefficient, mutual information, and the χ^2 metric (Dumais et al. 1998; Lam and Ho 1998; Lewis and Ringuette 1994; Ng et al. 1997).

The representation phase represents a product as an N-dimensional vector, using the N features extracted and selected from the training sample. Several schemes have been proposed to represent an extracted feature, including binary, TF, IDF, and TF×IDF (Yang and Chute 1994). In the user profile learning phase, we can construct a preference profile model for each user, which we then use to analyze the relationship between a user's preference (i.e., dependent variable) and the feature values of various product (i.e., independent variables) to make the recommendation. User profile learning can be supported by statistical techniques or computation methods, such as inductive learning, back-propagation neural network, and Bayesian probability (Alspector et al. 1998; Mooney and Roy 2000; Rumelhart et al. 1986).

The use of content-based filtering thus constructs a classification model using a training sample that consists of cases (instances) that belong to different outcome classes, such as positive or negative examples. However, such training samples may not always be available. For example, some classification or prediction applications have relatively few positive examples but a much greater number of unlabeled examples, for which the negative outcome class is difficult to obtain. In online retailing, for example, it is generally difficult to determine whether customers are interested in a product they have not browsed or purchased before. To make automated recommendations, prediction systems must address this single-class learning challenge.

Brief Review of Single-Class Learning Approaches

Conventional classification techniques focus on discovering the patterns or rules that are important for distinguishing the cases in the training sample, such as positive and negative outcome classes. They are ineffective for single-class learning characterized by some positive examples and a much larger number of unlabeled examples in the training sample. Prior research proposes several techniques to extend existing techniques and enable them to learn from examples of one outcome class (Tax and Duin 2004), estimate relevant parameters using the positive and unlabeled examples to construct a probability prediction model (Denis et al. 2002; Calvo et al. 2007), or identify the probable negative examples from the unlabeled examples and use an adequate existing classification technique based on the estimated negative examples and known positive examples (Yu et al. 2004). In the following, we briefly review positive naïve Bayes (PNB) and positive example-based learning (PEBL), two salient approaches for the single-class learning, and address their inherent limitations.

Positive Naïve Bayes

Positive naïve Bayes (PNB) takes as inputs a set of positive examples *PD* and a set of unlabeled examples *UD* to construct, on the basis of naïve Bayes predictions, a binary prediction model (Denis et al. 2002). Let C_p and C_n be the positive and negative classes, respectively. To classify a document d_j that consists of n words $\{w_1, \ldots, w_n\}$, PNB estimates the possible multiple occurrences of a word as a member of the class according to $PNB(d_j) = \underset{C\in\{C_p,C_n\}}{\operatorname{argmax}} \hat{Pr}(C) \prod_{i=1}^{n} \hat{Pr}(w_i|C)$, where the a prior probability $\hat{Pr}(C_p)$ of the class C_p is estimated from *PD* and *UD*; the a prior probability $\hat{Pr}(C_n)$ of the class C_n is defined as $1-\hat{Pr}(C_p)$; and the conditional probability $Pr(w_i|C_p)$ of word w_i given in the positive class is estimated by the term frequency of word w_i in *PD*, divided by the number of word occurrences in *PD*, that is, $\frac{N(w_i, PD)}{N(PD)}$. However, if a word w_i in document d_j does not appear in any documents in C_p, $Pr(w_i|C_p)$ becomes 0. To address this scenario, PNB adopts Lidstone's law of succession to smooth the maximum likelihood estimate (Agrawal et al. 2000) and defines $\hat{Pr}(w_i|C_p) = \frac{N(w_i, PD) + \lambda}{N(PD) + \lambda\times|V|}$, where $|V|$ denotes the cardinality of the number of distinct features in the training documents, and $\lambda \geq 0$.

According to Bayes' theorem though, the conditional probability of word w_i in a negative class can be estimated as $Pr(w_i|C_n) = \frac{Pr(w_i) - Pr(w_i|C_p)\times Pr(C_p)}{1 - Pr(C_p)}$, where $Pr(w_i)$ denotes the probability that the underlying generative model will create w_i. Then, PNB uses the set of unlabeled documents to estimate the probability $Pr(w_i)$, according to $\hat{Pr}(w_i) = \frac{N(w_i, UD)}{N(UD)}$. The estimate of $\hat{Pr}(w_i|C_n)$ becomes $\hat{Pr}(w_i|C_n) = \frac{N(w_i, UD) - \hat{Pr}(w_i|C_p)\times\hat{Pr}(C_p)\times N(UD)}{(1 - \hat{Pr}(C_p))\times N(UD)}$. Finally, according to Lidstone's law of succession, the conditional probability of the word w_i in a negative class can be estimated as $\hat{Pr}(w_i|C_n) = \frac{(N(w_i, UD) - \hat{Pr}(w_i|C_p)\times\hat{Pr}(C_p)\times N(UD)) + \lambda}{(1 - \hat{Pr}(C_p))\times N(UD) + \lambda\times|V|}$.

The estimates (e.g., $\hat{Pr}(C_n)$ and $\hat{Pr}(w_i|C_n)$) involved in PNB derive from the estimate of the prior probability $\hat{Pr}(C_p)$ of the class C_p, because we lack negative training examples. As a result, the accuracy of $\hat{Pr}(C_p)$ dominates the effectiveness of PNB, even though accurate estimates of $\hat{Pr}(C_p)$ are difficult to obtain. For classification applications in which the class distribution changes over time, dynamic adjustments of the $\hat{Pr}(C_p)$ estimate, in congruence with the true class distribution of the current state (or situation), are extremely difficult. These limitations significantly constrain the effectiveness and applicability of PNB.

