

Estimating Sales Opportunity Using Similarity-Based Methods

Sholom M. Weiss¹ and Nitin Indurkha²

¹ IBM Research, Yorktown Heights, NY 10598, USA

² School of Comp Sci & Engg, UNSW, Sydney, NSW 2052, Australia

Abstract. We describe an application of predicting *sales opportunity* using similarity-based methods. Sales representatives interface with customers based on their potential for product sales. Estimates of this potential are made under optimistic conditions and referred to as the opportunity: How much can be sold if a sale were to be made? Since this can never be verified exactly, the direct use of predictive models is difficult. In building systems for estimating sales opportunity, the key issues are: (a) predictions for targets that cannot be verified, (b) explanatory capabilities (c) capability to incorporate external knowledge (d) parallel computation of multiple targets and other efficiencies (e) capability to calibrate optimism in the predictions. (f) method stability and ease of maintenance for incorporating new examples. Empirical experiments demonstrate excellent predictive accuracy while also meeting these objectives. The methods have been embedded in a widely-used similarity-based system for IBM's worldwide sales force.

1 Introduction

Consider the scenario of sales representatives marketing products, often in person, to companies. They prepare for a sales pitch and invest much time and effort. Actual success depends on many factors including the skills of sales representatives, product quality, competition and pricing. The amount of effort can be adjusted by the representatives and their management. For expensive products and services, these efforts are often made based on the magnitude of potential sales under optimistic scenarios, as opposed to the likelihood of an actual sale. Even though a sale is not likely, if the sales opportunity is large enough, a sales representative will do a pitch on the outside chance that the customer may switch from a competitor. In many organizations where sales representatives are involved, estimates of opportunity are taking on an increasing importance as a critical measure relative to sales effort. It gives the organization a rationale for dedicating increased resources to close a sale.

Our objective is to build a system that computes opportunity for sales to a customer. This system will compute sales opportunity both for current customers and for new customers. For current customers, opportunity will suggest possibilities for expanded sales. New customers make up a much larger universe. For these customers, opportunity will suggest direction in a large space of

possibilities. Representatives cannot call every customer. The opportunity computations can suggest those customers where a potential sale will compensate for the required effort.

In building systems for estimating sales opportunity, the key issues are: (a) predictions for targets that cannot be verified, (b) explanatory capabilities (c) capability to incorporate external knowledge (d) parallel computation of multiple targets and other efficiencies (e) capability to calibrate optimism in the predictions. (f) method stability and ease of maintenance for incorporating new examples. We show how these issues can be successfully addressed by using similarity-based methods to find current customers who are similar to new customers. Our approach results in excellent predictive accuracy in our application while also meeting all the operational requirements.

2 Sales Opportunity

Our interest is in sales opportunity, which is a more optimistic target than expected sales. One view of opportunity is the expected magnitude of sales when a sale is made. This relationship is expressed in equation 1, where the distribution of sales to a company is conditional on a company's features like its size and industry.

$$\text{opportunity} = \text{expected}(\text{sales} | \text{sales} > 0) \quad (1)$$

Sales opportunity is inherently an optimistic estimate, and optimism can be adjusted by modifying equation 1. Instead of an expected value over the distribution of possible sales to a company, which is an averaged value, a percentile could be used, for example the 75-th percentile. Instead of any positive sale, a constant threshold could be specified to reflect minimum satisfactory sales goals.

The real-world incarnation of this problem is expressed as a sample of company data over a designated feature set. For current customers, the actual sales are known. In our application, information is available for most companies of moderate or large size. For these companies, we know the number of employees, their revenue, and their industry code. We can assemble labeled data about companies, matching company features to actual product sales. To estimate sales to a company, whether expected or potential, a nearest-neighbor method will find similar companies. The sales to these similar companies can be used as a proxy for a sample from the distribution of possible sales to that company. Once similar companies can be identified, estimates for opportunity can readily be computed from recorded sales to these similar companies using equation 1 or an optimistic variation substituting percentiles or thresholds.

