Customer Targeting Models Using Actively-Selected Web Content

Prem Melville
IBM T.J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598
pmelvil@us.ibm.com

Saharon Rosset[*]
IBM T.J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598
srosset@us.ibm.com

Richard D. Lawrence
IBM T.J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598
ricklawr@us.ibm.com

ABSTRACT

We consider the problem of predicting the likelihood that a company will purchase a new product from a seller. The statistical models we have developed at IBM for this purpose rely on historical transaction data coupled with structured firmographic information like the company revenue, number of employees and so on. In this paper, we extend this methodology to include additional text-based features based on analysis of the content on each company's website. Empirical results demonstrate that incorporating such web content can significantly improve customer targeting. Furthermore, we present methods to actively select only the web content that is likely to improve our models, while reducing the costs of acquisition and processing.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning; I.5.1 [**Pattern Recognition**]: Models

General Terms

Algorithms, Economics, Experimentation, Management

Keywords

Text categorization, web mining, active learning, active feature-value acquisition

1. INTRODUCTION

Over the past decade, the Internet has become an increasingly dominant channel for companies of all sizes to transact business. Just as importantly, companies now use their websites to communicate essentially everything that they want the world to know about them. Companies that sell products go to great lengths to describe these products with the hope of connecting to potential buyers through search engines and sales bots. Companies that sell services similarly describe their offerings with the same end objective. In building increasingly comprehensive websites, literally millions of companies around the world are implicitly communicating insight about their own business requirements and strategies. In comparison with information that can be purchased from various third-party data vendors, this insight is likely to be more reliable, because it is generated directly by the company in question.

Indeed, if properly analyzed, this web content can be of enormous value to another class of companies, namely those that sell goods and services to other companies. Like any company that sells to consumers, these business-to-business sellers seek to target their offerings to segments of customers (companies in this case) that meet certain firmographic requirements like industry (e.g. "Financial Services") and company size (e.g. "annual revenue between $50M and $500M"). Many current analytical approaches to this targeting problem are built largely on such firmographic segments. As an example, let us assume that we are selling a risk management application that is designed to integrate other, existing risk models. We could target all companies in the above firmographic segment. However, it is very likely that companies with a strong relevance to "risk management" on their web site would be better prospects. Hence, it seems very plausible that the utility of such targeting models could be greatly enhanced by augmenting the firmographic view with what the company says about itself on its website.

In this paper we present a case study which examines the contribution of website content to improving the performance of our existing targeting models, supporting the search for new customers for IBM products. IBM offers its customers a wide range of Information Technology (IT) products, ranging from servers like System P, through corporate software offerings like Rational, to high-end services in IT and business transformation. We have developed a targeting tool called OnTarget to help in sales efforts of these products, which has been successfully deployed in the last several years. This tool offers *propensity estimates*, indicating the likelihood that the customer will buy the product. These estimates are the results of predictive modeling efforts on the combined information from IBM-internal databases (containing mainly sales history) and *firmographics*, that is, information on the companies which are potential IBM customers, which is purchased from data collection agencies

[*]Current address: Department of Statistics, Tel Aviv University, Israel

KDD'08, August 24–27, 2008, Las Vegas, Nevada, USA.
Copyright 2008 ACM 978-1-60558-193-4/08/08 ...$5.00.

such as Dun & Bradstreet (*http:www.dnb.com*) and Standard and Poor (*http:www.standardandpoor.com*). Several aspects of the methodology and results of OnTarget are discussed in our previous publications [15, 9].

An important distinction in OnTarget is between two types of propensity models:

- **Whitespace** models, which estimate the propensity of companies *that are not currently IBM customers* to buy each product. These models can only take advantage of firmographic data, since they aim to predict propensity for companies that do not have a prior relationship with IBM, and hence no useful data in the internal IBM databases.
- **Existing customer** models, which estimate the propensity of established IBM customers to buy new products from IBM. These models can take advantage of the rich information about these customers and their historical relationship with IBM in the internal databases.

It is clearly expected that the use of website content for improving predictive performance will bring much more value in whitespace models than in existing customer models, and so in this paper we concentrate on examining the effect of utilizing web content on the quality of OnTarget whitespace models. We offer two main contributions:

- We present a methodology for crawling websites of potential customers, extracting information, and integrating it into the predictive modeling tool, and demonstrate that it does indeed generate a significant improvement in the predictive performance of our models.
- Given the significant processing and modeling costs associated with using web content, we implement *active instance completion* [11, 12], which selectively extracts web content for only a subset of the companies in the modeling universe. We demonstrate that most (sometimes all) of the benefit of acquiring web-content information can be achieved when it is generated for only a small subset of the companies, and that the active feature acquisition algorithms we adapt from the literature carry significant benefit over acquisition of random subsets of all companies.