Positive Example-Based Learning

Positive example-based learning (PEBL) recommends Web pages presumably of interest to a focal user (Yu et al. 2004), which it derives from a training sample that consists of positive examples (positive data set, *PD*) and unlabeled samples (unlabeled data set, *UD*). The resulting classification model ideally can delineate the cases with different outcome classes, namely, positive or negative. As we show in Figure 1, PEBL adopts a two-stage learning process for positive and unlabeled examples. In the mapping stage, PEBL employs a rough classifier derived from initial approximations of "strong negative" examples. Specifically, PEBL first identifies "strong positive" features by analyzing and identifying the important features most frequently observed in the positive examples but that rarely appear in the unlabeled examples. A feature can be considered a "strong positive" feature if it occurs in more than α_{PT}% of the positive examples but less than α_{UT}% of the unlabeled examples. Using these strong positive features, PEBL then identifies unlabeled examples (from *UD*) that contain none of these strong positive features and thus generates strong negative examples. The examples that remain in *UD* are plausible positive examples.

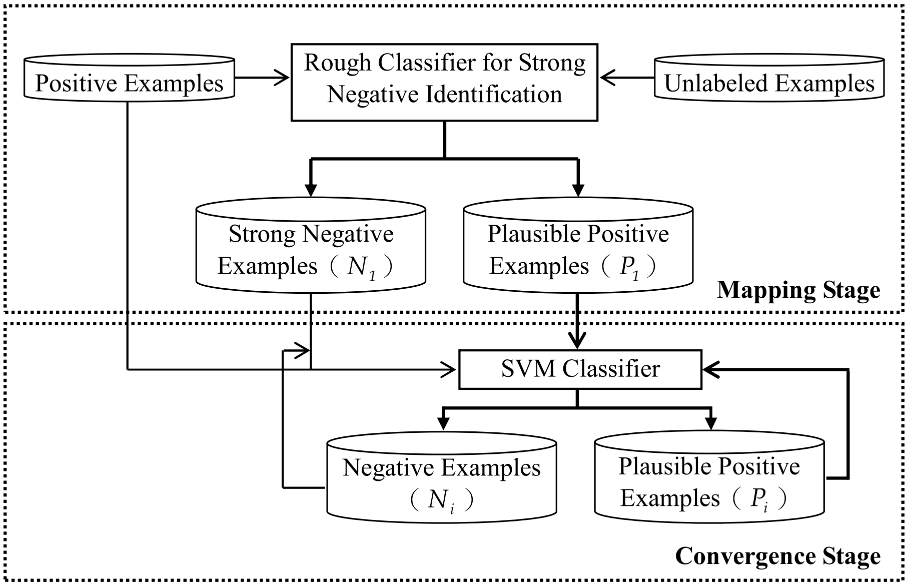
Positive Examples
Rough Classifier for Strong Negative Identification
Unlabeled Examples
Strong Negative Examples（N_1）
Plausible Positive Examples（P_1）
Mapping Stage
SVM Classifier
Negative Examples（N_i）
Plausible Positive Examples（P_i）
Convergence Stage

Figure 1. Overall PEBL Process

In the convergence stage, PEBL constructs an initial classifier using the positive examples and the strong negative examples obtained from the previous mapping stage. With a support vector machine (SVM), PEBL iteratively searches for additional negative examples from the remaining unlabeled examples (i.e., plausible positive examples) and then labels them "negative examples" in the training sample. In each iteration, PEBL uses the classification model constructed in its previous iteration to classify the current set of plausible positive examples into positive and negative examples. That is, PEBL expands the set of negative examples by including the (additional) negative examples identified in its current iteration, then uses SVM to reconstruct a new classification model. The set of examples predicted as the positive class in each iteration serves as plausible positive examples in the next iteration. By continuing this process for identifying and selecting negative examples from the remaining unlabeled examples, PEBL builds new classification models until no additional negative examples can be found from the unlabeled examples in *UD*. At the completion of the convergence stage, the delineation of learning examples pertaining to different outcome classes also converges to form reasonable boundaries of the positive-class examples in the feature space.

COst-Sensitive Learning-Based Positive Example Learning Approach (COLPEL)

We propose COLPEL, for which we adopt a cost-sensitive learning method and a committee machine strategy, to address the single-class learning problem. As we show in Figure 2, COLPEL comprises three phases: strong positive feature selection, misclassification cost estimation, and cost-sensitive learning. The objective of the first stage is to select the features most representative of the positive examples (i.e., customer's preference). The cost of misclassifying each unlabeled example can be estimated on the basis of the strong positive features. Subsequently, these estimated costs provide an important criterion for sampling negative examples when COLPEL performs cost-sensitive learning. To sample the unlabeled examples, we adopt cost-proportionate rejection sampling according to the respective misclassification costs, which generates negative examples that can be used in combination with positive examples to construct a set of classifiers (i.e., committee machine) for automated recommendation. We provide further details about each COLPEL phase in the following sections.