3 Similarity-Based Methods

Similarity-based methods, nearest neighbors, can offer highly competitive results with other predictive methods. They have strong theoretical bounds on predictive performance. In practice, the similarity or distance function may need extensive tuning to achieve good results. Prediction methods are typically compared

and evaluated by measuring their accuracy on sample data. For structured data and very large numbers of examples, alternative prediction methods have often been preferred because they can be applied by invoking a compact function without online access to the original examples. With more powerful computing, similarity-based prediction methods become more enticing for embedding in web-based systems. The similarity-based methods are relatively slow – each new case must be compared to the stored cases. Some new computational approaches can substantially reduce computing time[1]. Alternatively, for some systems, the predictions may be computed and stored prior to invoking the system, thereby reducing the time observed by a system user.

In modern similarity-based systems, for example search or information retrieval systems, users have access to a huge set of examples. These are specialized operating environments, where examples are sparse within the feature space, for example, a word representation within a document, where most words do not occur in a specific document or the search string. For predictive purposes, not all data may be needed, and methods for simplification have evolved from condensed nearest neighbors [2] to conditional random sampling [3]. Still, from a systems perspective, it is important to have access to full data. While this is critical for document retrieval, it is also essential for structured data management systems containing customer information that must be accessible in full detail.

Many commercial applications of similarity methods fall into the category of collaborative filtering methods and recommender systems[4], [5]. From a methodological perspective, our application is closely related to those systems, while maintaining strong explanatory capabilities[6]. There are numerous twists to our application that distinguish it from recommender systems. Here, products are not recommended directly. Rather, an estimate of opportunity for product sales is provided that can range from zero to millions of euros. The sales representative is an intermediary, not a purchaser, who decides whether to pursue an opportunity based on many additional factors including personal knowledge of the customer or industry. Unlike most collaborative filtering applications, like those based on product ratings, the input features are nonuniform and are presented on many different scales from binary to real-valued. Because our target is opportunity, rather than an outright purchase, the exact value of the target variable, i.e. the opportunity, is unknown for all members of the customer database.

4 Data Description and Data Transformation

In our application, customers are companies. We obtained data on American and Canadian companies from Dun and Bradstreet. This includes such fields as number of employees, revenue, and an industry code. Extracting data from a proprietary internal database, we consider the sales of 5 IBM brands and the sum of sales for the five brands. These brands constitute most of IBM's software sales. Table 1 summarizes the dimensions of the data we used in all experiments.

Table 1. Data Summary

total inputs	16
numericals	14
codes	2
number of code values	3 and 26
outputs	6
total records	650,110
non-customers	612,677
customers	37,433

The feature space is relatively modest, but the number of examples is large. This can be a good environment for a nearest neighbors method because of the absence of large numbers of noisy features that can degrade predictive performance [7] [8]. One of the main challenges of designing a similarity function is to compute similarity on a normalized scale when the features are all measured in different units, for example revenue versus employee counts. One general approach is to weight the features by importance. Both linear [9] and nonlinear approaches have been described [10]. Examples of classification applications using weighted similarity functions are [11] and [12]. A generalized and automated scheme for weighting features is highly advantageous, but the best results may only be achievable by experimentation and application-specific tuning.

We employed a simple, yet highly effective transformation. All features were transformed to binary variables as follows: The codes are mapped into n binary variables, where n is the number of distinct codes. All continuous numerical variables are binned into 4 intervals as determined by k -means clustering. The net result is the transformation of the data from 16 features into 85 binary input features ($14 \times 4 + 3 + 26$). Now all input features are on the same binary scale. The quality of the similarity function remains a question, but the similarity function is completely transparent. Just rank the neighbors by counting the number of matches between the new vector and the stored vectors.

To test hypotheses, we randomly divide the customer data into 70% train and 30% test. In our results, we report on the test data. In our experiments, we stratified the data into 5 groups based on the total sales to make sure that the smaller number of very large sales were represented in both train and test samples. The customer data will have labels in the form of actual sales. The non-customers have zero sales across all products, and estimates will be inferred by comparing to the smaller customer population.

5 System Evaluation

5.1 Evaluating Predictive Performance

While our major interest is examining factors other than predictive accuracy, we must still consider whether the similarity-based method is competitive with alternatives. We know that our primary target for estimation, the sales opportunity, is not directly known. It is highly related to expected or actual sales which

are known for current customers. For a given company's set of similar companies, the computation of expected sales versus sales opportunity differs only in the thresholds or designated percentiles for equation 1. We will use these targets in a sample of companies and their actual sales. Our immediate goal is to see whether a nearest-neighbors method will produce predictions that are close in accuracy to results produced by prominent alternative methods. This goal should not require that nearest neighbors produce the best results for opportunity estimates. But, it does require that the nearest-neighbor approach be competitive to other methods when using the same features.