2. MODELING SETUP

In this section, we describe the relevant details of the OnTarget application and the models it builds. As mentioned above, more details can be found in [9] and references therein.

The goal of the propensity models is to differentiate customers (or potential customers) by their likelihood of purchasing various IBM products. The whitespace prediction models created in OnTarget currently estimate propensity to buy for ten IBM product families (or *brands*). There are five software brands: DB2, Lotus, Rational, Tivoli and Websphere — covering a wide range of corporate software areas from database management (DB2), through software development (Rational) to application and transaction infrastructure (Websphere). There are also four hardware brands, representing families of server products: Series I, Series P, Series X and Series Z. Finally, OnTarget also predicts propensity to buy for IBM's Storage solutions brand.

With multiple geographic areas (the Americas, Europe, Asia Pacific), multiple countries within each geographic area, and multiple product brands, a large number of propensity models (currently about 160) are built in each quarter. Given a geographic area and a brand Y, our first step is to identify positive examples and negative examples to be used for modeling. In the whitespace modeling problem, we want to try to understand what drives the decision to purchase brand Y by companies that are not currently IBM customer, and to delineate companies by the likelihood of their purchase. Assuming a time period t, we formulate our modeling problem as:

> Differentiate companies that had never bought from IBM until period t, then bought brand Y during this period, from companies that have never bought from IBM.

The time period t is typically the most recent one or two years. Thus, for the whitespace problem, our positive and negative examples are:

- **Positive:** Companies that had never bought from IBM before t, then bought Y during t.
- **Negative:** Companies that had never bought from IBM before or during t.

Companies that do not fall into either one of these categories play no role in this modeling problem.

Next, we define the variables to be used in modeling. These come from the D&B firmographic data, including:

- Company size indicators (revenue, employees), both in absolute and relative terms (rank within industry)
- Industry variables, both raw industry classification from D&B and derived sector variables
- Company location in corporate hierarchy (e.g., headquarters or subsidiary)

We then use these datasets to build prediction models, using mostly logistic regression, and these generate the predictions presented in OnTarget. We refer to these firmographic-based Whitespace models as *Firmographic* models in the rest of the paper.

OnTarget models are typically evaluated using cross-validated lift curves or ROC curves. These approaches have the advantage that they are statistically stable, and that they evaluate the performance of the models in *ranking* the companies correctly according to their real propensity to buy; which reflects well the OnTarget goal of being a sales aid, complementing the sales representative's personal knowledge with reliable rankings of the potential customer among its peers.

2.1 Data sets

For the experiments in this paper, we chose to focus whitespace propensity models for two brands *Rational* and *Websphere*. Following the process outlined above, we obtained data sets for these modeling problems. The period t for definition of positives and negatives was two years between 7/1/05 and 6/30/07. The resulting data set for *Rational* has a total of 506 companies for which we have an identified website, and of them, 77 are positive examples and the other 429 are negatives. The corresponding numbers for *Websphere* are 494 (total) and 65 (positives).

3. ANALYZING WEB CONTENT

As mentioned earlier, a company's website is a rich source of information for potential sellers. So, we begin by analyzing what companies say on the their websites, and the relevance of this content to identifying new customers for our specific brands.

For each company in our data, we crawl the corresponding website up to a depth of 4, and merge the content from all downloaded HTML documents into one. In case the combined content of a website exceeds one megabyte, we restrict ourselves to the first megabyte of content, in order to filter possible noise from too many irrelevant pages. We then pre-process the text by removing stop words, stemming the words into inflected forms, and filtering out words that appear in less than three web pages. Each company can now be represented as a document in a fixed vocabulary, which we convert into word vectors using the bag-of-words representation with TF-IDF term weighting [1].

To examine relevant terms, we rank-ordered the 20,687 unique terms from the *Rational* data set using χ^2 scores [18]. The 20 (stemmed) terms that are most discriminative in identifying positive examples are listed below:
interfac, enabl, deploi, scalabl, integr, deploy, simplifi, autom, multipl, platform, configur, sophist, workflow, leverag, interoper, enterpris, proposit, softwar, partner, strateg

IBM Rational software "offers industry leading proven software development tools and processes to enhance ... application development projects"[1]. Clearly, the terms identified above are consistent with what one might expect to find on the websites of companies that are heavily involved in software development. Hence, the web content does indeed contain very relevant terms that would be helpful in identifying new customers for Rational.

