Product Targeting from Rare Events: Five Years of One-to-One Marketing at CPI

Brendan Kitts[1], Martin Vrieze[2], David Freed[3]

[1]iProspect
311 Arsenal Street, Watertown, MA
bkitts@excite.com

[2]Central Purchasing Inc
3505 232nd Street East
Hampton, MN
mvrieze@harborfreight.com

[3]Exa Corporation
3 Burlington Woods Drive
Burlington, MA 01803
dmfreed@yahoo.com

Keywords: recommendation system, one to one, personalization, customer specific marketing, recommender, nearest neighbor, cross sell, cross-sell

Abstract

We describe five years of work developing, analyzing and running one-to-one marketing systems in a successful cross-channel retailer. Central Purchasing, Inc. (CPI) is a large 200 store retail chain selling quality hand-tools and home improvement goods throughout the United States. In 1999, CPI wanted to utilize personalization technologies to improve the relevance of their email communications. They ran a pilot study in which they replaced their static product offers with alternatives selected by a one-to-one recommendation algorithm. The pilot test was very successful. Clickthrough increased by +40%, revenue by +38%, and units sold by +61%. The full system was deployed in October of 2000. At this time a different recommendation strategy based on the hypergeometric distribution was tested, and doubled the response rate. Since the deployment in 2000, the retailer has continued to use the recommendation system each week, maintaining control groups and reporting on performance. The Hypergeometric recommendation method is noteworthy because it is a well-defined significance test, performs well on small numbers of events, and scales well on large amounts of data. Infrequent customer purchases are common in a great number of merchandising domains, including books, music, electronics, home improvement and video games. A reliable solution to this problem is therefore of great interest to marketers.

INTRODUCTION

The dream of one-to-one marketing is to send engaging, relevant, individualized communications to each customer. Customers would look forward to the communications, since the information would be relevant and would help them in their day-to-day tasks. Customers would learn about products that can improve their quality of life, and receive attractive incentives, offers and discounts that make the product or service even more valuable.

Marketers would gain by sending communications to a more selective and higher quality audience, thereby reducing the costs associated with broadcasting to millions of customers with the same message in search of a tiny number of potential clients. Response rates would rise, allowing marketers to invest more per customer into the offer.

Unfortunately this dream is difficult to realize. It is hard to predict individual needs. Prediction methods must contend with sparse and incomplete data. Even if these problems could be overcome, the delivery technology for millions of unique one-to-one messages may be cost prohibitive.

This paper describes how we implemented one-to-one marketing at a large, multi-channel retail chain. To be successful, we needed to develop new methods for customer analysis that were designed for conditions of rare and noisy purchase events. We applied these techniques to identify products that would interest each unique customer. We then delivered these product offers to customers via email and print catalogs.

The email program has been running for over five years. The results have been excellent. Control groups are maintained every week, and these show that response rates and revenue generated from personalized communications are consistently twice as high for personalized communications as controls. Over a five year period, Central Purchasing, Inc.'s one-to-one program has generated significant increases in profit, customer satisfaction, and has connected consumers with relevant information. We believe that CPI's one-to-one program stands as a model for similar initiatives, and particularly retailers in domains where customers buy very few products.

THE 1TO1 PROBLEM

The 1to1 problem is the problem of identifying which special offer, coupon, or information should be delivered to a particular customer so as to maximize the customer's response or revenue. This differs from traditional marketing problems, in that it is assumed that every customer may receive a completely unique offer. This simple requirement puts special strains upon both the delivery technology (catalog, newspaper, T.V., web, mobile device) and upon the targeting algorithm.

In general, this problem can be formulated as follows: Select an item i, for delivery to a customer y, such that a customer response variable R (such as response rate or revenue) is maximized. R is a function of the presented product, the customer's history of purchases x, and a number of unknown factors θ related to the customer's interests, attitudes and day-to-day circumstances.

$$\mathrm{r}(y) = i : \max R(y,i,x,\theta) \qquad (0)$$

Given such a problem, a classic approach might be to develop a logistic regression model that uses promotion and past responses to solicitations to predict the probability of a response to a solicitation. More commonly, the problem is formulated as one of predicting the probability of the customer purchasing product i in the future, based on their history. This is not strictly an estimation of R, yet it is a common approach since promotion histories are often not available.

There are several technical challenges with classic prediction algorithms when applied to the one-to-one problem.

The scale of data in real-world applications is formidable. For example, the EachMovie dataset has been used by several authors to test recommendation systems (Breese et. al., 1998; Ansari, et. al., 2000). However, this data set is 2,500 times smaller than the retail data described in this article, and 40,000 times smaller than Amazon.com's data size (Linden, et. al., 2003) (Figure 1).

At this scale, space and time complexity issues become a critical factor. We've provided an analysis of a small number of representative predictive algorithms in Figure 2. This shows that simple methods such as nearest neighbor have a considerable advantage in space and time complexity.

Approaches exist to try to improve speed and memory usage. These include matrix reduction and approximation algorithms (Drineas, et. al., 2004a and 2004b; Frieze, 2004), indexing (King-Ip and Yang, 2001), support thresholding (Agrawal, 1996) and incremental training (Polikar, et. al., 2001). However, these methods introduce their own set of problems. For example, fast matrix reduction techniques can be applied equally to each of the algorithms, and also introduce error. Incremental learning schemes are susceptible to "catastrophic forgetting".

Another way to achieve scaling is to move away from modeling individual items. The large customer-item matrix can be re-cast as a smaller matrix with rich product and customer attributes - an approach sometimes referred to as "conjoint analysis." Ansari et. al. (2000) used this approach and reported good results. However, developing and maintaining accurate customer and product attributes is a challenge.

METRIC	EACHMOVIE	CPI	AMAZON
Customers	75,000	3,500,000	29,000,000
Items	1,628	100,000	2,000,000

Figure 1: Data volume from three different data sets

ALGORITHM	TIME COMPLEXITY		SPACE COMPLEXITY	
	TRAINING	EXECUTION	TRAINING	EXECUTION
Regression	f	f	f	$\frac{f}{C}$
Naïve Bayes	f	f	f	$\frac{f}{C}$
User Nearest Neighbor	$\frac{f}{CN^2}$	$\frac{Cf}{N}$	$\frac{f}{CN^2}$	$\frac{f}{CN}$
Cross-sell	f	$\frac{f}{N^2}$	$\frac{f}{C}$	$\frac{f}{CN}$

Figure 2: Worst case time and space complexity for four types of algorithms when applied to the one-to-one problem, assuming a full matrix of customers and items is used for each algorithm. Only Customer and Item terms are shown for simplicity, but this is reasonable since most other terms such as K for nearest neighbor are constant. C = Customers, N = Items, f=CN^2. We assume that $C > N > K$ so as to simplify the formulae. Only main memory space costs are shown, where-as mass storage space such as tape data which can be read in a "pass efficient" manner (Drineas, et. al., 2004) is not counted.

A second problem that one-to-one systems must deal with is reliability and accuracy. In regression, a single outlier can dramatically skew a regression model, and practitioners may need to experiment with alternatives to squared error, or transforming the variables. 1to1 magnifies this problem, since large numbers of models may need to be created autonomously without human intervention.

These problems related to 1to1 have led to a flowering of models based on the concept of K-Nearest Neighbor (Billsus and Pazzani, 1998; Ungar and Foster, 1998; Chien and Edward, 1999; Mobasher et. al., 2000). These have been variously referred to as “collaborative filtering”, “instance based learning”, “memory-based”, “case-based”, or “recommendation systems”.

Nearest Neighbor approaches are well suited to 1to1 in many respects. They are reliable because they use actual customer instances to base recommendations. They scale because expensive matrix inversions or Monte Carlo sampling methods are not needed. Finally, they are heterogeneous, treating every customer uniquely.