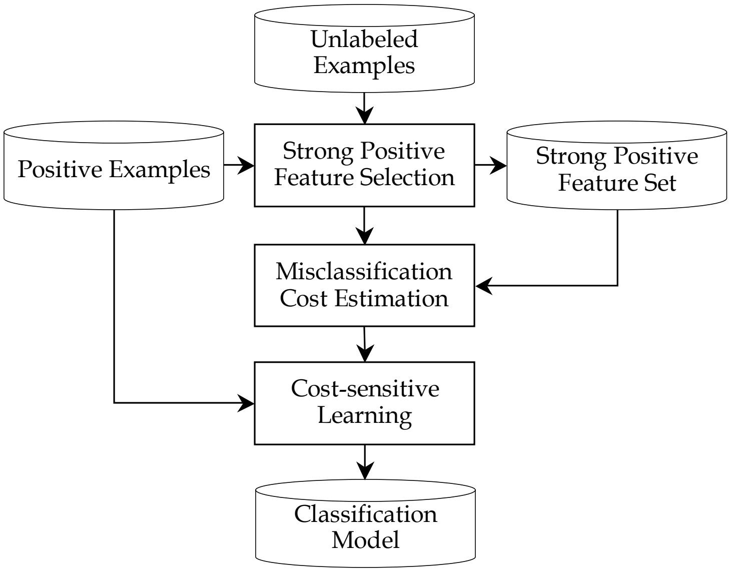
Unlabeled Examples
Positive Examples
Strong Positive Feature Selection
Strong Positive Feature Set
Misclassification Cost Estimation
Cost-sensitive Learning
Classification Model

Figure 2. Overall COLPEL Process

Strong positive feature selection. In this phase, COLPEL identifies the features that are most representative of the positive examples. We adopt the method used by PEBL to select a strong positive feature set, F_p. Specifically, a feature is included in the strong positive feature set if it appears in more than α_{PT}% of the positive examples and less than α_{UT}% of the unlabeled examples.

Misclassification cost estimation. We assume that each unlabeled example is a negative example, though not all the unlabeled examples have an equal probability of being negative examples. In this phase, we therefore examine the importance of each unlabeled example by estimating its misclassification cost—that is, the cost of misclassifying an unlabeled example as a positive example. Intuitively, the higher the misclassification cost of an unlabeled example, the more likely it is to be a negative example. To estimate the misclassification cost of each unlabeled example, we consider not only the difference between the sample's features and the strong positive feature set but also the length of the feature list of the unlabeled examples. An example is probably relevant or similar to another example if they share more features; otherwise, they may not be relevant. However, an example with many features has a better chance of overlapping with other examples. Furthermore, an example with a longer list of features should exhibit a higher tendency toward a negative example than will those with a shorter list, even if the overlap between the strong positive feature set and their respective features is identical. That is, an unlabeled example tends to belong to the negative outcome class if it has more features but a small overlap between its features and the strong positive feature set. We therefore define a function for estimating the misclassification cost: $MC(e_i) = (\underset{i=1}{\overset{n}{Max}}(|F_p \cap F_{e_i}|)\text{-}|F_p \cap F_{e_i}|)^2 \times \log(|F_{e_i}|)$, where F_p represents the strong positive feature set, F_{e_i} denotes the set of features appearing in an unlabeled example e_i, and $\underset{i=1}{\overset{n}{Max}}(|F_p \cap F_{e_i}|)$ is the calculated maximum number of the strong positive features appearing in all unlabeled examples (i.e., the maximum overlap between the strong positive feature set and the features of each unlabeled example). The purpose of using maximum overlap to deduct the overlap obtained by a specific unlabeled example is to make the minimal misclassification cost equal 0. An example e_i may result in a larger misclassification cost if the overlap between its features and the strong positive features set is small. We also increase the effect of feature overlapping by using the squared difference between the maximum overlap and the respective overlap of the examples thus obtained. To illustrate the misclassification cost function, assume F_p = {A, B, C, D, E}, F_{e_1} = {A, F, H, I}, and F_{e_2} = {A, B}. In this case, the misclassification cost for e_1 is $(2 – 1)^2 \times \log(4) = 2$, and the cost for e_2 is $(2 – 2)^2 \times \log(2) = 0$. Therefore, e_1 has a higher misclassification cost than e_2. Because it exhibits a greater likelihood of being a negative example, e_1 should have a higher probability of being included in the training sample as a negative example. Misclassification cost is relative; thus, an example with a higher misclassification cost suggests that it has a higher probability of being, but will not necessarily be, sampled as a negative example during the cost-proportionate rejection sampling. In contrast, an example with a lower cost has a lower probability of being sampled, which should not be taken to mean that example cannot be sampled prior to examples with higher misclassification costs.

Cost-sensitive learning. The use of the unlabeled examples of differential importance (i.e., misclassification costs) to construct an automated classifier resembles cost-sensitive learning, which performs learning to build a classifier by comparing with examples that have non-uniform misclassification costs (Zadrozny et al. 2003). Several cost-sensitive learning methods have been proposed (Domingos 1999; Drummond and Holte 1999; Fan et al. 1999; Margineantu 2002; Zadrozny et al. 2003); of these methods, using a sampling method to select the examples from the original training data set—on the basis of the respective importance or misclassification cost—and using them to construct an automated classifier with an appropriate classification learning algorithm seems effective and computationally efficient. Thus, we can transform a cost-sensitive learning problem to a cost-insensitive learning task (Zadrozny et al. 2003).