We repeated this exercise for the 19,718 unique terms in the *Websphere* data set, and the top ranked words are:
payment, knowledg, electron, elig, transact, surg, sql, insur, desir, ensur, banker, equiti, mastercard, backup, purpos, check, scalabl, provid, retriev, benefit

IBM Websphere software "provides a next-generation solution for all of a company's e-commerce needs"[2]. These keywords clearly indicate that the websites of Websphere customers are used to support electronic transactions involving payment by credit cards such as Mastercard (we note that Visa appears further down the list). Again, the focus of these *Websphere* positive examples, as measured by the relevant keywords, is well-aligned with the capabilities provided by the Websphere product line.

In order to build our propensity models we collect recent firmographic and web content for companies. This data collection is done after the positives (actual buyers) have been identified using the historical transactional data. Ideally, we want to build models using the firmographic and web data that existed *prior* to the purchase. However, this is not feasible since we are evaluating our modeling techniques on historical data. Hence, there is a chance that this data has changed from its state prior to the purchase that produced a positive example. For example, the word "websphere" can occur in the web content simply because that site is deployed using the Websphere Application Server as a result of a prior Websphere purchase. Thus, in order to reduce the risk of being biased by websites that may have changed after adoption of our products, we explicitly remove the terms "rational" and "websphere" from the raw web content.

3.1 Web-content propensity models

Given the web content described above, we cast the task of propensity modeling into one of text categorization, i.e., given a text document representing a company, classify it as a positive or negative example of a potential customer. We can now use one of many text classification methods available to solve this problem. In particular, we experimented with two popular approaches. The first approach is SVM-light [7], which is an efficient and scalable implementation of Support Vector Machines for text classification. The second approach is Naïve Bayes using a multinomial text model[10].

We also ran versions of the classification algorithms modified to deal with the high imbalance between the positive and negative class. SVM-light provides a straightforward mechanism for dealing with class imbalance by specifying a cost factor by which training errors on positive examples outweigh errors on negative examples. For *Rational*, we set this cost factor to 6.5, and refer to this variant as SVM(c=6.5). In the case of Naïve Bayes, we re-weighted the instances in the training data so that a positive instance has 6.5 times the weight of a negative instance (which corresponds to the imbalance in this data set). We refer to this approach as Naïve Bayes (c=6.5). To understand the effect of this reweighting, let us denote the learning sample for our propensity model by $\{\mathbf{x}_i, y_i\}_{i=1}^n$, with $\mathbf{x}_i$ the vector of p features for the instance i and $y_i \in \{0, 1\}$ the binary response. Then looking at the Naïve Bayes formula:

$$\hat{P}(Y=1|\mathbf{x}) = \frac{\Pi_{j=1}^p \hat{P}_j(x_j|Y=1)\cdot\hat{\pi}_1}{\Pi_{j=1}^p \hat{P}_j(x_j|Y=1)\hat{\pi}_1 + \Pi_{j=1}^p \hat{P}_j(x_j|Y=0)\hat{\pi}_0}$$

where *hats* above the quantities mean they are estimated from the data and $\hat{\pi}_1, \hat{\pi}_0$ are the percentage of positive and negative examples in the sample, respectively. By reweighting we are only affecting the $\hat{\pi}$ quantities, and in fact making them equal, leading to a "likelihood ratio" type probability estimate:

$$\hat{P}_{c=6.5}(Y=1|\mathbf{x}) = \frac{\Pi_{j=1}^p \hat{P}_j(x_j|Y=1)}{\Pi_{j=1}^p \hat{P}_j(x_j|Y=1) + \Pi_{j=1}^p \hat{P}_j(x_j|Y=0)}$$

which basically examines which class makes the observed sequence more likely, compared to the original formulation, which significantly downweights the contribution of the smaller class in the denominator.

3.2 Feature selection

Selecting a subset of features for training can significantly speed up training text classification models, and quite often can improve classification accuracy. A simple approach to feature selection for text is using Document Frequency, which ranks words for selection based on the number of documents in which the word appears. Past studies have shown that this approach is very effective, often comparable to more computationally intensive approaches, such as χ^2 feature selection [18, 5, 13].

We compare the different text classification algorithms described in the previous section, at different levels of feature selection. Figure 1 shows performance on *Rational* of the different models in terms of area under the ROC curve (AUC)

[1] *http:www.ibm.com/software/rational*

[2] *http:www.ibm.com/software/websphere*

on data with increasing dimensionality. The SVM models perform quite well and are not affected much by feature selection, except when we drop below 2500 features. This seems to be consistent with the observation by Zheng et al. [20], that regularized linear methods, such as SVMs are not helped by standard feature selection methods. Naïve Bayes, on the hand seems to be quite sensitive to the selection of features. The re-weighted Naïve Bayes is also affected, but to a smaller extent. Performance on Naïve Bayes, in general, improves with fewer selected features, up until we are left with 2500 features. The computational complexity of the Naïve Bayes algorithm is linear in the size of the vocabulary. So reducing the feature space to 2500 features speeds up the text classification 8 times, and additionally produces better models.