We compared results for several regression methods using the previously described train and test data sets. Figure 1 summarizes the results on the test examples. The results for prediction were measured by *rmse*, a measure of difference between actual and predicted sales. In the graph, the RMSE values are plotted relative to the baseline RMSE of ten nearest neighbors, 10-NN. For k-NN methods, the average of the k nearest neighbors is taken as the prediction.

The methods that we compared to nearest neighbors, k-nn, were linear least squares, a regression tree, and bagged trees. We also include the results of simply using the mean value of product sales computed from training cases. The linear method was run with all features. The trees were checked at various sizes and k-nn was tested with various values of *k*. Results for k-nn were slightly better with feature selection (FS), when a few features were deleted by greedy backward elimination on the training data.

Clearly, the k-nn method is competitive and effective. The linear method, which is highly popular in industrial circles for predicting sales, was relatively weak. Bagging and k-nn were competitive. Nearest neighbor, and other similarity-based methods, have a rich history of effectiveness in many applications. These results support its use when an appropriate similarity measure can be developed for the application.

5.2 Describing and Measuring Optimism

One view of opportunity is the expected sale magnitude given a sale occurs. This was expressed in Equation 1. A company is trying to gauge potential for sales prior to committing additional resources. The hope is that these additional resources, whether additional preparation or a stronger sales team may improve the yield from past experiences, or they may entice a new customer with a better deal.

The degrees of overall optimism can be specified in several ways. In Equation 1, the threshold can be increased to some non-zero threshold, *m*, as in Equation 2.

$$opportunity = Average(sale | sale > m) \quad (2)$$

Increasing the threshold will evaluate customers as if a bigger sale will be closed. It treats the sales below the threshold as zero. Generally, this increases overall opportunity estimates unless raised to the point of non-occurrence of sales.

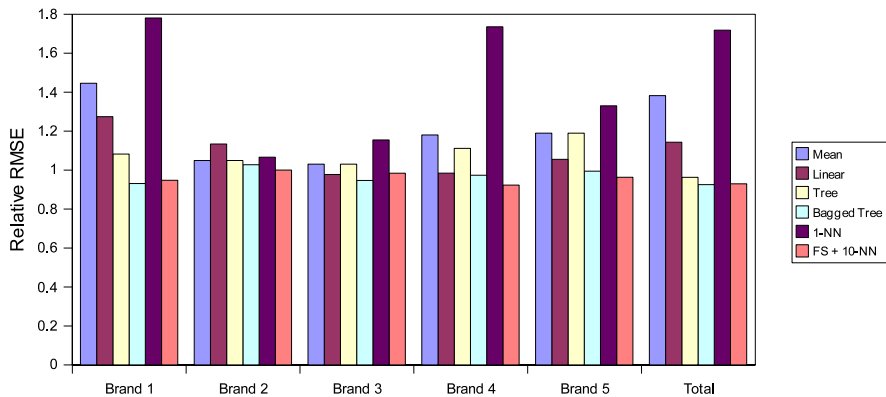


Fig. 1. Comparison of Methods for Sales Prediction

An alternative and complementary approach to varying optimism is to replace the "average" with another measure. The median, i.e. the 50th percentile value, is an obvious choice, but the concept can be generalized to any percentile. We have computed values at the 75th and 50th percentile for this application. The average has the highest optimism among these measures. That's because a few extremely large customers, like banks and insurance companies, have much larger purchases than most customers. These very large customers are typically at the tail of the sales distribution. Figure 2 shows the relative ratios to expected sales (i.e. no minimum threshold) for the average-value as a measure of optimism of this application. These are shown for exceeding several threshold values, m , of Equation 2 for current customers only. The average is contrasted with the 75th percentile and the 50th percentile in Figure 3 where the optimism at two thresholds is shown. As expected, the average has the highest value of the three. The maximum value could also be used as a measure of opportunity. It does not change with threshold, so the optimism measure would be unchanged over the thresholds. In our application, the optimism as measured by the maximum value was 37.39 – substantially higher than any of the other measures.