We adopt this method in the cost-sensitive learning phase, during which COLPEL performs two tasks: negative example sampling and classifier learning. For negative example sampling, COLPEL employs cost-proportionate rejection sampling (von Neumann 1951) to identify, from the costly unlabeled examples, those belonging to the negative outcome (i.e., negative examples). When deciding whether to include an example, cost-proportionate rejection sampling first generates a random number r_i ranging from 0 to 1, then compares it with the acceptance probability c_i/Z of the example under evaluation, where c_i represents the misclassification cost of an example and Z is a predefined constant. An example e_i gets excluded from the sampling if r_i is greater than its acceptance probability and is included otherwise. We refine the acceptance probability measure as $c_i/(\beta \times Z)$, where β denotes the weight of Z. As suggested by Zadrozny et al. (2003), the constant Z often gets assigned as the maximal cost among all the examples. We introduce a parameter β to control for the sampling size; as a result, we can increase the sample size by setting $\beta < 1$ or reduce the size by setting $\beta > 1$. Furthermore, using the sample set created by the cost-proportionate rejection sampling method can guarantee an approximately cost-minimizing classifier, as long as the classifier learning algorithm achieves an approximate minimization of the classification error (Zadrozny et al. 2003). Thus, COLPEL adopts a committee machine strategy and generates k sets of the negative examples randomly to construct individual classifiers that may join the committee, thereby reducing the probable classification biases that result from the use of any single classifier. For classifier learning, each set of randomly sampled negative examples, together with the positive examples, induces a classifier using a cost-insensitive learning technique, that is, SVM (also adopted by PEBL). As a result, *k* classifiers get generated and form a committee to recommend specific products to a customer if more than half of the classifiers favor those recommendations.

Empirical Evaluation and Results

We empirically evaluate the effectiveness of COLPEL, using book ratings collected from the Amazon.com Web site. Our evaluation includes PNB and PEBL as performance benchmarks. We therefore detail our data collection, evaluation design, and important evaluation results.

Data Collection

Our evaluation targets content-based, personalized recommendations. We randomly selected 36 subjects from the top-reviewer list of Amazon and gathered their book ratings, as well as the important features of each book they rated. Amazon's book rating system, ranging from 1 to 5, signifies each reviewer's preference with respect to a specific book. For each book, Amazon also provides statistically improbable phrases (SIP) and capitalized phrases (CAP), which give customers a quick glimpse of its contents. In addition, Amazon uses content-browsing software to identify the SIPs that represent the most distinctive phrases in a book. The CAPs are phrases pertaining to people, places, events, or topics frequently occurring in a book. Together, SIPs and CAPs offer the key (content) features of the books included in our study.

We collected 23,162 book ratings generated by the 36 top-reviewers' preference records pertaining to 18,469 books. Among these reviewers, the maximal number of book ratings generated is 2,681, and the minimal number is 65. The maximal, minimal, and average numbers of content features (i.e., SIPs and CAPs combined) are 155, 2, and 57, respectively. In our evaluation, we consider books that receive a rating of 4 or higher as positive examples (i.e., of interest to a customer), and negative examples otherwise. Overall, our data set consists of 20,962 positive examples and 2,200 negative examples. Among these reviewers, the maximal, minimal, and average ratios of positive examples to negative examples are 84.6:1, 2.8:1, and 17.2:1, respectively. The data set has a significantly skewed distribution between the positive and the negative examples; thus, it is appropriate for our evaluation purposes.

Evaluation Design

An analysis of our data set shows that negative examples account for a small portion of the total book ratings by the reviewers. For each reviewer, we created a testing data set by randomly selecting 40% of the negative examples and an identical number of positive examples to avoid any potential invalid results caused by prediction biases in the recommender system (i.e., tendency toward predicting an example would be positive or negative). At the same time, we randomly sampled 40% of the remaining positive examples to form positive examples and took the combination of the remaining positive and negative examples as the unlabeled examples. For example, if the original data set has 1,000 positive examples and 200 negative examples, we created a balanced testing data set by sampling 40% of negative examples (i.e., 80 negative examples) and 80 positive examples, which generates an identical number of positive and negative examples in the testing data set. We then sample 40% of remaining positive examples (i.e., 920) to serve as positive examples (i.e., 368), with the remaining positive (552) and negative (120) examples as unlabeled examples.

In our evaluation, PNB and PEBL serve as the performance benchmarks. To minimize potential biases resulting from the randomized sampling and to obtain accurate and robust performance estimates, we performed the evaluation process 10 times for each reviewer. That is, the effectiveness of each system (COLPEL, PNB, and PEBL) derives from the average performance recorded for each reviewer over 10 trials. Because the sheer volume of the testing data differs among reviewers, we determine the overall effectiveness of each system using the weighted average performance across the 36 reviewers, taking into account the number of testing examples received by the respective reviewers in relation to the total number of testing examples.

The accuracy measure conveys how well each system performs with respect to different outcome classes. The F1 metric is a common measure for assessing prediction accuracy, calculated as follows: $\frac{2\times\text{recall}\times\text{precision}}{(\text{recall}+\text{precision})}$. This measure provides a single-value summary of both recall and precision and their intrinsic trade-off. To evaluate the effectiveness of each system with respect to each outcome class (positive or negative), we used the F1 score for the positive class and the F1 score for negative class; together, they provide more insights into their relative effectiveness for supporting single-class learning about product recommendations in online retailing.

Parameter-Tuning Experiments

Before our evaluation, we conducted a series of parameter-tuning experiments to select an optimal value for each parameter required by the respective systems. We randomly sampled three reviewers and used their preference records for the parameter tuning required by each system. The PNB requires a parameter λ to smooth the maximum likelihood estimate of the word probability, given a class. To identify strong positive features, we need two parameters for the rough classifier in the mapping stage of PEBL: α_{PT} (the lower-bound threshold in positive examples) and α_{UT} (upper-bound threshold in unlabeled examples). Finally, COLPEL requires two additional parameters: the weight of constant Z uses in cost-proportionate rejection sampling (β) and the number of classifiers to be constructed (k).