INTRODUCTION TO NEAREST NEIGHBOR RECOMMENDATION SYSTEMS

Intuition

The general idea of Nearest Neighbor Recommendation Systems is to find customers that have a similar product preference distribution to a user of interest, to allow a system to estimate the purchase probabilities for a set of new products for which the user has yet to purchase (Goldberg, et. al., 1992; Wyner, 1998; Herlocker, et. al., 1999; Greening, 2000; Deshpande and Karypis, 2005).

This is accomplished in two stages. In the first step, customers in the population who are similar to the candidate are found. One could think of them as the customer's "soul mates" (Nightline, 1999; Riedl, 2000). In the second step, the system predicts a customer's interest in a new product based on the advisors, and selects a product to offer (De Bruyn et. al., 2003). There are a large number of variations of this technique, and we will try to capture these in the next section.

User based recommendation

Let X be a matrix of customer profiles, such that $x_{c,i}$ refers to the units customer c has purchased of item i. Let y be a candidate customer for whom we want to find a product to recommend. The product i to recommend to customer y is selected as follows:

$$r(y) = \{i\}: \max_{I} \; D(x_{y,i}, x_{A,i}) \tag{1}$$

$$\text{where } A=\{c\}: \max_{K} \; S(x_{y,1..N}, x_{c,1..N}) \tag{2}$$

$A=\{c\}$ are the set of K customers who are most similar to y as measured by a similarity function S. They are sometimes referred to as "advisors". $\{i\}$ is a set of I items that are recommended. N are the number of items in the retailer's inventory. C are the number of customers in the population. The difference function D compares the items purchased by the advisors to the candidate, allowing the item with the greatest under-purchase to be selected. Sometimes this is described as predicting the purchase probability, but in addition to doing this, the D function also selects a final set of products. These selections may not necessarily have the highest predicted purchase likelihood, as we shall explain later. Thus, the product recommended is the product *for which sales are most favored by the advisors*, within a cluster of advisors *who are most similar* to the candidate.

In practice we often introduce two further constraints:

$$\{i\} \cup \{j\} \neq \varnothing \text{ where } x_{y,i} > 0 \text{ and } x_{c,j} > 0 \tag{3}$$

$$\neg\,[\{j\} \subseteq \{i\}] \text{ where } x_{y,i} > 0 \text{ and } x_{c,j} > 0 \tag{4}$$

(3) requires advisors to share at least one product in common with the candidate. (4) requires advisors to have purchased at least one product the candidate customer hasn't bought.

Typical functions for measuring customer-advisor similarity S include mean squared difference (Shardanand and Maes, 1995), entropy, class probabilities (Pennock, et. al., 2000), Pearson Correlation (Resnick, et. al., 1994) and Vector Cosine. The last two, Vector cosine and Pearson Correlation are very widely used, and we note that Vector Cosine is the same as the Pearson Correlation except that the vectors are not mean-centered. The formula for Pearson correlation is below as an example

$$S(x_{y,1..N}, x_{c,1..N}) = \text{"Pearson's R"} = \frac{\sum_{i=1}^{N}(x_{y,i} - m_y)\cdot(x_{c,i} - m_c)}{\sqrt{\sum_{i=1}^{N}(x_{y,i} - m_y)^2 \cdot \sum_{i=1}^{N}(x_{c,i} - m_c)^2}} \quad \text{where } m_c = \frac{1}{N}\sum_{i=1}^{N} x_{c,i} \tag{5}$$

The most simplistic difference scoring function predicts item purchase probability as the average of the advisor values. After creating this average, the item most purchased by the advisors and not yet purchased by the candidate, is the product that is recommended to the candidate.

$$D(x_{y,i}, x_{A,i}) = \begin{cases} \frac{1}{K}\sum_{c \in A} x_{c,i}, & x_{y,i} = 0 \\ \infty, & otherwise \end{cases} \tag{6}$$

Some researchers change the average into a weighted average, where the advisors who are more similar to the candidate exert a stronger influence on the average (Li, 2005).

$$D(x_{y,i}, x_{A,i}) = \begin{cases} \frac{\sum_{c \in A} S(x_{c,1..N}, x_{y,1..N}) \cdot x_{c,i}}{\sum_{c \in A} S(x_{c,1..N}, x_{y,1..N})}, & x_{y,i} = 0 \\ 0, & otherwise \end{cases} \tag{7}$$

However, simply recommending the most probable/frequent/highest-volume products in the advisor group tends to result in many products that are very common throughout the population. For instance, in terms of music, everyone may have bought a Madonna CD at one point. These items are not specific to the advisor group, but represent high purchase levels in the population. Early practitioners discovered that the quality of recommendations could be improved dramatically by dividing each item score by the expected value of the item score in the population (Kitts, et. al., 2000; Deshpande and Karypis, 2005). This is where *D* diverges from being a simple prediction of the most probable product that would be purchased.

$$D(x_{y,i}, x_{A,i}) = \begin{cases} \frac{\left(\frac{\sum_{c \in A} S(x_{c,1..N}, x_{y,1..N}) \cdot x_{c,i}}{\sum_{c \in A} S(x_{c,1..N}, x_{y,1..N})}\right)}{\frac{1}{C}\sum_{c \in [1..C]} x_{c,i}}, & x_{y,i} = 0 \\ \infty, & otherwise \end{cases} \tag{8}$$

This same idea has been idea is called "lift" in the association rules literature (PMML, 2005), and "inverse document frequency" normalization in the tf-idf method from Information Retrieval literature to improve the relevants of documents retrieved from a search (Spärck Jones, 1972; Robinson, 2004; Salton, 1989; Chan, 1999). The idea of dividing rate of purchase by the expected rate in the population turns the item score into something like a significance test, in which the higher the score, the less likely it is that the interest in the item could be different from the population level of interest by chance. Later in this paper we suggest changing all of the heuristic scoring functions into fully fledged significance tests that select advisors and make recommendations based upon this probability.

Item based recommendation

We have described what is commonly referred to as "user-user recommendation". The other major approach in the literature is "item-item recommendation". The strategy now looks for items that are complimentary with a product *i*, in that they have a high probability of being purchased with *i*. Compliments are determined by applying a similarity function $S(i,j)$ to every product pair (i,j),. The K items with the highest similarities to product *i* then populate a set of compliments, $A=\{j\}$. After identifying the set of complements for *i*, the purchase levels in those complements $x_{y,A}$ are used to predict the purchase level in product *i*, $x_{y,i}$. After predicting purchase level, the item with the highest lift interest (or a similar score) can be recommended in much the same way as before.

$$r(y) = \{i\}: \max_{I} D(x_{y,i}, x_{y,A}) \tag{9}$$

$$\text{where } A=\{j\}: \max_{K} S(x_{1..C,\,i}, x_{1..C,\,j}) \tag{10}$$

An example D function is below:

$$D(x_{y,i}, x_{y,A}) = \begin{cases} \dfrac{\frac{1}{K}\sum_{j \in A} x_{y,j}}{\frac{1}{C}\sum_{c \in [1..C]} x_{c,i}}, x_{y,i} = 0 \\ 0, otherwise \end{cases} \tag{11}$$

It is possible to implement item-item algorithms that work even more simply than the above. After identifying the complements, it is possible to simply consider each item purchased by the candidate, and recommend complimentary products.

Another useful "side effect" of the item recommendation approach is that the complimentary products discovered each product *i*, can be saved and used to populate "related products" pages (eg. "people who purchased at this item, also purchased these….."). This can assist users find a product of interest to them when navigating product pages.

Item recommendation methods are gaining favor with a growing number of researchers (Sarwar, et. al., 2001; Papagelis and Dimitris, 2005; Deshpande and Karypis, 2005). The primary reason is scaling. In typical retail applications there are more customers than items. For instance, the retailer described in this paper has a customer base of 3 million and an inventory of 100,000 products. Thus, the item-item similarity matrix would be 900 times smaller at $1\text{x}10^{10}$ versus an equivalent user-user matrix of size $9\text{x}10^{12}$. The smaller size allows for increased speed and reduced memory consumption.