Naïve Bayes trained on imbalanced data produces predictions that are biased in favor of large classes, as noted by Rennie et al. [14] and Frank and Bouckaert [6]. Re-weighting instances, as we do for Naïve Bayes (c=6.5) does indeed significantly improve on the equally weighted Naïve Bayes, and furthermore, outperforms SVMs on this data. For the rest of this paper we use this re-weighted Naïve Bayes approach for the *Web-content* models.

For *Websphere*, using Naïve Bayes (c=10) produces the best models when applied to the full feature set. Performance on AUC does not drop much when the dimensionality is reduced up until 5000 features. However, to get the best propensity models in the next section, we will use the entire feature set for *Websphere*.

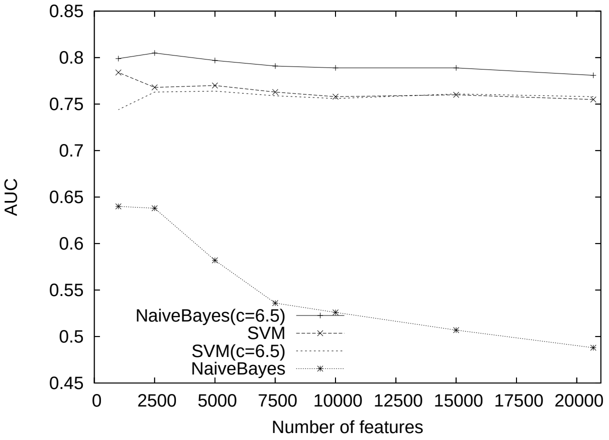

Figure 1: Comparison of text classification methods with feature selection on *Rational*.

4. INCORPORATING WEB CONTENT INTO FIRMOGRAPHIC MODELS

In Section 2 we described the generation of structured *Firmographic* models, and in Section 3 we described the use of unstructured text content for building *Web-content* models. In this section, we describe different approaches to combining the structured and unstructured features in order to build more effective propensity models.

We compare the following approaches of incorporating web content into our structured *Firmographic* model.

Voting: In this approach, we train separate *Firmographic* and *Web-content* models on the same training instances, and then combine the class probability estimates predicted by both models on test instances. We consider two variants of this approach: 1) Vote-Avg, in which probability estimates produced by individual models are simply averaged; and 2) Vote-Prod, in which probability estimates produced by individual models are multiplied and renormalized.

Nesting: In this approach, we use the output of the *Web-content* model as an input to the Logistic Regression *Firmographic* model. Specifically, we add another variable in the Logistic Regression, corresponding to the predicted probability that the candidate company is a potential customer as given by a *Web-content* model. In order not to bias our evaluation, we build the *Web-content* model using the same training set as used by the Logistic Regression model. However, in order to train the Logistic Regression on the additional *Web-content* score, we need to provide values for this variable on the training data. We could do this by training a *Web-content* model on the training set and providing the scores on the same training data. However, the Logistic Regression trained on this input could be prone to over-fitting. Hence, a better approach is to use cross-validation on the training set to get unbiased scores from a *Web-content* model, i.e., the training set is further split into 10 folds, and the instances in each fold are scored by a *Web-content* model that has been trained on the remaining 9 folds. These unbiased estimates are then used as inputs to the Logistic Regression along with all the other structured features.

Table 1 compares AUCs produced by the individual models and the composite models, averaged over 10 runs of 10-fold cross-validation. As indicated in bold, all methods incorporating web content into *Firmographic* models performed statistically significantly better than the component models, based on paired t-tests ($p < 0.05$). Clearly, there is much value in the website content – both by itself, and more so when combined with existing structured content. Overall, Vote-Avg performs best for *Rational*, and Vote-Prod is the most effective for *Websphere*. The ROC curves in Figure 2 and 3 show the relative impact of building these composite model over using only firmographics or web content.