Parameter Tuning for PNB

For PBN, we estimated $\hat{P}(C_p)$, the a priori probability of the positive class, for each reviewer using the percentage of their respective positive examples in the training data set and then examined λ between 0.3 to 3.3, in increments of 0.3. As we show in Table 1, as λ increases, prediction accuracy remains steady; there seems to be a trade-off between the positive F1 and negative F1 measures. However, the average F1 score improves marginally as λ increases from 0.3 to 0.9, then deteriorates as λ increases further. Therefore, we set λ to 0.9, at which point PNB reaches the highest average F1 score.

Table 1. Parameter Tuning for PNB: Effects of λ

λ	Accuracy	Positive F1	Negative F1	Average F1
0.3	50.1%	0.472	0.525	0.498
0.9	51.2%	0.440	0.566	0.503
1.5	51.2%	0.393	0.589	0.491
2.1	51.1%	0.353	0.603	0.445
2.7	50.9%	0.315	0.614	0.416
3.3	50.7%	0.276	0.623	0.382

Parameter Tuning for PEBL

For PEBL, we examined α_{PT} between 0% to 20% in increments of 5%; we assessed α_{UT} from 10% to 50% in increments of 10%. We first set the value of α_{UT} to 30% and examine the effects of α_{PT}. As Table 2 shows, as α_{PT} increases, prediction accuracy remains steady, and we observe a trade-off between the positive and negative F1 measures. In addition, the average F1 score improves considerably from 0% to 15% and reaches a high score of 0.48. Accordingly, we set α_{PT} to 0.15 to examine the effects of α_{UT}.

Table 2. Parameter Tuning for PEBL: Effects of α_{PT}

α_{PT}	Accuracy	Positive F1	Negative F1	Average F1
0%	50.0%	0.666	0.001	0.333
5%	49.0%	0.646	0.082	0.364
10%	48.6%	0.576	0.329	0.452
15%	50.1%	0.397	0.562	0.480
20%	50.5%	0.128	0.651	0.389

We summarize the experimental results for α_{UT} in Table 3. Similar to α_{PT}, the increase of α_{UT} has a marginal effect on prediction accuracy; there seems to be a trade-off between positive F1 and negative F1 measures. However, setting α_{UT} lower than 10% appears to create a serious prediction bias, because PEBL predicts all examples as belonging to the negative class. On the basis of our parameter tuning experimental results, we set α_{PT} to 15% and α_{UT} to 30%, at which levels PEBL seems to achieve the best performance over the range of the parameter values we examine.

Table 3. Parameter Tuning for PEBL: Effects of α_{UT}

α_{UT}	Accuracy	Positive F1	Negative F1	Average F1
0%	50.0%	0.000	0.666	0.333
10%	50.0%	0.001	0.666	0.333
20%	50.1%	0.397	0.562	0.480
30%	49.9%	0.511	0.474	0.493
40%	48.9%	0.587	0.326	0.457
50%	48.9%	0.611	0.244	0.428

Parameter Tuning for COLPEL

For COLPEL, we examine β between 1.0 and 2.0 in increments of 0.1, as well as k from 3 to 11 in increments of 2. The value range of α_{PT} and α_{UT} examined and the procedure used are identical to those for PEBL. We first fixed the values of β and k and determined the respective values of α_{PT} and α_{UT}. We set α_{PT} to 0% and α_{UT} to 40%, at which level COLPEL achieves the best performance over the range of the parameter values under examination. We then set k to 7 and examined the effects of β. As Table 4 reveals, the effects of β seem marginal for both prediction accuracy and the average F1 measure. Thus, we finally set β to 1.5; COLPEL seems to arrive at a relatively better effectiveness with this value.

Table 4. Parameter Tuning for COLPEL: Effects of β

β	Accuracy	Positive F1	Negative F1	Average F1
1.1	50.3%	0.382	0.578	0.480
1.3	51.2%	0.451	0.541	0.496
1.5	50.8%	0.501	0.503	0.502
1.7	49.0%	0.525	0.438	0.481
1.9	49.0%	0.560	0.387	0.473

We also examine the effects of *k* on the effectiveness of COLPEL. In Table 5, we show that the number of classifiers has no obvious effect on the performance of COLPEL. We select 7 as a value for *k*, which seems appropriate because COLPEL obtains a relatively better prediction accuracy and average F1 measure at this point. In line with our experimental results, we set α_{PT} to 0%, α_{UT} to 40%, β to 1.5, and *k* to 7.

Table 5. Parameter Tuning for COLPEL: Effects of *k*

k	Accuracy	Positive F1	Negative F1	Average F1
3	49.6%	0.496	0.502	0.499
5	51.0%	0.496	0.505	0.501
7	51.1%	0.500	0.511	0.505
9	50.2%	0.492	0.501	0.497
11	48.2%	0.493	0.499	0.496

Evaluation Results

Using these parameter values, we evaluate and compare the effectiveness of COLPEL, PNB, and PEBL. As we summarize in Table 6, COLPEL outperforms PNB and PEBL in terms of the overall prediction accuracy. The overall F1 score attained by COLPEL is higher than that by PNB or PEBL, though COLPEL has a lower positive F1 score than that of PEBL and a lower negative F1 score than that of PNB. In addition, the performance of PNB and PEBL are comparable in terms of their overall prediction accuracy and overall F1 score. However, we notice that PNB arrives at the worst positive F1 (or positive recall) score and the best negative F1 (or negative recall) score among the investigated approaches; that is, it appears that PNB tends to predict examples as negative. The prediction bias of PBN might result from the predomination of unlabeled examples, which likely affect the estimation of a priori probability. Furthermore, COLPEL arrives at the best scores in both positive and negative precisions and offers some advantages over PNB and PEBL in terms of predicting both positive and negative examples.