HISTORY OF RECOMMENDATION SYSTEMS

The technology of recommendations has a long and colorful history. In the following section we will attempt to sketch some of the major developments[1].

The US Census Bureau was one of the first groups to develop recommendation-system like algorithms. The US Census Bureau was faced with very large data sources - millions of records of consumer data - with high numbers of missing fields. To fill in those missing values, the Census Bureau developed a technique that they called "Hot Deck Imputation" (Ford, 1980; O'Brien, 1983; Oh and Scheuren, 1980; David, Little, et. al., 1986; O'Brien, 1983; US Census SIPP Manual, 2005). The method essentially predicted customer attributes based on customers with similar profiles. US Census had developed the first large-scale Nearest Neighbor customer attribute predictor.

The concept of using Nearest Neighbor for marketing purposes was not really developed (to our knowledge) until Hey (1989) published a patent on the idea. Hey's patent computed *S* and *D* as

$$S(x_{y,1..n}, x_{c,1..n}) = \left[\frac{1}{n} \sum_{i:x_{y,i} \neq 0} CVT(|x_{y,i} - x_{c,i}|) \right] \log_2 n \qquad (12)$$

$$D(x_{\mathrm{y},i}, x_{\mathrm{c},i}) = \sum_{c} x_{c,i} 2^{-RANK(S(x_{y,1..n}, x_{c,1..n}))} \qquad (13)$$

where *n* were the number of non-zero attributes, CVT is roughly the negation of the difference scores. The technology appeared to work. LikeMinds Corporation purchased the rights to Hey's patent lock, stock and barrel and began marketing it in their Personalization Server. Macromedia later purchased LikeMinds and continues to use the method today (Greening, 2000).

With the emergence of the World Wide Web in 1992, the potential of one-to-one could be truly realized. The web is an ideal delivery technology for one-to-one. Unlike print newspapers, the web can display content that is completely customized for each and every user. As a result, web-based one-to-one applications quickly appeared.

One of the earliest was Group Lens. Group Lens attempted to filter newsgroup content and show the most interesting content to the user based on their interests (Resnick, et. al., 1994; Konstan, et. al. 1997). Group Lens used a nearest-neighbor algorithm based on Pearson correlations to match up users, and then recommend content that the other users liked (Herlocker, Konstan, Borchers and Riedl, 1999).

[1] The authors apologize in advance for any recommendation systems that have been missed or dates that are incorrect. We have made a more detailed version of this history publicly available on the web at http://www.recommendationsystems.com, and we welcome any corrections or additions.

Reidl who worked on Group Lens found that the recommendation methods could be easily applied to other domains such as movie recommendation. This system was entitled “Movie Lens.” Riedl eventually went on to found Net Perceptions, which later fell victim to the dot com bubble and ceased operations in 2002.

In the mid-1990s the MIT Media Lab also experimented with web-based recommendation systems. Ringo (Shardanand and Maes, 1995) and Video Recommender (Hill, Stead, et. al., 1995) were systems that generated recommendations on music and movies, respectively. Ringo expanded upon the Group Lens algorithm by tweaking the Pearson correlation by only considering neighbors with correlations over some threshold. Ringo later turned into FireFly, which also was the name of the company founded by its developers. FireFly was later acquired by Microsoft in 1998

Sometime around 1996, Amazon began including recommendations on their book Web pages. These were displayed as “customers who bought this also bought….” Barnes and Noble followed suit in 1997-98 with their own recommendation system (Johnson, 1998). In 2003 Amazon revealed their recommendation method to the general public for the first time, publishing it in *IEEE Internet Computing* (Linden, Smith and York, 2003). It turns out that Amazon uses an item-item recommendation method. In this method, the *S* function is computed as a cosine of the item vectors (equation 5).

In 2000 Amazon was granted a patent on its BookMatcher functionality (Jacobi and Benson, 2000). Pinpoint, Inc. attempted to sue Amazon over its collaborative filtering technology (Jannarkar, 2003). However another patent holder, John Robinson (Robinson and Ellsworth, 1998 and 1999), claimed in the mid 1980s that he developed a prototype system called 212-ROMANCE, which allowed users to access personal dating recommendations via a touch-tone telephone (Robinson, 2003). If Robinson did indeed have a working commercial product in the mid 1980s, this would obviate the later patents.

Recommendation systems continue to be used in an increasing range of applications. In 2004, TiVo published their recommendation method (Ali and van Stam, 2004). TiVo also uses an item-item system much like Amazon. Their problem is to predict the number of thumbs that a TV show will receive by a customer, $x_{c,i}$ for a TV show that this customer has not yet seen. Their *S* function is a modified Pearson Correlation, and their *D* function is a weighted average of the Pearson correlation scores:

THE HYPERGEOMETRIC METHOD FOR RARE EVENTS

In applying recommendation system technology to our domain, we needed a method that would meet the challenges posed by one-to-one applications, which was to (a) be heterogeneous, (b) scale in both time and space, and (c) provide high quality recommendations. Our method addresses these requirements by (i) allowing for dynamic clusters of customers just like in nearest neighbor, (ii) efficiently dealing with 0 data, and (iii) using a well-defined significance test to govern customer selection and product recommendation.

Heuristic scoring methods for *S* and *D*, like those that we have described in the previous section, have a number of disadvantages. The primary problem is in correctly accounting for degrees of freedom and noise. For instance, a vector dot product between two profiles can generate a high score just because one of the vectors has larger values. A correlation of 1.0 can be generated from a match on 1 product or 100 products - but we would intuitively expect the latter case to be more significant than the first case. Our use of significance tests allows us to determine whether the similarity between two product purchase distributions could have been caused by chance - the null hypothesis. Consequently, we can cope with noise elegantly, and arrive at better estimates of the similarity of advisors.

Method

Recall that we have two customer profiles, $x_{y,1..N}$ and $x_{c,1..N}$, where y is the "candidate" customer for whom we want to find products to recommend, and c is the "advisor" customer. $x_{c,i}$ is the quantity of item i purchased by customer c.

Let us consider a single candidate y and an advisor c. Let all products this advisor purchased be labeled "red" with the number equal to R, and let all other products in the retailer inventory be regarded as "blue" with cardinality B. When we inspect the candidate's product purchases, we can now say that we see some assortment of red and blue products.

Let us define D as the number of items in the candidate's profile, M as the number of red items in the candidate's profile (i.e. the number of products in their profile that were also bought by the advisor), and N as the total number of items in the retailer's inventory (N=R+B).

We want to calculate the probability that the candidate drew their assortment of red and blue products from an urn containing R red and B blue, by chance. In other words, we want to know the probability that candidate's assortment was produced at random.

The lower the probability of the null hypothesis, the more likely it is that the similarity between the candidate and advisor is not by chance - specifically, both customers have the same product preference distribution.

The probability of drawing M red balls in D attempts from an urn containing R red and B blue is given by the hypergeometric distribution (Wallenius, 1967). Using this formula, we are able to calculate a well-defined p-value for the probability that the candidate drew their number of red balls at random. Because we are interested in a significance test, for the probability of drawing this number or more of red balls, we use the cumulative hypergeometric probability distribution.

$$S(x_{y,1..N}, x_{c,1..N}) = \sum_{m=M}^{D} \frac{\binom{R}{m}\binom{N-R}{D-m}}{\binom{N}{D}} \tag{14}$$

$$\text{where } R = \sum_{i=1}^{N} (x_{c,i} > 0)\,,\ D = \sum_{i=1}^{N} (x_{y,i} > 0)\,,\ M = \sum_{i=1}^{N} (x_{y,i} > 0 \wedge x_{c,i} > 0) \tag{15}$$

$$\binom{p}{q} = \frac{p!}{q!(p-q)!} \tag{16}$$

The procedure finds the K customers for which it is unlikely that their similarity in purchases could have been caused at random. We will use this to presume that advisors have the same preference distribution to the candidate.