The degree of optimism is best decided by knowledge and goals set by the sales executives. Alternatively, the system designers can examine the effects of various measures on global optimism, and then make decisions accordingly. We want optimism, but we do not want to set the bar so high that it is unrealistic. How can we measure the global effect of optimism? Equation 3 describes a simple ratio for yearly sales of a product. The overall optimism is the ratio of total opportunity to total current sales.

$$\text{optimism} = \text{opportunity} / \text{currentSales} \quad (3)$$

All these computations are trivial for a similarity-based method. The k-most similar companies are identified and grouped. This can be a relatively small group, in our case, as small as 10 or 20 similar companies. Once grouped, almost no additional effort is needed to estimate opportunity by any of the measures

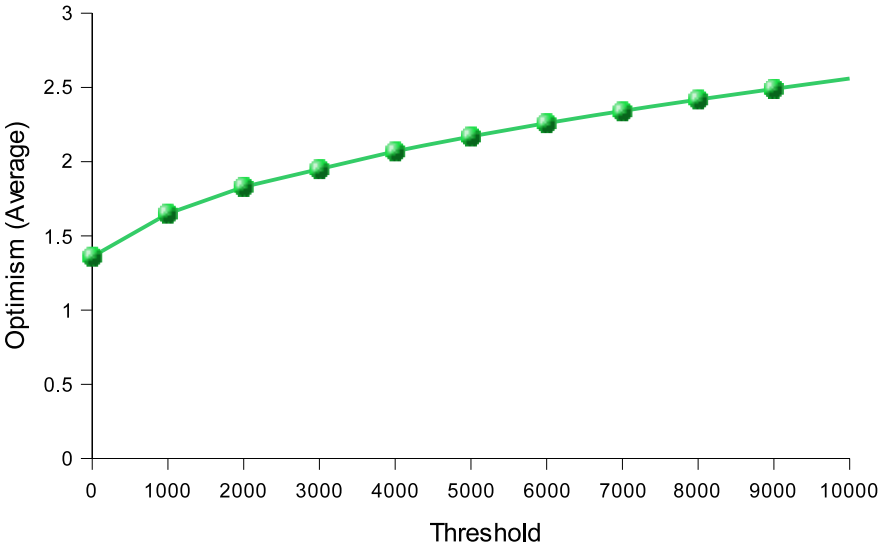


Fig. 2. Average as a Measure of Optimism

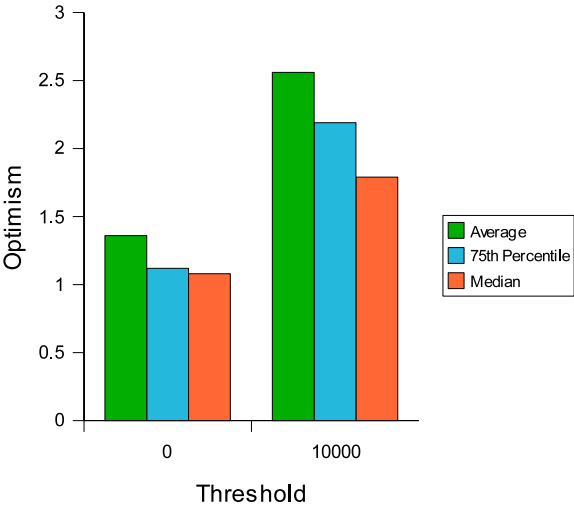


Fig. 3. Comparison of Optimism Measures

of optimism. Because our notion of opportunity is based on similar companies, we add a floor to the estimate as in Equation 4. Opportunity for selling to a company can be no less than current sales to that company.

$$finalOpportunity = Max(initialOpportunity, currentSales) \tag{4}$$

5.3 Explanatory Capabilities

The ultimate consumers of our opportunity estimates are sales representatives. They will make decisions on sales efforts by examining these numbers in the context of current sales results. An analogy can be made with a search engine and its rankings. There, the rankings are a means to presenting answers to a user. Multiple answers are listed, all of them similar to the query. The objective is to rank based on similarity criteria, but many answers are presented to the user. The user will examine results and make decisions about the usefulness of the retrieved documents. The quality of the answer depends both on the similarity rankings and the actual answers.

We could look narrowly at our sales opportunity task and conclude that the estimates are the sole conveyor of meaning in this application. Yet that would ignore an important component of the estimates. Just like the answers from the search engine, the actual answers, the most similar companies, also have meaning. They explain how the estimate was derived, but just as important, can aid the sales representative in moving forward. For example, when a sale is concluded in a particular industry, it is natural to look for similar companies to make a similar sales pitch. Or even if a sale was not closed and was won by a competitor, it could be expedient to examine similar possibilities.