Table 6. Comparative Evaluation Results

	Accuracy	Positive F1	Positive Precision	Positive Recall	Negative F1	Negative Precision	Negative Recall	Average F1
PNB	52.4%	0.381	0.562	0.288	0.615	0.516	0.762	0.498
PEBL	50.0%	0.529	0.501	0.561	0.466	0.496	0.439	0.498
COLPEL	56.0%	0.517	0.586	0.462	0.604	0.557	0.659	0.560

We test the performance differences further by focusing on the prediction accuracy and average F1 score of the respective systems. As we show in Table 7, COLPEL outperforms PNB and PEBL in both prediction accuracy and the average F1 measure, and the performance differentials are statistically significant ($p< 0.01$). However, the performance differences between PNB and PEBL, measured by prediction accuracy and average F1 value, are not statistically significant ($p > 0.1$). According to our evaluation results, COLPEL is advantageous compared with PNB and PEBL, which both seem comparable in their performance.

Table 7. Significant Test (*p*-Value)

(a) Prediction Accuracy

	PBN	PEBL	COLPEL
PNB			
PEBL	0.481		
COLPEL	0.000***	0.000***	

(b) Average F1 Measure

	PBN	PEBL	COLPEL
PNB			
PEBL	0.482		
COLPEL	0.000***	0.000***	

*** Significant at $p < 0.01$ on a two-tailed t-test.

Effect Analysis for Cost-Proportionate Rejection Sampling

To analyze the effect of the cost-proportionate rejection sampling used by COLPEL, we replace it with random sampling and implement COLPEL$_{\text{random}}$. We randomly assign each unlabeled example a number between 0 and 1 and use the examples with numbers greater than 0.5 to form a set of negative examples; thus, all unlabeled examples have a uniform misclassification cost. To evaluate COLPEL$_{\text{random}}$, we follow the same evaluation procedure as described for COLPEL, but the effectiveness of COLPEL is better than that of COLPEL$_{\text{random}}$ in terms of prediction accuracy and average F1 measure, according to the results in Table 8. In addition, COLPEL$_{\text{random}}$ has a strong tendency to predict the negative class and, in most cases, fails to recommend positive examples. Consequently, it has a low positive recall score. Our proposed misclassification cost function therefore appears effective for measuring the probability of unlabeled as negative examples; the cost-proportionate rejection sampling in COLPEL seems advantageous for making recommendations.

Table 8. Effects of Cost-Proportionate Rejection Sampling

	Accuracy	Positive F1	Positive Precision	Positive Recall	Negative F1	Negative Precision	Negative Recall	Average F1
COLPEL$_{\text{random}}$	50.7%	0.117	0.541	0.066	0.668	0.503	0.952	0.388
COLPEL	56.0%	0.517	0.586	0.462	0.604	0.557	0.659	0.560

Effect Analysis for the Committee Machine

Our proposed COLPEL technique also adopts a committee machines strategy, using k classifiers that jointly make the recommendation decision to reduce the prediction bias associated with using a single classifier. We examine the effects of the committee machine on the effectiveness of COLPEL by implementing COLPEL$_{\text{single}}$, which uses only one classifier to determine whether to recommend an example. As the data in Table 9 show, COLPEL outperforms COLPEL$_{\text{single}}$ in terms of its prediction accuracy and positive and average F1 measures. Although COLPEL is less effective than COLPEL$_{\text{single}}$ for identifying negative examples (i.e., lower negative recall), it is more precise in its predictions, which indicates it has a higher negative precision. Overall, the committee machine strategy benefits COLPEL by enabling to it to make more effective recommendations.

Table 9. Effects of Committee Machine

	Accuracy	Positive F1	Positive Precision	Positive Recall	Negative F1	Negative Precision	Negative Recall	Average F1
$\text{COLPEL}_{\text{single}}$	54.2%	0.440	0.573	0.357	0.613	0.529	0.728	0.527
COLPEL	56.0%	0.517	0.586	0.462	0.604	0.557	0.659	0.560

Conclusion

Content-based filtering can make appropriate product recommendations by analyzing important data about products (e.g., features) and customers (e.g., demographics, preferences, browsing and purchase behaviors), without reference to other customers. Such content-based recommendations require the support of a supervised learning technique that constructs a classification model from a training sample with a sufficiently large number of cases that belong to different outcome classes. However, such training samples are not always available. Conventional classification techniques become ineffective in recommendation or prediction scenarios characterized by a training sample that consists of only positive examples and a much greater number of unlabeled examples, which is commonly known as the single-class learning challenge. We propose a cost-sensitive, learning-based, positive example learning method (i.e., COLPEL) that adopts cost-sensitive learning and committee machines to address the limitations inherent to salient single-class learning techniques. Our analysis suggests the use of cost-proportionate rejection sampling can enhance the construction of effective classifiers, because it identifies highly probable negative examples. The committee machines (i.e., use of multiple classifiers to co-decide recommendations) can improve the effectiveness of COLPEL. Overall, our evaluation results show that COLPEL outperforms the benchmarks PNB and PEBL, as measured by accuracy, positive F1 scores, and negative F1 scores.