After finding the advisors that are uncannily similar to the candidate, it is now necessary to select an item. The idea in selecting an item i, is that we want to identify products that are truly characteristic of the advisors. These items should be unusually high in terms of observed probability in the cluster, to expected probability in population. Once again, a statistical significance test will allow us to discern truly unusually preferred products. A chi-square statistic χ^2 is a classic statistical method for determining whether events counted in two groups could have been caused at random. The lift statistic introduced in (8) is monotonically proportional to χ^2 around the mean and is arguably more interpretable that it's statistical cousin. For instance, a lift of "5" means that the product was purchased at a rate 5 times higher than the population, where-as a χ^2 of 25 does not have a simple interpretation.

$$D(x_{y,i},x_{A,i}) = \chi^2 = \left(\frac{E(x_{A,i}) - E(x_{1..C,i})}{E(x_{1..C,i})}\right)^2 \propto \frac{E(x_{A,i}) - E(x_{1..C,i})}{E(x_{1..C,i})} \text{ if } E(x_{A,i}) > E(x_{1..C,i}) = \frac{E(x_{A,i})}{E(x_{1..C,i})} - 1 \propto \frac{E(x_{A,i})}{E(x_{1..C,i})} \quad (17)$$

Profile Representation for Rare Events

Storing explicit customer purchases for very large inventories becomes wasteful. Each customer could require over 2 million zeros. Since purchases are rare, profiles can be generalized to be a record of "events." The customer profile can merely contain a list of events. These in turn can be stored in variable length records, each of which is a list of events that occurred for that customer or user. Events that did not occur are not stored. The result is a considerable compression of data, allowing for improved recommendation speed and scaling. Each customer profile for customer i is defined as:

$$Profile_i = (c_i, L_i, E_{i,1..Li}, \lambda_i, B_{i,1\ldots\lambda i}) \quad (18)$$

where c_i is the identifier for the customer, L_i is the list size of customer i, $E_{i,j}$ is the event-id for event j of customer i, λ_i are the number of blocked items, and $B_{i,j}$ is the jth item that is blocked for customer i. Blocked items are those that the customer has viewed and should not be recommended again. Figure 4 shows the basic representation of customer data. This data representation improves the scaling from O(CN) to O(Cn) where n are the average number of events stored per customer. Bit-wise operators can be used to count matches, creating a further speedup (Freed, et. al., 2000).

customerindex	listsize	list-of-events	blocksize	list-of-blocked-items
c_1	L_1	E_{11}, ... , E_{1L1}	λ_1	B_{11}, ... , $B_{1\lambda 1}$
c_2	L_2	E_{21}, ... , E_{2L2}	λ_2	B_{21}, ... , $B_{2\lambda 2}$
...	...	..., ... , ...	...	..., ... , ...
c_n	L_n	E_{n1}, E_{n2}, ... , E_{nLn}	λ_n	B_{n1}, ... , $B_{n\lambda n}$

Figure 4: Profile data structure for rare events

Extensions to the Method

The basic hypergeometric method ignores the fact that different products appear in the population with different probabilities. Thus, when computing the similarity between one customer and another, if advisor A matched the candidate on five common products, and advisor B matched the candidate on five products - but this time, rare products - then advisor B should have a lower probability of accomplishing that match at random. This can be modeled by *The Multivariate Non-Central Hypergeometric Probability Distribution* (Chesson, 1976; Wallenius, 1963). This distribution models an urn that contains balls of different colors *and different sizes*. An efficient numerical approximation to the non-central hypergeometric distribution remains an open area of research (Fog, 2004).

1TO1 AT CENTRAL PURCHASING

Introduction to CPI

Central Purchasing, Inc. (CPI) is a large multiple sales channel distributor of quality hardware, tools and equipment for the general public, handymen, resellers and contractors. CPI has over 3.5 million direct mail customers, 7 million retail customers and 203 retail stores.

CPI started its weekly emails in 1995. At that time these emails were in plain-text ASCII format, and included a static set of products that the merchandising department would select each week for its customers.

With the advent of the Hyper Text Markup Language (HTML), product descriptions and images could be retrieved dynamically from CPI's server and displayed in each email. Thus, instead of sending the same products to everyone, every customer could be delivered an email featuring only products of interest to them. This would be as simple as changing the <img src> link to point to a different product. The missing part of this equation was the system that would assemble the product selection for each customer.

Pilot Experiment

In 1999 CPI contacted two marketing companies specializing in predictive analytics and set up a competition to determine which of the two firms produced better results. 60 million rows of customer/SKU-level (Stock Keeping Unit) purchase history were provided to each company as well as a list of customers who were to receive recommendations. The problem was to recommend 15 products to each customer, and then monitor the response rates, average order sizes and sales per thousand pieces mailed to both test groups and a third control group who would receive a normal weekly email message.

Each company used quite different approaches in solving this problem. One used a collaborative-filtering approach, the details of which were not disclosed. The second company used an item-item based recommendation approach later called "Cross-sell" (Kitts, et. al., 2000). This technique involved recommending products that had a high "affinity" to products in the customer's profile, and yet were not already in the customer's purchase history.

Little tuning could be performed for this test. Personalized emails were delivered to customers in June of 1999. Most of the response results from the broadcast were collected after only 3 days.

Pilot Experiment Results

The results of this bake off are provided in Figure 5. CPI's analysis of the test was slightly different to our own, as reported in Kitts et. al. (2000). The differences are not systematically biased one way or another. Here we report CPI's own assessment of this test.

Company B achieved higher revenue. Company A achieved a higher response rate. Both companies significantly increased conversion rate, revenue and orders ($p<0.01$, t-test). Response rate increased by 37% and 29%, respectively. Revenue increased by 32% and 46%.

Thus, given two independent companies, employing different one-to-one strategies, and without tuning, a significant increase in revenue could be achieved by personalizing the emails.

These results were compelling. CPI proceeded to integrate one of the recommendation engines into its operational marketing system.

Pilot test results

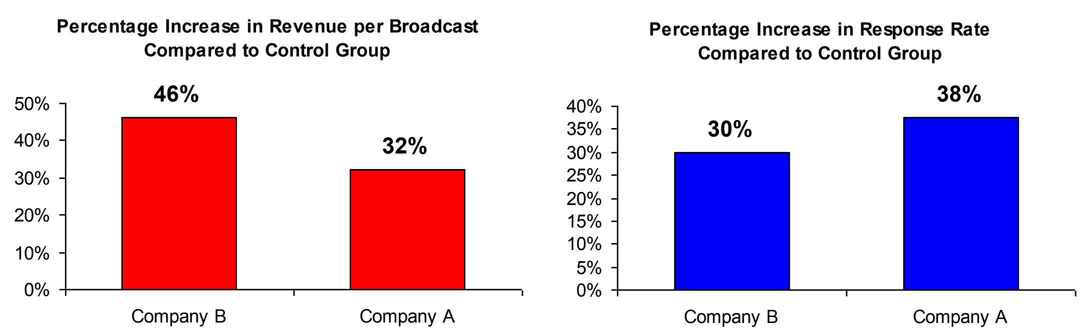


Figure 5: The main result of the pilot experiment was that Company A generated better response rate, where-as Company B generated better revenue per broadcast. Both companies significantly increased each variable. The major conclusion was that two independent companies each generated a significant increase, thus, improvements were not only possible, but were likely.