Because company records are structured with precise values, additional unstructured information may be missing. Once the estimates are derived and similar companies are found, all supporting materials could be examined to formulate a sales plan. The list of similar companies could be considered a form of triage, where these companies are examined carefully to look for additional clues, possibly not recorded in the structured record, that could lead to success in the sales effort. The key concept is that the task need not be just the estimation of sales opportunity. Rather, the component records of companies used to estimate sales opportunity can be most valuable in supporting both the estimate and the planning for future actions.

Using similarity measures, our objectives can be restated as illustrated in Figure 4. Estimating sales opportunity is still a principal goal. To this, we add the presentation of the constituent companies used to formulate the estimate. These can be ordered by similarity as would be done for a search engine. Depending on user queries, full information about these companies and their prior purchases could be made available.

-
- estimate sales opportunity
 - order similar companies
 - present k most similar companies with complete sales histories

Fig. 4. Objectives for Similarity Measures

5.4 Compatibility with Knowledge

The general approach is to build an automated system. The system will process empirical data about companies and estimate their sales opportunities. After assembling a feature set based on availability and knowledge, we might expect to make this a purely empirical application of similarity-based methods. Yet, we have strong incentives to incorporate knowledge into the system. Two considerations drive our preference for additional knowledge:

1. Sales opportunity is unknown, making our data unlabeled.
2. Presenting a list of similar companies to a user implies consistency with intuitive knowledge of the field.

Enhancing the system with knowledge can constrain the answers to be less variable and more compatible with human expectations, possibly with a reduced accuracy when tested on data sampled from a specific time period. Still, mating knowledge to an empirically-driven system can be satisfying for this application because the consumers are humans who must take action. They will not blindly accept numbers that are incompatible with their experience and might defeat their sales approach.

What kind of knowledge is of interest, and how well does it fit with a similarity-based method? With hundreds of thousands of examples, a case-based reasoning[13] approach to specifying the examples is ill-suited. Our main vehicle for incorporating knowledge is the similarity function. A company may be represented as a vector of data, but the components of that vector also have meaning. When we say that two companies are similar, it must also meet our intuitive notion of similarity. For example, we may not feel that two companies are similar when they engage in completely different industries. Adding conditions that enhance the intuitive notion of similarity will meet the objective of effective explanations to the sales representatives. They will review the matched companies, and various constraints on the matches will increase satisfaction or at least remove some answers that might seem inadmissible. Two simple notions are very sensible, yet readily fit within a function for computing similarity of two companies. If two companies are similar, then

1. they are in the same industry
2. they are approximately the same size.

The industry code for a company is readily obtainable from Dun and Bradstreet or other sources at different resolutions. We used a single letter code with 25 possibilities. For size, both the number of employees and company revenue can be obtained. A window on size can be placed on the compatibility of two companies. We use a heuristic that a company Y must have sales (or employees) no more than twice that of company X for it to be considered a match.

Figure 5 illustrates the difference in predictive performance for predicting actual sales with these heuristics. We show the performance of four simple heuristics and compare with the baseline where no match heuristics are used. For sales

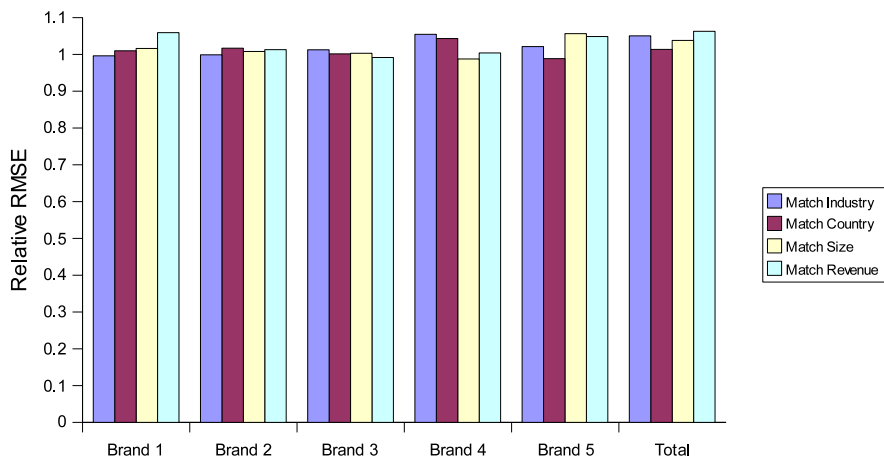


Fig. 5. Impact of Match Heuristics

and size matching, we clustered values into four bins (determined by training data) and declared a match if values fell in the same bin. The performance drops slightly on the test data, mostly from the required industry match. However, the matches themselves would be more intuitive and more acceptable to the sales representatives who must take the information and call on customers, while often citing purchases by their competitors.