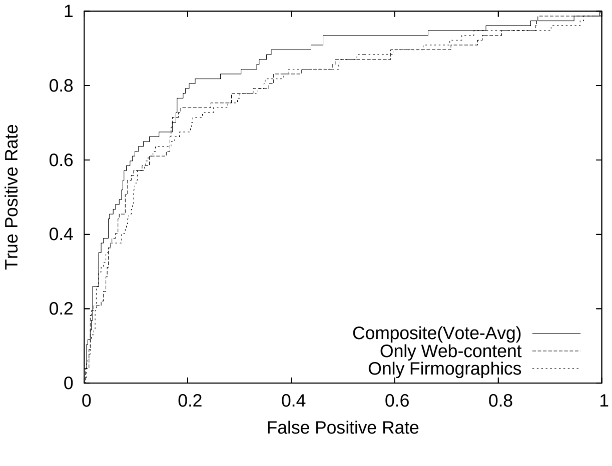

Figure 2: Comparing composite propensity models with component models for *Rational*.

In [9], we compared previous firmographic-based OnTarget models with a baseline model that simply ranks prospects

Table 1: Comparing different propensity models in terms of AUC.

Model	*Rational*	*Websphere*
Firmographic	0.793	0.776
Web-content	0.796	0.756
Vote-Avg	**0.844**	**0.809**
Vote-Prod	**0.825**	**0.810**
Nesting	**0.841**	**0.801**

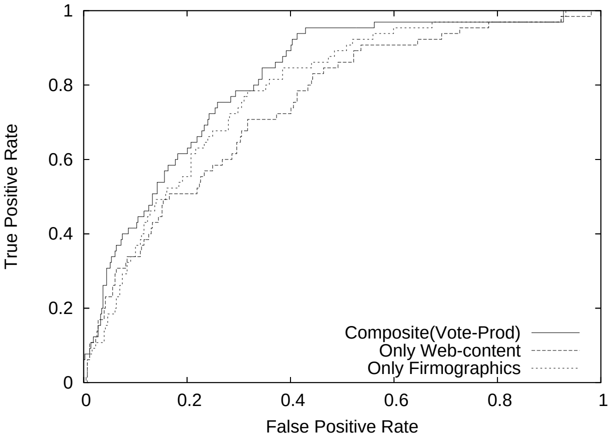

Figure 3: Comparing composite propensity models with component models for *Websphere*.

by a measure of company size. Cross-validation showed that the OnTarget models out-performed the baseline consistently. We also compared the models in a more realistic setting, namely their success in predicting actual sales based on model scores computed in a previous quarter [9]. In this case, we also found that the OnTarget model dominantly out-performed the baseline, suggesting that the performance advantage, seen in statistical cross-validation, carried over to the actual deployment of the models. Hence, there is an indication that the improved accuracy of the composite models shown in Table 1 can translate to improved performance in an actual marketing application.

5. ACTIVELY SELECTING WEB CONTENT

We have demonstrated that using web content of potential customers can significantly improve customer targeting, compared to using purely firmographic data. However, the firmographic data for over 15 million companies is readily available for modeling from the D&B database; which is not the case for their web content. The advantage of the web content data is that it comes at no *direct* monetary costs, since it is freely available on the Internet. It does, however, entail some significant processing and modeling costs. Namely, we need to:

1. automatically map company names to home page URLs — a non-trivial task, for which we had to develop a separate machine learning approach;
2. crawl the identified homepages for raw web content, which must also be kept up-to-date;
3. automatically process raw content into a form useful for modeling, as described in Section 3.

Having larger amounts of web content may lead to better *Web-content* models, but it also increases the training time for the text-classification models. For the above reasons, it is beneficial to minimize the number of websites that we need to process.

Randomly selecting a subset of websites to drive the *Web-content* models may be sub-optimal. A better approach would be to use a model, trained on the firmographic and available (partial) web data, to actively acquire the most useful new web content from which to learn. This problem corresponds to the Instance-Completion setting of the active feature-value acquisition (AFA) task [11, 19, 12]. In our specific instantiation of this problem, we have complete firmographic information for each training instance, along with the class label. However, we begin with a partial (possibly empty) set of web-content features. Our task is to select the next best instance from the set of incomplete instances (I) for which to acquire web-content, and add to the set of complete instances (C), so as to maximize the marginal improvement over the current model.

We study this AFA task of actively selecting web content in the iterative framework shown in Algorithm 1. Each iteration estimates the utility of acquiring web content features for each instance that currently has only firmographic features. The missing web content of a subset $S \in I$ of incomplete instances with the highest utility are acquired and added to T (these examples move from I to C). A new model is then induced from T, and the process is repeated.