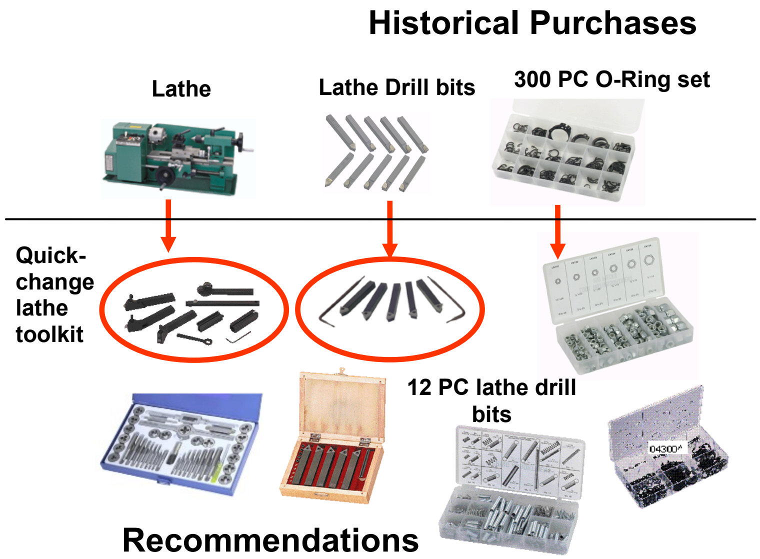


Figure 12: On June 18, 1999 the above customer bought a $329.99 Mini Lathe, along with some replacement cutting bits, a toolkit of O-rings and Retaining rings. In response the system has recommended an additional set of carbide lathe cutting bits, a Lathe quick-change toolkit, and toolkits with locknuts and springs. The customer purchased the lathe toolkits within three days.

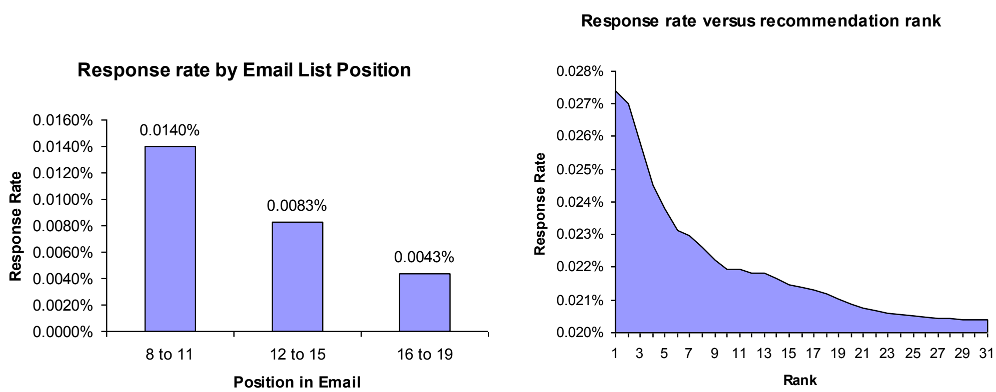


Figure 11 (left) SKU-level Response rate by position in email, from live customers who were promoted. Items in the top positions have higher response rates. **(right):** Effect of blocked items on SKU response rates on live email response rates. As more items become blocked, the recommendation algorithm has to reach further down into its ranked items to provide items that have lower and lower suitability for the customer. With more blocked items, the response rate decreases.

Production System, 2000-Present

In 2000 the full system was deployed at CPI. The major changes in the production deployment were (a) the use of a new recommendation method based on the hypergeometric significance test, and (b) maintenance utilities for "model retraining," "product blocking," and other tasks that allowed for continuous operation. Retailer controlled operation was achieved in October of 2000.

Parameter Optimization

A common problem in machine learning systems is the number of free parameters, and how they can be set optimally. These free parameters couldn't be completely tested experimentally, since there was a financial cost to showing poor-performing algorithms to customers. As a result, we needed a way of tuning the recommendation algorithm without subjecting the company to poor performance.

Recommendation Algorithm Scoring

In order to achieve large-scale parameter optimization, we developed a "recommendation algorithm simulator." This system divided the transaction data into two sets - one before a certain date called the training set, and another after that date called the test set. We then built customer profiles - and predicted recommendations - based on the training set only. If the recommendation algorithm could predict what the customer later bought, then we determined that the recommendation algorithm was effective in correctly predicting the customer's interests. We defined a "simulated response rate" R^S for each recommendation algorithm as

$$R^S = \frac{\sum_y \delta(i,y) : i \in r(y)}{\sum_y \# r(y)} \tag{19}$$

where $\delta(i,y)$ is 1 if customer y bought product i in the test set and 0 otherwise, and $\#r(y)$ are the number of items recommended for customer y.

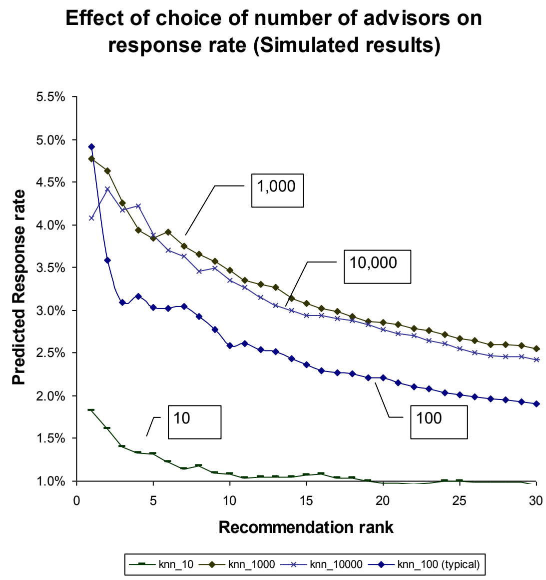
Using the simulator we tested 40 versions of the recommendation algorithm settings around our initial set of parameters. This led to a major discovery. In our initial deployment we had selected K=100 because we thought the smaller number of advisors would lead to better specificity. However, Simulator analysis indicated that K=100 was actually an extremely poor setting (Figure 6). As a result we introduced the K=1,000 variant to the production system, which according to the simulator had the highest response rate. After changing this one parameter, we approximately doubled our live response rate (Figure 13).

Problems with Offline Scoring

The Simulator Analysis was useful; however, we noticed some curious results. In testing the parameter variants, we included three hand-coded algorithms for comparison to our recommendation algorithm: (a) Recommend Same – recommends the same products that the customer had purchased in the past, (b) Recommend Frequent – recommends the most frequent products (the top sellers or most probable), and (c) Recommend Random – recommends a random product from the 1,000 available.

Recommend Same generated the highest R^S and Recommend Frequent was the best performing algorithm behind Recommend Same. If these results were to be believed, it would suggest that mass merchandising could generate higher response rates than one-to-one, which we knew to be incorrect.

Many authors including Breese et. al. (1998), Popescul, et. al. (2001) and Deshpande et. al. (2005) have used similar protocols for measuring recommendation algorithm accuracy. Our analysis suggests a problem that may not have been previously noted in the literature. Since the most frequent item has the highest probability of appearing in the data set, the "simulated response rate" will tend to score anything that produces frequent items very highly. A human being would likely react differently. The simulated response rate is actually a lower bound on the true response rate (assuming advertising exposure increases the propensity to respond), which may explain why we still obtained good results using the method.



ALGORITHM VARIANT	SIMULATED RESPONSE RATE
Recommend Same	5.44%
Recommend Frequent	3.17%
Recommend Random	0.10%

ALGORITHM VARIANT	SIMULATED RESPONSE RATE
K=10	1.00%
K=100	2.01%
K=1000	2.66%
K=2000	2.64%
K=5000	2.56%
K=10000	2.56%

Figure 6 (top): Simulated response rates for some of the *K* algorithm variants tested versus rank of recommendation. **(bottom)** Simulated response rates for different algorithm variants at rank 50.

RESULTS

CPI does not buy or sell email lists, only emails its own customers who have signed up for emails over the web, and conforms to US legislation on anti-spam efforts (Burns and Wyden, 2003). All emails were constructed using software developed internally by CPI's marketing department. This involved reading customer recommendations in customer-sorted order, retrieving dynamic content including graphics, item description, URL, price, and so forth, then embedding those elements in individualized emails.