The records of companies are not extracted from a single data source. Rather, the information is found in different sources and merged. In our application, we consider the sales of five product lines. For each company the proprietary sales numbers for these five products are merged with other information about these companies, such as sales revenue and number of employees. Although companies can be stored with unique identifiers, for example, "duns numbers," errors are unavoidable, both in the record matching process for assembling data or in the data sources. Knowledge of the relationships among variables leads to other heuristic rules for adjusting the numbers used for estimation. For example, given a company has revenue X , we can estimate that no more than 10% of X will be spent on information technology purchases. Although there may be exceptions to this rule, generally it is effective because a higher estimate will not serve as a good template for similar expectations.

5.5 Ease of Maintenance

Because a similarity-based method does not directly model labels and does not build any intermediate structure (like a tree), system maintenance is greatly simplified. No need to learn a distinct model for each target. This reduces the problem to computing similarity from the initial set of input features. In our application, the set of features will remain stable. It's very unlikely or at least infrequent that new features will be found that are useful and can be measured

for tens of thousands of examples. New companies may be added to the database and the measurements for the stored companies, both input features and output product sales, will regularly be updated. In our application, these numbers are updated every quarter. The similarity-based method is remarkably tranquil in this environment. Compare the expectations for a decision tree versus similarity methods. The tree would likely be re-learned for every target and a new tree induced for each. With similarity methods, one possible course of action is to do nothing, since the estimates are computed dynamically based on a query. If the estimates are needed in a batch for all companies, then very little is done other than recomputing the k most similar companies for each company and the estimates from these neighbors. The binarized distance function that we have specified leaves little to vary. Possibly the size of k could be rechecked as well as the number of discrete intervals for a real-valued variable. In our application, these would be checked every quarter. How stable is a nearest neighbors method? Table 2 contrasts the k -nearest neighbors with the decision tree for the target variable of total sales. All results are relative to those for 10-nn. For each of the n companies in the database, we measure the standard error within the partition for the tree and form the weighted average over all companies as in Equation 5 where $StdDev_i$ for selected tree = standard deviation over all members of partition i , and $StdDev_i$ for k -nn = standard deviation across all members of the k neighbors of i .

$$stability = \sum_{i=1}^n StdDev_i \tag{5}$$

Table 2. Stability of tree and k -NN

Method	Relative Average Mean	Relative Stability
Tree	1.5512	9.0718
10-NN	1.0	1.0
20-NN	0.8794	0.7855
50-NN	0.7142	0.5531

Here we see that the nearest neighbors estimates are far more stable than those for the tree. While the accuracy of results are important, the stability that the similarity-based method is also desirable, since it would need less maintenance and less frequent retraining.

5.6 Computational Efficiency

Similarity methods can be quite slow. Given a training set, the simplest variation compares each new example to the full training set, an $O(n)$ process, where n is the size of the training set. To get estimates for all stored companies, the process is $O(n^2)$. Fortunately, when embedded within a system for computing sales opportunity, various factors ameliorate this disadvantage. The fastest algorithms for computing nearest neighbors offer a large improvement over direct

comparison. While the high computational cost for finding similar companies is a criticism of nearest neighbors, several critical characteristics of our system, discussed in the previous sections, lessen its importance. Some of these factors may even make nearest neighbors computationally advantageous when compared to other methods.

The results of similarity computations are reusable. For any given company, we need to know its estimated sales opportunity, which is computed from similar companies. However, we need not recompute these number dynamically every time a company is referenced. The possible list of queries is limited to single companies, so we know every possible query, and the estimates can be pre-computed and stored. When a user queries a company, the estimates can be directly retrieved. Thus, the more intensive computation is separated from its repeated use, and values can be computed in batch mode with no ramifications for user delays.