Algorithm 1 Framework for Active Selection of Web Content

```
Given:
$C$ - set of (complete) instances with both firmographic and web content
$I$ - set of (incomplete) instances with only firmographic features
$T$ - set of training instances, $C \cup I$
$\mathcal{L}$ - learning algorithm
$m$ - size of each sample

1. Repeat until stopping criterion is met
2.     Generate a classifier, $M = \mathcal{L}(T)$
3.     $\forall x_j \in I$, compute $Score(M, x_j)$
           based on the current classifier
4.     Select a subset $S$ of $m$ instances with the
           highest utility based on the score
5.     Acquire web-content features
           for each instance in $S$
6.     Remove instances in $S$ from $I$ and add
           to $C$
7.     Update training set, $T = C \cup I$
8. Return $\mathcal{L}(T)$
```

5.1 Alternative approaches to feature acquisition

Different AFA methods correspond to different measures of utility employed to evaluate the informativeness of acquiring features for an instance. Our baseline policy, Random Sampling, selects acquisitions uniformly at random. In past work, Error Sampling has been used as an effective approach to AFA, with the objective of maximizing classification accuracy [11]. We describe this approach below, and propose two alternative approaches, which may be better suited for our modeling objective of maximizing AUC.

5.1.1 Uncertainty Sampling

The first active feature-value acquisition policy we explore is based on the uncertainty principle that originated in work on optimum experimental design [8, 4] and has been extensively applied in the active learning literature for classification, regression and class probability estimation models [3, 2, 16]. The notion of uncertainty has been proposed for the acquisition of class labels in the traditional active learning setting, but has not been previously used for feature-value acquisition. For a model trained on incomplete instances, acquiring missing feature values is effective if it enables a learner to capture additional discriminative patterns that improve the model's prediction. Acquiring feature values for an example is likely to have an impact, if the model is uncertain of its class membership. In contrast, acquiring feature values of instances for which the current model already embeds strong discriminative patterns is not likely to impact model accuracy considerably. *Uncertainty Sampling*, is based on this observation.

The *Uncertainty* utility measure captures the model's ability to distinguish between instances of different classes. For a probabilistic model, the absence of discriminative patterns in the data results in the model assigning similar likelihoods for class membership of different classes. Hence, the *Uncertainty* score is calculated as the absolute difference between the estimated class probabilities of the two most likely classes. Formally, for an instance x, let $P_y(x)$ be the estimated probability that x belongs to class y as predicted by the model. Then the *Uncertainty* score is given by $P_{y_1}(x) - P_{y_2}(x)$, where $P_{y_1}(x)$ and $P_{y_2}(x)$ are the first-highest and second-highest predicted probability estimates respectively. At each iteration of the feature acquisition algorithm, complete feature information is acquired for the m incomplete instances with the lowest scores, i.e. the highest prediction uncertainties. Note that lower scores correspond to higher utilities in Algorithm 1.

5.1.2 Error Sampling

Prediction uncertainty implies that the likelihood of correctly classifying an example is similar to that of misclassifying it. Hence *uncertainty* provides an indication of a model's performance and potential for improvement through feature acquisition. A more direct measure of the model performance and of the value of acquiring missing features for a particular instance is whether the instance has been misclassified by the current model. Additional feature values of misclassified examples may embed predictive patterns and improve the model's classification accuracy. Error Sampling is motivated by this reasoning, and as such prefers to acquire feature values for instances that the current model misclassifies. At each iteration, it randomly selects m incomplete instances that have been misclassified by the model. If there are fewer than m misclassified instances, then Error Sampling selects the remaining instances based on the *Uncertainty* score (defined earlier). Formally, the Error Sampling score for a potential acquisition is set to -1 for misclassified instances; and for correctly classified instances the *Uncertainty* score is used. At each iteration of the feature acquisition algorithm, complete feature information is acquired for the m incomplete instances with the lowest scores.

5.1.3 Labeled-margin Sampling

The Error Sampling approach described above, prefers the selection of incomplete instances that are misclassified by the current model. However, it does not distinguish between misclassified instances. An alternative would be to prefer misclassified instances that are misclassified with high confidence, working under the assumption that correcting a more confident error is more likely to improve the subsequent ranking of instances. This approach can be nicely captured by selecting incomplete instances based on their labeled margin [17]. Formally, the labeled-margin score of instance x is given by $P_{y_{correct}}(x) - P_{y_{incorrect}}(x)$; where $P_{y_{correct}}(x)$ is the predicted probability estimates of the correct class and $P_{y_{incorrect}}(x)$ is the maximal probability assigned to any incorrect label. We refer to this approach as Labeled-margin Sampling, where at each iteration of the feature acquisition algorithm, complete feature information is acquired for the m incomplete instances with the lowest labeled-margin scores.