Both control and personalized emails used the best promotion pricing available for that week. Control emails were identical to personalized emails except for the products featured in the email. On some of the weeks, personalized emails were not created due to staff being away, and only the (static, but otherwise identical) control emails were broadcast. Because of the consistency between control and personalized emails and pricing, this was an unusually precise test of the impact of one-to-one targeting.

Algorithm performance

Three variants of the Cross-sell algorithm, and two of the Hypergeometric algorithm, were tested against live customers, as shown in Figure 13. Initially the Hypergeometric method was deployed with K=100, however, this produced disappointing results. We subsequently introduced a new variant of Hypergeometric with K=1000, and the results dramatically improved.

All algorithms were executed at the same time - the recommendation procedure ran each of the algorithm variants in sequence, with each variant producing its own set of 15 product recommendations for a customer along with a label to indicate which algorithm was responsible. The final set of 15 recommendations was then selected at random from these outputs. This allowed us to measure response rate by algorithm per SKU exposure from each algorithm (sometimes different algorithms produced the same recommendation, but in those cases one was selected randomly).

Figures 11 and 13 use "Response per SKU exposure"; the probability of a customer purchasing the same SKU within 7 days of being exposed to the SKU in an email. Figures 7-10 show "Response per Broadcast"; the number of orders within 7 days for any of the same 15 SKUs featured in the email. CPI used "Response per broadcast" in its own internal reporting.

Overall Results

Figure 7 shows the performance of the Email program during five years of operation from 1999 to 2003 as measured by CPI. In 1999 a pilot study was run, and we report those results for comparison. In October 2000, the full system was deployed and email broadcasts were begun in earnest.

In early 2001 we tuned the recommendation algorithm with our simulator. The response rate increase now rose to 100%. The response rate has remained about 100% higher than the control response rate since this time. The three years following 2001 were 105%, 118% and 96%. Figure 8 shows the results in tabular form. In all cases the experimental groups were significantly higher than the control groups with ($p<0.01$, t-test).

Weekly Results

Figure 9 (top) shows broadcast-by-broadcast results for 1 year and 8 months of weekly email broadcasts. Breaks in the time-series were when a mass-email was run without a corresponding personalized email. The Personalized Response rate is about twice the rate of the Control group, and this difference is maintained each week.

Figure 9 (bottom) shows revenue per week. In order to publicly disclose these results, we have shown this in units that are normalized. Both experimental and control revenue per email are shown as "lift scores" over the average control email revenue. For instance, if 1to1 emails on 26th June 2002 were generating a revenue lift of 3, and a control email generated $1 per customer, then the 1to1 email would be generating $3 per customer. Figure 10 shows improvement in different email formats.

Effectiveness Compared to Other Channels

One of the most interesting findings is the cost effectiveness of the email channel. US Bancorp Piper Jaffray Equity Research (Rashtchy and Avilio, 2003) estimated that email Cost Per Sale was approximately 20 times cheaper than catalog. The reality is even more startling. Central Purchasing, Inc.'s Cost Per Sale across different channels are shown in figure 14. The Cost Per Sale for generic email is 833 times lower than for catalog solicitation, and 1,666 times lower for personalized emails.

However, before pouring the entire marketing department's budget into email, it is worth understanding some other characteristics. Email solicitations are opt-in, and too many broadcasts result in requests to unsubscribe. As a result, CPI felt it important to keep to the weekly broadcast schedule.

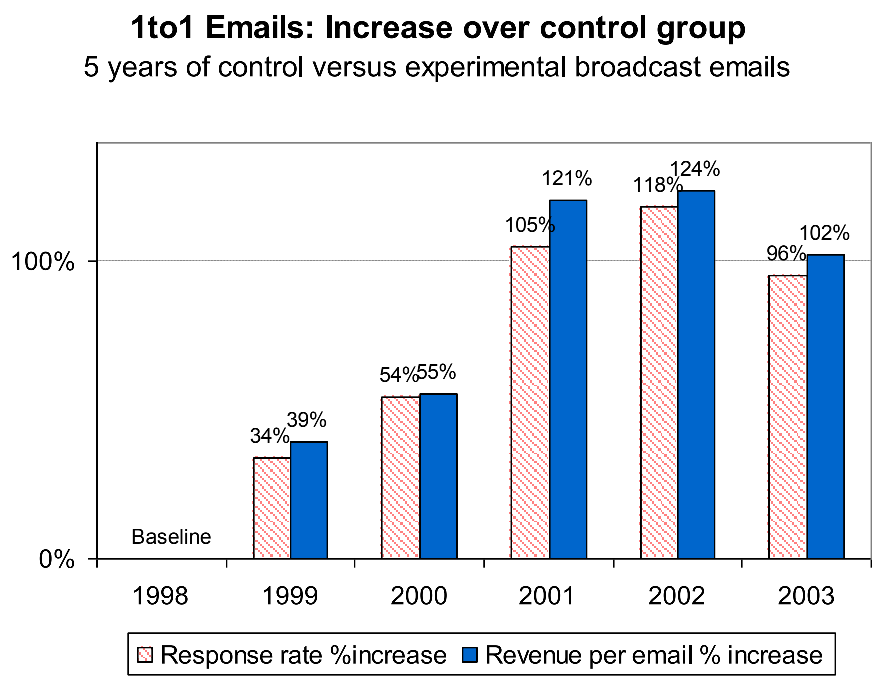


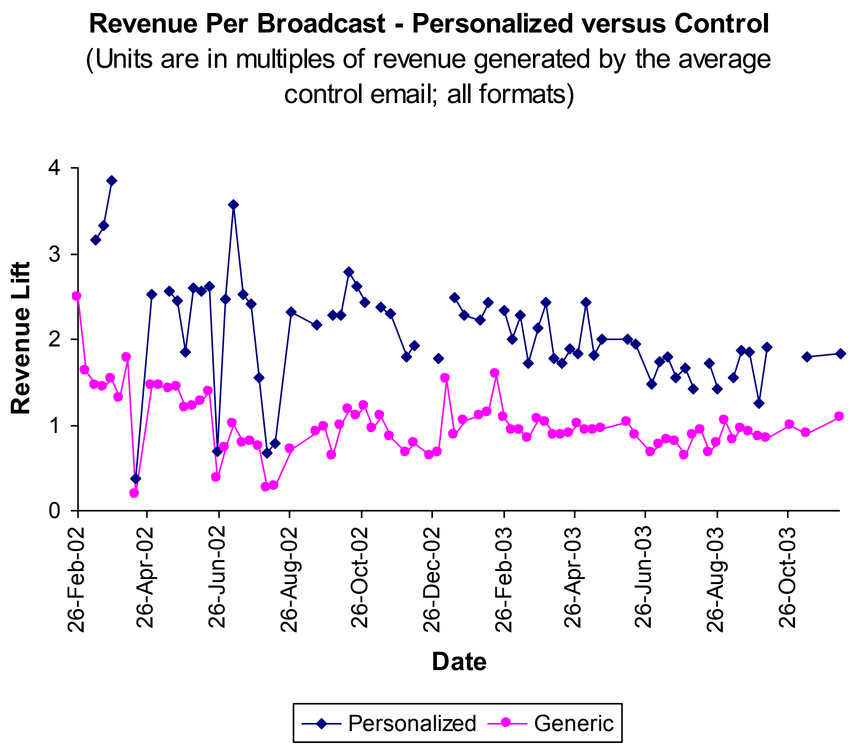
Figure 7: Personalized email performance

Yearly Results

DESCRIPTION	YEAR	BROADCASTS	MAIL QTY	ORDERS	PCT RESP	REVENUE LIFT
Generic Email-Control	1999	1	7,182	128	1.78%	$1.00