This study contributes to automated recommendation research in general and to single-class learning for product recommendation in particular. We propose COLPEL to address the limitations of previously proposed approaches that have been developed for automated classification using only positive examples and more unlabeled examples. We empirically evaluate the effectiveness of a recommender system built on the proposed cost-sensitive, learning-based, positive example learning approach, with PNB and PEBL as performance benchmarks. According to our results, the system using COLPEL is more effective than either PNB or PEBL. In many real-world situations, misclassification costs associated with learning examples are not readily available and must be assigned manually; in this study, we design a measure to estimate the misclassification cost of each unlabeled example and thus facilitate the use of cost-sensitive learning in these situations. Finally, our proposed COLPEL effectively addresses the single-class learning challenge for automated recommender systems and could be extended and applied to other scenarios with similar characteristics, such as filters for spam messages or clinical diagnosis support.

Finally, this study has several limitations that in turn suggest some interesting future research directions worthy of attention. First, the COLPEL approach uses the cost-proportionate rejection sampling method to draw a set of negative examples from the unlabeled examples on the basis of their misclassification costs. Therefore, the assignment of proper misclassification costs to each unlabeled example might be key to inducing an even more effective learning classifier. Second, this study considers customer preferences static. However, a customer's preference may change constantly, which demands appropriate adaptive capability in the underlying single-class learning supporting the content-based recommendations. Third, our study includes PNB-based and PEBL-based systems for performance benchmark purposes; additional studies could consider other common approaches for evaluation baselines. In addition, this study only considers monolingual content-based recommendations, so an important extension would include cross-lingual or multilingual recommendations that involve positive, unlabeled training examples pertinent to products or services that may be available in different languages.

References

Adomavicius, G. and 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* (17:6), 2005, pp.734-749.

Agrawal, R., Bayardo, R., and Srikant, R., "Athena: Mining-based Interactive Management of Text Databases," in *Proceedings of the Seventh Conference on Extending Database Technology*, Konstanz, Germany 2000, pp.365-379.

Alspector, J., Kolcz, A., and Karunanithi, N., "Comparing Feature-based and Clique-based User Models for Movie Selection," in *Proceedings of the Third ACM Conference on Digital Libraries*, Pittsburgh, PA, 1998, pp. 11-18.

Ampazis, N., "Collaborative Filtering via Concept Decomposition on the Netflix Dataset," in *Proceedings of the 18th European Conference on Artificial Intelligence (ECAI 2008): Workshop on Recommender Systems*, Patras, Greece, 2008, pp. 26-30.

Balabanovic, M. and Shoham, Y., "Fab: Content-based, Collaborative Recommendation," *Communications of the ACM* (40:3), 1997, pp. 66-72.

Basilico J. and Hofmann T., "Unifying Collaborative and Content-based Filtering," in *Proceedings of the 21st International Conference on Machine learning* (69), Banff, Alberta, Canada, 2004.

Calvo, B., Larranaga, P., and Lozano, J., "Learning Bayesian Classifiers from Positive and Unlabeled Examples," *Pattern Recognition Letters* (28:16), 2007, pp. 2375-2384.

Cheng, T. H., and Hu, P. J. H., "Content-based Recommendations Using Positive-Only Examples: A Single-Class Learning Approach," in *Proceedings of Sixth Workshop on e-Business*, 2007, pp. 857-868.

Denis, F., Gilleron, R., and Tommasi, M., "Text Classification from Positive and Unlabeled Examples," in *Proceedings of Ninth International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, Annecy, France, 2002, pp. 1927-1934.

Dewan, R., Jing, B., and Seidmann, A., "Adoption of Internet-Based Product Customization and Pricing Strategies," *Journal of Management Information Systems* (17:2), 2000, pp. 9-28.

Domingos, P., "MetaCost: A General Method for Making Classifiers Cost Sensitive," in *Proceedings of the Fifth International Conference on Knowledge Discovery and Data Mining*, 1999, pp. 155-164.

Drummond, C. and Holte, R., "Exploiting the Cost (In)Sensitivity of Decision Tree Splitting Criteria," in *Proceedings of the Fifth International Conference on Knowledge Discovery and Data Mining*, 1999, pp. 155-164.

Dumais, S., Platt, J., Heckerman, D., and Sahami, M., "Inductive Learning Algorithms and Representation for Text Categorization," in *Proceedings of the ACM 7th International Conference on Information and Knowledge Management*, Bethesda, MD, 1998, pp.148-155.

e-tailing.com, "Merchant Secrets for Driving Conversion," 2007. (available at http://www.e-tailing.com/graphics/ms_ResearchReport05.pdf.)

Fan, W., Stolfo, S., Zhang, J., and Chan, P., "AdaCost: Misclassification Cost-sensitive Boosting," in *Proceedings of the 16th International Conference on Machine Learning*, 1999, pp. 97-105.

Goldberg, D., Nichols, D., Oki, B., and Terry, D., "Using Collaborative Filtering to Weave an Information Tapestry," *Communications of the ACM* (35), 1992, pp. 61-70.

Haubl, G. and Trifts, V., "Consumer Decision Making in Online Shopping Environments: The Effects of Interactive Decision Aids," *Marketing Science* (19:1), 2000, pp. 4-21.

Herlocker, J. L., Konstan, J. A., Borchers, A., and Riedl, J., "An Algorithmic Framework for Performing Collaborative Filtering," in *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Berkeley, CA, 1999, pp. 230-237.

Hostler, R. E., Yoon, V. Y., and Guimaraes, T., "Assessing the Impact of Internet Agent on End Users' Performance," *Decision Support Systems* (41:1), 2005, pp. 313-323.

Kittler, J., "Mathematical Methods of Feature Selection in Pattern Recognition," *International Journal of Man-Machine Studies* (7), 1975, pp.609-637.