5.2 Experimental evaluation

We ran experiments to compare Random Sampling and the three AFA strategies described above. The performance of each method was averaged over 10 runs of 10-fold cross-validation. In each fold, we generated learning curves in the following fashion. Initially, the learner has access to all incomplete instances, i.e., instances with only firmographic features. The learner builds a classifier based on this data. For the active strategies, a sample of instances is then selected from the pool of incomplete instances based on the measure of utility using the current classification model. The missing values (web-content features) for these instances are acquired, making them complete instances. A new classifier is then generated based on this updated training set, and the process is repeated. In the case of Random Sampling, the incomplete instances are selected uniformly at random from the pool. Each system is evaluated on the held-out test set after each iteration of feature acquisition. As in [11], the test data set contains only complete instances, since we want to estimate the true generalization performance of the constructed model given complete data. The resulting learning curves evaluate how well an active feature-value acquisition method orders its acquisitions as reflected by model AUC. To maximize the gains of AFA we acquire features for a single instance in each iteration, i.e., sample size $m = 1$.

For the base learner in Algorithm 1 we use the best method for each data set, i.e., Vote-Avg and Vote-Prod for *Rational* and *Websphere* respectively. The results comparing the different active feature-value acquisition approaches is presented in Figure 4 and Figure 5. The results show that, by actively selecting the most informative websites to learn from, all three AFA approaches build models with higher AUCs for the same number of websites acquired by Random

Sampling. In fact, with active selection of only 100 websites for *Rational*, we can reach close to the performance of using all web content. For *Websphere*, we can even exceed the performance of a model that uses all web content, by reaching an AUC of 0.827 with only 100 websites. Clearly, acquiring data from all websites is not only unnecessary, but also carefully selecting informative web content and ignoring potentially noisy data can lead to even better models.

In Figure 4 for *Websphere*, we see that Labeled-margin Sampling and Error Sampling clearly outperform Uncertainty Sampling. Uncertainty Sampling uses the current model's predicted class probability estimates to acquire information for instances it is uncertain about. However, it does not make use of the class labels, which are present for all (including incomplete) instances. Error Sampling uses these class labels to determine which incomplete instances are being misclassified by the current model. By preferentially selecting these misclassified instances, Error Sampling improves on Uncertainty Sampling. Labeled-margin Sampling goes a step further, by preferentially selecting misclassified instances that have been classified with high confidence. Acquiring more information, via web content, for these instances helps to correct egregious errors in ranking faster than Error Sampling. For *Rational* (Figure 5) the distinction between the different AFA approaches is less clear, as the curves cross over at multiple points. However, it is clear that any of these active sampling methods is a better approach to selecting websites to process than sampling instances uniformly at random.

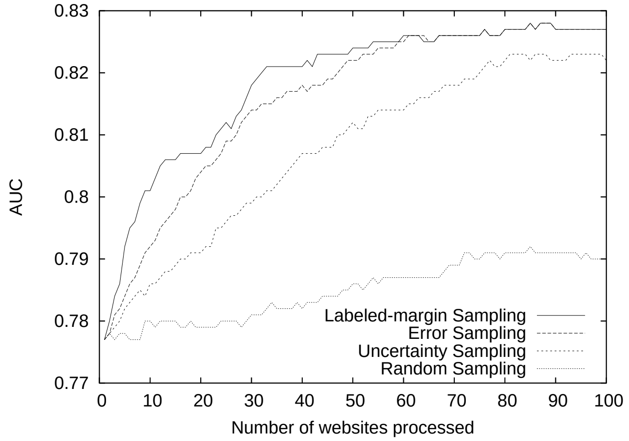

Figure 4: Different active feature-value acquisition methods compared to Random Sampling for *Websphere*.

6. CONCLUSION

In this paper, we address the problem of estimating the propensity of a company to buy a new product. In doing so, we demonstrate how the websites of potential customers provide a rich source of information for determining a company's propensity to buy a specific product or brand. In particular, we show how text classification models built on such web content can significantly boost the performance of existing targeting methods that rely solely on firmographic information. Furthermore, we present methods to minimize the cost of acquiring web content, by actively selecting only the most informative websites to process for modeling. The resulting models, using a small subset of actively-selected web content, are not only faster and cheaper to build, but can produce targeting models that are as good as (or better than) models that use content for all companies in the training data.

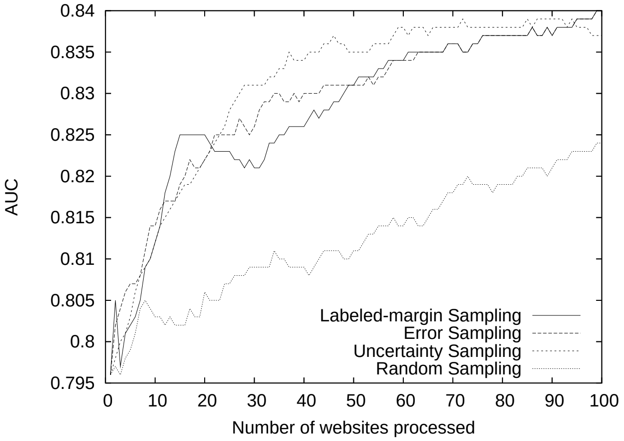

Figure 5: Different active feature-value acquisition methods compared to Random Sampling for *Rational*.