Company B	1999	1	8,301	192	2.31%	$1.46
Company A	1999	1	8,233	202	2.45%	$1.32
Personalized EMail-Test	1999	1	16,534	394	2.38%	$1.39
Total			23,716	522	2.20%	
Generic EMail-Control	2000	6	981,315	6,281	0.64%	$1.00
Personalized EMail-Test	2000	6	122,852	1,214	0.99%	$1.55
Total			1,104,167	7,495	0.68%	
Generic EMail-Control	2001	32	2,287,353	11,751	0.51%	$1.00
Personalized EMail-Test	2001	32	1,772,846	18,674	1.05%	$2.21
Total			4,060,199	30,425	0.75%	
Generic EMail-Control	2002	32	1,386,250	3,227	0.23%	$1.00
Personalized EMail-Test	2002	31	4,963,131	25,241	0.51%	$2.24
Total			6,349,381	28,468	0.45%	
Generic EMail-Control	2003	34	1,761,490	3,935	0.22%	$1.00
Personalized EMail-Test	2003	34	6,093,952	26,616	0.44%	$2.02
Total			7,855,442	30,551	0.39%	

Figure 8: Aggregated results each year from personalized email program versus control emails



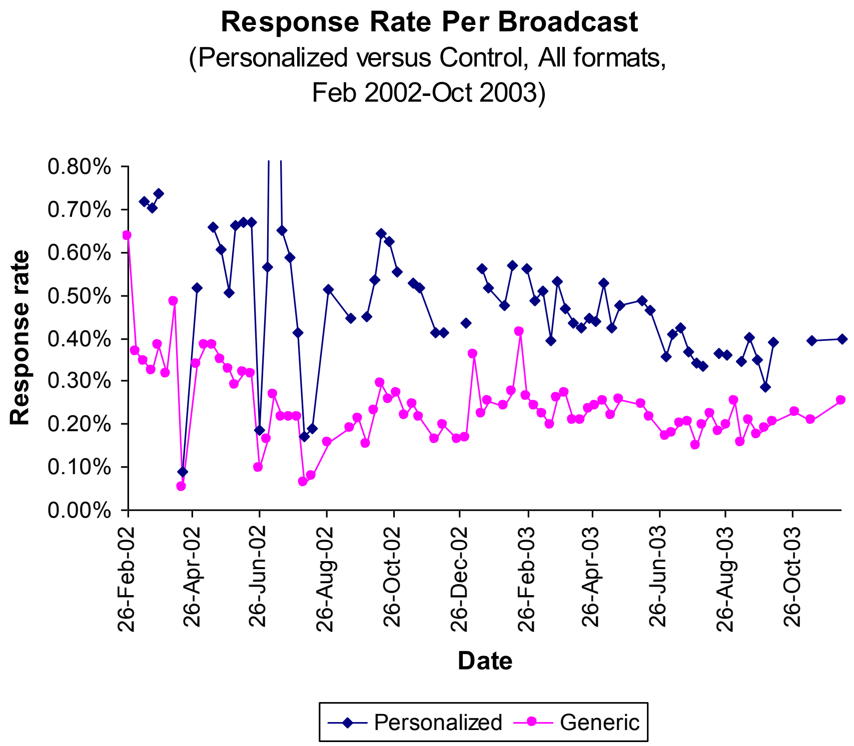
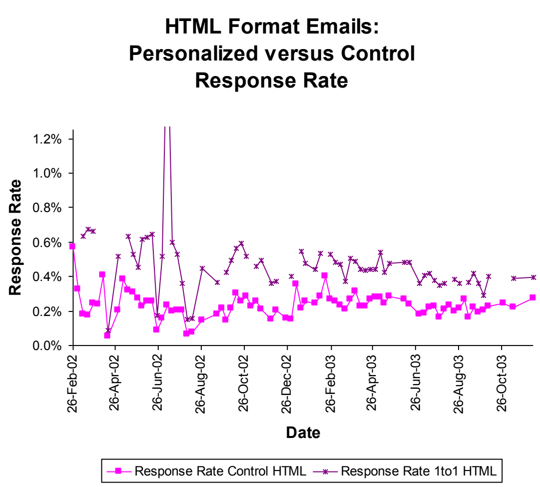
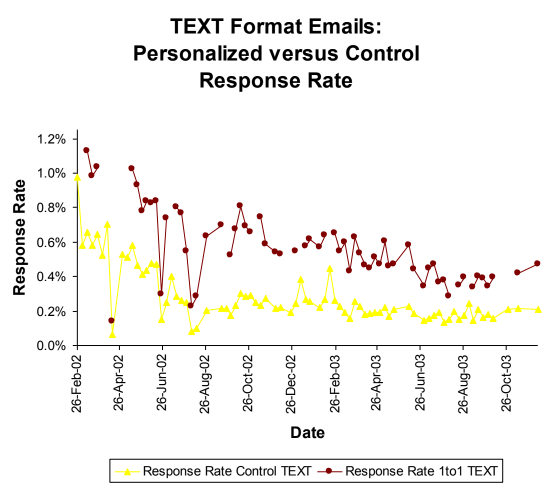


Figure 9: Personalized versus control emails week by week





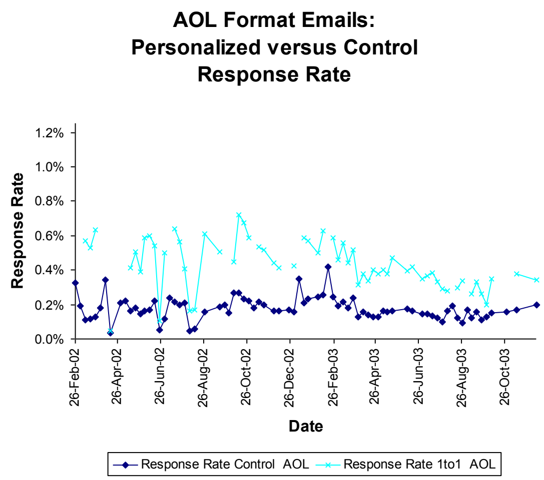


Figure 10: Impact of personalization for different email formats. All show a similar increase in response rate.

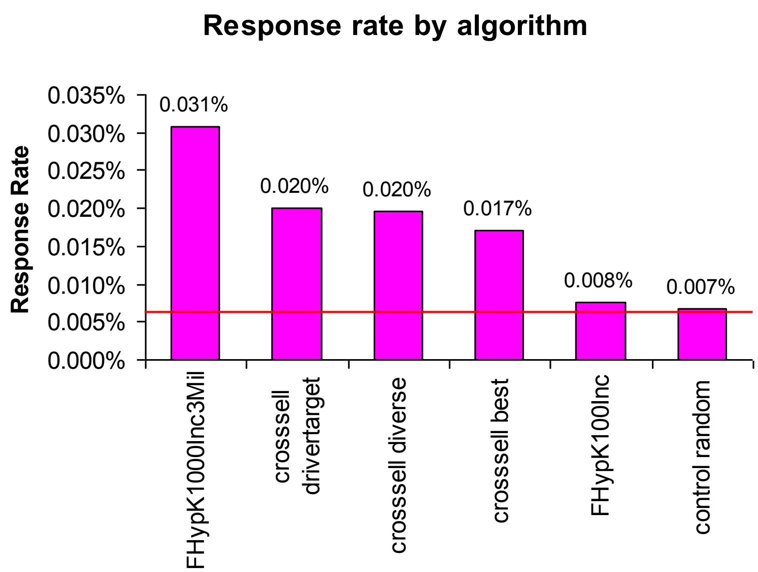


Figure 13: Overall Response rate per SKU exposure for live email customers who received targeted products from one of 6 methods. Control-random selected a product at random. The cross-sell variants were described in Kitts, et. al. (2000).

Channel Efficiency

Channel	Response Rate	Cost Per Solicitation	Cost Per Customer Sale
Retail	13	1/1.43	½
Catalog	10	1	1
Email Personalized	2	1/8333	1/1666
Email Generic	1	1/8333	1/833

Figure 14: Response rate, cost per solicitation, and cost per sale, for different channels at Central Purchasing, Inc, as compiled from actual marketing response data. These numbers were indexed to avoid disclosing business costs; however, they clearly show why email is attractive.