Krulwich, B. and Burkey, C., "Learning User Information Interests through Extraction of Semantically Significant Phrases," in *Proceedings of the AAAI Spring Symposium on Machine Learning in Information Access*, Stanford, CA, 1996.

Lam, W. and Ho, C. Y., "Using a Generalized Instance set for Automatic Text Categorization," in *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Melbourne, Australia, 1998, pp. 81-89.

Lang, K., "NewsWeeder: Learning to Filter Netnews," in *Proceedings of the 12th International Conference on Machine Learning*, San Francisco, CA, 1995, pp. 331-339.

Lee, B. K. and Lee, W. N., "The Effect of Information Overload on Consumer Choice Quality in An On-line Environment," *Psychology & Marketing* (21:3), 2004, pp. 159-183.

Lewis, D. and Ringuette, M., "A Comparison of Two Learning Algorithms for Text Categorization," in *Proceedings of Symposium on Document Analysis and Information Retrieval*, Las Vegas, NV, 1994.

Margineantu, D., "Class Probability Estimation and Cost-sensitive Classification Decisions," in *Proceedings of the 13th European Conference on Machine Learning*, 2002, pp. 270-281.

Modrzejewski, M., "Feature Selection Using Rough Sets Theory," in *Proceedings of the Fourth European Conference on Machine Learning*, 1993, pp.213-226.

Mooney, R. J. and Roy, L., "Content-Based Book Recommending Using Learning for Text Categorization," in *Proceedings of the Fifth ACM Conference on Digital Libraries*, 2000, pp. 195-204.

Murthi, B. P. S. and Sarkar, S. "The Role of the Management Sciences in Research on Personalization," *Management Science* (49:10), 2003, pp. 1344-1362.

Ng, H. T., Goh, W. B., and Low, K. L., "Feature Selection, Perceptron Learning, and a Usability Case Study for Text Categorization," in *Proceedings of Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '97)*, Philadelphia, PA, 1997, pp.67-73.

Pazzani, M. and Billsus, D., "Learning and Revising User Profiles: The Identification of Interesting Web Sites," *Machine Learning* (27), 1997, pp. 313-331.

Perez, J. C., "U.S. e-Tailing Sales to Have Brisk Growth through 2010," 2006. (available at http://www.infoworld.com/article/06/02/07/75135_HNetailingsales_1.html?B-TO-C).

Piramuthu, S., "Evaluating Feature Selection Methods for Learning in Data Mining Applications," in *Proceedings of Thirty-First Annual Hawaii International Conference on System Sciences*, Hawaii, 1998.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J., "Learning Internal Representations by Error Propagation," *In Parallel Distributed Processing Explorations in the Microstructures of Cognition* (1), D. E. Rumelhart and J. L. McClelland (Eds.), MIT Press, Cambridge, MA, 1986, pp. 318-362.

Sanchez-Hernandez, C., Boyd, D. S., and Foody, G. M., "One-class Classification for Mapping a Specific Land-cover Class: SVDD Classification of Fenland," *IEEE Transaction on Geoscience and Remote Sensing* (45:4) April 2007, pp. 1061-1073.

Senecal, S. and Nantel, J., "The Influence of Online Product Recommendations on Consumers' Online Choices," *Journal of Retailing* (80), 2004, pp. 159-169.

Shardanand, U., and Maes, P, "Social Information Filtering: Algorithms for Automating Word of Mouth," *Human Factors in Computing Systems*, ACM CHI, 1995, pp. 210-217.

Siedlecki, W. and Sklansky, J., "A Note on Genetic Algorithms for Large-scale Feature Selection," *Pattern Recognition Letters* (10:5), 1989, pp. 335-347.

Tax, D. and Duin, R., "Support Vector Data Description," *Machine Learning* (54:1), 2004, pp. 45-66.

TrendGo Research and Survey Company, "2007 Taiwan Publishers Marketing Survey," Government Information Office of Taiwan, 2007. (available at http://www.gio.gov.tw/info/publish/2007market_e/)

von Neumann, J., "Various Techniques Used in Connection with Random Digits," *National Bureau of Standards, Applied Mathematics Series* (12), 1951, pp. 36-38.

Wei, C. P., Chen, H. C., and Cheng, T. H., "Effective Spam Filtering: A Single-Class Learning and Ensemble Approach," *Decision Support Systems* (45), 2008, pp. 491-503.

Wei, C. P., Shaw, M. J., and Easley, R. F., "A Survey of Recommendation Systems in Electronic Commerce," in *E-Service: New Directions in Theory and Practice*, R. T. Rust and P. K. Kannan (Eds.), M. E. Sharpe Publisher, 2002.

Winters, T., King, C., and Davie, W., "Quarterly Retail E-Commence Sales," *U.S. Census Bureau News*, US Department of Commence, November 2008. (available at http://www.census.gov/mrts/www/ecomm.html)

Xiao, B. and Benbasat, I., "E-Commerce Product Recommendation Agents: Use, Characteristics, and Impact," *MIS Quarterly* (31:1), 2007, pp. 137-209.

Yang, Y. and Chute, C. G., "An Example-Based Mapping Method for Text Categorization and Retrieval," *ACM Transactions on Information Systems* (12:3), 1994, pp. 252-277.

Yu, H., Han, J., and Chang, K. C. C., "PEBL: Web Page Classification without Negative Examples," *IEEE Transaction on Knowledge and Data Engineering* (16:1), 2004, pp. 70-81.

Zadrozny, B., Langford, J., and Abe, N., "Cost-Sensitive Learning by Cost-Proportionate Example Weighting," in *Proceedings of the Third IEEE International Conference on Data Mining*, 2003, pp. 435-442.