7. ACKNOWLEDGMENTS

We would like to thank Maytal Saar-Tsechansky and Foster Provost for valuable discussions and insight on active feature-value acquisition. We would also like to thank Ildar Khabibrakhmanov, Cezar Pendus, and Yan Liu for their help in data processing.

8. REFERENCES

[1] C. Buckley, G. Salton, and J. Allan. The effect of adding relevance information in a relevance feedback environment. In *Proceedings of the seventeenth annual international ACM-SIGIR conference on research and development in information retrieval*. Springer-Verlag, 1994.

[2] D. Cohn, L. Atlas, and R. Ladner. Improving generalization with active learning. *Machine Learning*, 15(2):201–221, 1994.

[3] D. A. Cohn, Z. Ghahramani, and M. I. Jordan. Active learning with statistical models. *Journal of Artificial Intelligence Research*, 4:129–145, 1996.

[4] V. Federov. *Theory of optimal experiments*. Academic Press, 1972.

[5] G. Forman. An extensive empirical study of feature selection metrics for text classification. *Journal of Machine Learning Resesearch*, 3:1289–1305, 2003.

[6] E. Frank and R. R. Bouckaert. Naive bayes for text classification with unbalanced classes. In *Proc 10th European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 503–510, 2006.

[7] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In *Proceedings of the Tenth European Conference on*

Machine Learning (ECML-98), pages 137–142, Berlin, 1998. Springer-Verlag.

[8] J. Keifer. Optimal experimental designs. *Journal of the Royal Statistical Society*, 21B:272–304, 1959.

[9] R. Lawrence, C. Perlich, S. Rosset, J. Arroyo, M. Callahan, J. M. Collins, A. Ershov, S. Feinzig, I. Khabibrakhmanov, S. Mahatma, M. Niemaszyk, and S. M. Weiss. Analytics-driven solutions for customer targeting and sales-force allocation. *IBM Systems Journal*, 2007.

[10] A. McCallum and K. Nigam. A comparison of event models for naive Bayes text classification. In *Papers from the AAAI-98 Workshop on Text Categorization*, pages 41–48, Madison, WI, July 1998.

[11] P. Melville, M. Saar-Tsechansky, F. Provost, and R. Mooney. Active feature-value acquisition for classifier induction. In *Proceedings of the Fourth IEEE International Conference on Data Mining (ICDM-04)*, pages 483–486, 2004.

[12] P. Melville, M. Saar-Tsechansky, F. Provost, and R. Mooney. An expected utility approach to active feature-value acquisition. In *Proceedings of the International Conference on Data Mining*, pages 745–748, Houston, TX, November 2005.

[13] D. Mladenić and M. Grobelnik. Feature selection for unbalanced class distribution and naive bayes. In *Proceedings of the Sixteenth International Conference on Machine Learning (ICML-99)*, pages 258–267, 1999.

[14] J. Rennie, L. Shih, J. Teevan, and D. Karger. Tackling the poor assumptions of naive bayes text classifiers. In *Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003)*, pages 616–623, 2003.

[15] S. Rosset and R. D. Lawrence. Data enhanced predictive modeling for sales targeting. In *Proceedings of the 2005 SIAM International Conference on Data Mining (SDM-05)*, 2005.

[16] M. Saar-Tsechansky and F. J. Provost. Active learning for class probability estimation and ranking. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI-2001)*, pages 911–920, 2001.

[17] R. E. Schapire, Y. Freund, P. Bartlett, and W. S. Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. *The Annals of Statistics*, 26(5):1651–1686, 1998.

[18] Y. Yang and J. O. Pedersen. A comparative study on feature selection in text categorization. In *Proceedings of ICML-97, 14th International Conference on Machine Learning*, pages 412–420. Morgan Kaufmann Publishers, San Francisco, US, 1997.

[19] Z. Zheng and B. Padmanabhan. On active learning for data acquisition. In *Proceedings of IEEE International Conference on Data Mining*, 2002.

[20] Z. Zheng, X. Wu, and R. Srihari. Feature selection for text categorization on imbalanced data. *SIGKDD Explorations Newsletter*, 6(1):80–89, 2004.