The central comparison is "Compare company X to current customers." We want to estimate opportunity for both current customers and potential new customers. The set of potential new customers, sometimes referred to as whitespace, is generally much larger than current customers. In our case, 20 times larger. Given j customers and k new customers, all estimates are made by comparing to the j customers. If sequential comparisons are made, the number of comparisons is $j * (j + k)$. The number of comparisons is substantially less than suggested by the full number of companies in the sample.

A major advantage of the similarity methods over many alternative learning methods is its inherent parallelism in computing estimates for multiple targets. Most learning methods rely on training to a specific target, the label; a separate solution is induced for each target. For example, a separate tree would be induced for each label. In our system, 5 products would require 5 different trees and possibly another for total sales. The estimates computed by similarity can be described by following simple two-step algorithm:

1. Find k most similar companies to company X.
2. From the k companies of step 1, compute all estimates.

For example, if expected sales were to be computed, then the estimates are the average sales for each product to the k companies. To compute the opportunity, an estimate for opportunity in Equation 1 is the average of the positive values for each product. Because of the little work in computing the estimates from the product values of the k companies, the number of products is almost inconsequential to the computations. One thousand products will take just a little more time than one product. In all our results, the same value of k was used over all companies and products.

In our application, we have a mixture of categorical and real-valued features. A similarity or distance function can benefit from re-scaling the variables. Earlier, we described our data transformations, where the original features are all mapped into binary variables. This is straightforward for the categorical variables, and it requires some slight additional computations for the real-valued

variables. The transformed data had 85 binary features from which similarity is computed.

This type of binary representation has potential for very efficient computation. Consider the simple similarity function that we actually used:

$$\text{similarity} = \text{Count}(\text{matchingOnes}(X, Y)) \quad (6)$$

We measure similarity as the count of matching ones between the company X and a candidate neighbor, company Y. Because the features are binary-valued, the counts can be computed like words in text for information retrieval (IR). There, for each feature, i.e. a word, token or phrase, an inverted list of records with a positive feature occurrence is computed. When a new search pattern of words is entered, only the inverse lists for these words are accessed to compute similarity. In our application, the search space will not be as sparse, but efficiencies can still be great. When computing the similar companies for company X, only the positive features in the vector for X will be considered. For each of the positive features, the companies appearing in the inverted list will have their matching count incremented by one. This simple algorithm is typical for IR and can be readily adapted to our similarity computations as describe below, where $\text{inverse}(i,j)$ is the j -th company on the inverse list of feature i .

1. Initialize counts all training cases to zero.
2. For each positive feature i in company X
 For each j such that $\text{inverse}(i,j)$ exists, increment $\text{count}(j)$.
3. Select top k counts.

One additional computational efficiency is worth noting. A suitable value for k must be determined. This is determined both empirically and based on knowledge of the domain. Our data support a small value of k , on the order on 10 or 20. All our results are for 10. With such a small value of k , the 10 largest values can be found by dynamically keeping a simple linear table of the 10 current best companies and adjusting when a new insertion is made.

6 Discussion

We have described an important application, where opportunities for sales to companies are estimated. The end-user of these estimates is most likely a sales representative who will call on customers. These sales representatives need an estimate that captures the potential for sales under optimistic conditions. The optimism could be a variation of the concept of "how much product could be sold if a sale is made?"

In a situation with human involvement, such as this one with sales representatives, merchants may move away from "expected" value to a "potential" perspective. This allows for optimism and separates the past from the future. A more concerted effort may result in improved sales. However, this effort will only be expended when justified by potential sales.

A similarity-based method is a good fit to this application. All necessary company data is kept online, so no additional resources are devoted to this representation. The method itself has a very strong intuitive appeal for promoting sales initiatives. The process of estimation is not arcane. Home appraisals have long been made by comparing a new home on the market to similar homes that have recently sold. Here, we compare potential product sales to a customer with those sales made to similar customers.

The target of the estimate is sales opportunity which is an unknown quantity. We offered a formulation of the problem that circumvents the need for labeled data and bootstraps from the available prior sales records. We offered several ways to quantify and adjust the optimism in these estimates. Because the training data are unlabeled (technically, the opportunity is not known), but still related to the known and labeled historical sales data, there is room for alternatives to our approach to estimating and predicting opportunity. One related approach for regression is semi-supervised learning using nearest neighbors[14], where both labeled and unlabeled samples are collected. Here our data are labeled, but the actual target, opportunity, is related, but not identical to the label.

From an applications perspective, this project is a success. The system, as described in this paper, has been implemented and deployed. The estimates are embedded in a system that is routinely accessed by hundreds of sales representatives. Starting with just the USA sales, the system has gradually been expanded to cover two dozen countries including most of Europe and the larger countries in the Asia-Pacific region. These estimates are updated worldwide every quarter by geographical region. Besides providing estimates for sales opportunity, the system can function on a query basis, giving details of similar companies and their transactions. Studies of the users' clicks relative to completed sales show strong usage on the part of the sales force, and executives estimate that over \$100M of sales are traceable to the system.

From an evaluation perspective, we showed that nearest neighbors was a highly competitive approach. This evaluation could follow the typical scenario of considering machine learning methods as interchangeable building blocks and basing decisions on predictive performance. In our study, that goal remains important, and it is no surprise that k-nearest neighbors do well in this application, even when implemented in a binary fashion.

Our goals for evaluation are more general. The prediction method is embedded in a system that will function in a structured environment, yet can respond to online queries. This leads us to consider an evaluation along dimensions other than empirical and predictive performance. Such factors as compatibility with knowledge, explanatory capabilities, stability, computational parallelism, and quantification of sales optimism can all play important roles in a real-world system's architecture. We have shown that a similarity-based system can perform well along these dimensions and have supplied empirical evidence and practical arguments to support these claims. We have provided the case for similarity-based systems but an alternative approach may also do well along these very

same dimensions[15]. In an age of building systems with embedded learners and predictors, one can expect that factors beyond pure empirical performance will play a larger role in the design of applied systems.

References

1. Beygelzimer, A., Kakade, S., Langford, J.: Cover trees for nearest neighbor. In: ICML 2006: Proceedings of the 23rd International Conference on Machine Learning, pp. 97–104. ACM Press, New York (2006)
2. Hart, P.: The condensed nearest neighbor rule. *IEEE Trans. Info. Theory* IT-14, 515–516 (1968)
3. Li, P., Church, K., Hastie, T.: Conditional random sampling: A sketch-based sampling technique for sparse data. In: Schölkopf, B., Platt, J., Hoffman, T. (eds.) *Advances in Neural Information Processing Systems 19*. MIT Press, Cambridge (2007)
4. Herlocker, J.L., Konstan, J.A., Terveen, L.G., Riedl, J.T.: Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.* 22(1), 5–53 (2004)
5. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering* 17(6), 734–749 (2005)
6. Tintarev, N., Masthoff, J.: A survey of explanations in recommender systems. In: *Workshop on Recommender Systems and Intelligent User Interfaces associated with ICDE 2007* (2007)
7. Fukunaga, K., Hummels, D.M.: Bias of nearest neighbor estimates. *IEEE Trans. Pattern Anal. Mach. Intell* 9(1), 103–112 (1987)
8. Beyer, K., Goldstein, J., Ramakrishnan, R., Shaft, U.: When is “nearest neighbor” meaningful? In: Beeri, C., Bruneman, P. (eds.) *ICDT 1999*. LNCS, vol. 1540, pp. 217–235. Springer, Heidelberg (1998)
9. Aggarwal, C.C.: Towards systematic design of distance functions for data mining applications. In: *KDD 2003: Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 9–18. ACM Press, New York (2003)
10. Wu, G., Chang, E.Y., Panda, N.: Formulating distance functions via the kernel trick. In: *KDD 2005: Proceeding of the 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 703–709. ACM Press, New York (2005)
11. Yao, Z., Ruzzo, W.L.: A regression-based K nearest neighbor algorithm for gene function prediction from heterogeneous data. *BMC Bioinformatics* 7 (2006)
12. Yang, S., Zhang, C.: Regression nearest neighbor in face recognition. In: *Proceedings of the 18th International Conference on Pattern Recognition*, pp. 515–518 (2006)
13. Kolodner, J.: *Case-based Reasoning*. Morgan Kaufmann, San Francisco (1993)
14. Zhou, Z., Li, M.: Semi-supervised regression with co-training. In: *IJCAI 2005: Proceedings of the 19th International Joint Conference on Artificial Intelligence*, pp. 908–913 (2005)
15. Merugu, S., Rosset, S., Perlich, C.: A new multi-view regression approach with an application to customer wallet estimation. In: *KDD 2006: Proceeding of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 656–661. ACM Press, New York (2006)