CONCLUSION

One-to-one systems have been one of the biggest commercial success stories of marketing technology in the last 15 years. Nearest Neighbor approaches are a natural vehicle for 1to1 because they are capable of running against huge datasets, are heterogeneous, simple, and produce excellent recommendations. Many interesting new applications are becoming possible because of the ability to scale these algorithms. For instance, Yahoo! recently launched a music recommendation system, and after rating a few pieces of music, will create a personalized "radio station" that only plays music that the user likes.

At present one-to-one is still a technology that is at the disposal of the marketing department. However, we believe that in the future, users will be able to tune their recommendation algorithms. A user could adjust the algorithm to show more or less novelty, only products below a certain price, or only products within certain categories. Such a model empowers the consumer, allows them to become an active participant in their own service fulfillment. Response rates could increase, consumers may become more loyal and engaged, but marketers would also be challenged by a model in which consumers provide input into the type of offers they receive.

The emergence of one-to-one presents new opportunities and challenges for marketers and consumers alike. However, in the end, consumers - ordinary people - stand to benefit the most from these new technologies, as systems become more adept and helping them in their day-to-day activities.

REFERENCES

PMML v3.0, Predictive Model Markup Language - Association Rules, Data Mining Group website, http://www.dmg.org/v3-0/AssociationRules.html

Group Lens website http://www.cs.umn.edu/research/GroupLens/index.html

Movie Lens website http://movielens.umn.edu   MovieLens website. Movie recommendation system.

EachMovie dataset. http://research.compaq.com/SRC/eachmovie

Form 10-K for NET PERCEPTIONS INC, http://biz.yahoo.com/e/050331/netp.pk10-k.html

Problems with Soul-Mate Searches? *ABC News Nightline*, December 10, 1999. http://www.abcnewsstore.com/store/index.cfm?fuseaction=customer.product&product_code=N991210%2001&category_code=37

Data Editing and Imputation, *Survey of Income and Program Participation User's Guide (SIPP)*, US Census Bureau, http://www.bls.census.gov/sipp/tutorial/SIPP_Tutorial_Beta_version/SIPPtutorialHTML/PDF/C04.PDF

Adomavicius, G. and Tuzhilin, A. (2001), Extending Recommender Systems: A Multidimensional Approach, *Proceedings of the IJCAI Workshop on Intelligent Techniques for Web Personalization*, August 2001.

Agrawal, R. and Srikant, R, (1994) Fast algorithms for mining association rules, *Proceedings of the 20th International Conference on Very Large Databases*, Santiago, Chile

Ali, K. and van Stam, W. (2004), TiVo: Making Show Recommendations Using a Distributed Collaborative Filtering Architecture, *Proceedings of the 10th International Conference on Knowledge Discovery and Data Mining*, pp. 394-401.

Ansari, A., Essegaier, S. and Kohli, R. (2000), Internet Recommendation Systems, *Journal of Marketing Research*, Vol. 37, pp. 363-374. August.

Balabanovic, M. and Shoham, Y. (1997) Content-based collaborative recommendation, *Communications of the ACM*, Vol. 40, pp. 66-72, March

Billsus, D. and Pazzani, M. (1998), Learning Collaborative Information Filters, *Proceedings of the International Conference on Machine Learning*, pp. 46-53.

Breese, J., Heckerman, D., Kadie, C., (1998), Empirical Analysis of Predictive Algorithms for Collaborative Filtering, *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence* (UAI 1998), pp. 43-52. Morgan Kaufmann, Madison, WI.

Burns, C. and Wyden, R. (2003), CAN-SPAM Act of 2003, Pub. L. 108-187, S. 877, Full text available from http://www.spamlaws.com/federal/summ108.shtml#s877

Chan, P. (1999) A non-invasive learning approach to building web user profiles, *Workshop on Web usage analysis and user profiling, Fifth International Conference on Knowledge Discovery and Data Mining*, San Diego.

Chesson, J (1976), A non-central multivariate hypergeometric distribution arising from biased sampling with application to selective predation. *Journal of Applied Probability*, vol. 13, no. 4, pp. 795-797.

Chien, Y. and Edward, G. (1999), A Bayesian model for Collaborative filtering, University of Texas, Austin.

Chien, Y. and George, E. (1999), Bayesian Collaborative Filtering, in Heckerman, D. and Wittaker, J. (eds), *Artificial Intelligence and Statistics*, Morgan Kaufmann, pp. 187-192.

David, M., Little, R., et. al. (1986) Alternative methods for CPS income imputation, *Journal of the American Statistical Association*, Vol. 81, No. 393, pp. 29-41.

De Bruyn, A., Liechty, J., Huizingh, E., Lilien, G. (2003), Offering Online Recommendatoins to Impatient, First-time customers with conjoint based segmentation trees, *Proceedings of the 2003 European Marketing Academy Conference*.

Deshpande, M. and Karypis, G. (2005), Item-Based Top-N Recommendation Algorithms, *ACM Transactions on Information Systems*, Vol. X, No. X, pp. 1-34.

Drineas, P., Kannan, R., Mahoney, M. (2004a), Fast Monte Carlo Algorithms for Matrices III: Computing an Efficient Approximate Decomposition of a Matrix, Technical Report, YALEU/DCS/TR-1271, February, accepted for publication, SIAM J. Computing.

Drineas, P., Kannan, R., Mahoney, M. (2004b), Fast Monte Carlo Algorithms for Matrices II: Computing Low-Rank Approximations to a Matrix, Technical Report, YALEU/DCS/TR-1270, February, Accepted for publication, SIAM J. Computing.

Elkan, C. (1997) Boosting and Naïve Bayesian Learning, Technical Report No. CS97-557, Department of Computer Science and Engineering, University of California, San Diego.

Ellsworth, G. (1998a), Automated Collaborative Filtering System, USPTO 5790426

Ellsworth, G. (1998b), Automated Collaborative Filtering System, USPTO 5884282

Fog, A. (2005), Sampling Methods for Wallenius and Fisher Noncentral Hypergeometric Distributions, http://www.agner.org/random/theory/nchyp1.pdf

Ford, B., (1980) An Overview of Hot-Deck Procedures, in Madow, W. et. al. (eds), *Incomplete data in sample surveys*, Vol 2., Academic Press, NY. pp. 185-206.

Freed, D., Kitts, B. (2000), Data Storage and Processing Method for Fast Memory-Efficient Automated Collaborative Filtering on Sparse Data, US Patent and Trademark Office, Patent Application.

Frieze, A., Kannan, A., Vempala, S. (2004), Fast Monte-Carlo Algorithms for finding low-rank approximations, *Journal of the ACM*, Vol. 51, Iss. 6, pp. 1025-1041, ACM Press, NY

Greening, D. (2000), Building Consumer Trust with Accurate Product Recommendations: A White Paper on Macromedia's LikeMinds Personalization Technology, Macromedia, Inc.

Goldberg, D., Nichols, D., Oki, B. and Terry, D. (1992), Using collaborative filtering to weave an information tapestry, *Communications of the ACM*, Vol. 35, No. 12, pp. 61-70.

Herlocker, J., Konstan, J., Borchers, A., Riedl, J. (1999) An Algorithmic Framework for Performing Collaborative Filtering. *Proceedings of the 1999 Conference on Research and Development in Information Retrieval*.

Herlocker, J. Konstan, L. Terveen and J. Riedl, (2004) Evaluating Collaborative Filtering Recommender Systems. *ACM Transactions on Information Systems* Vol. 22, No. 1, pp. 5-53, January.

Herz, F., Eisner, J., et. al. (1998), System for Generation of User Profiles for a System for Customized Electronic Identification of Desirable Objects, US Patent 5,754,939, US Patent and Trademark Office. http://www.pinpointincorporated.com/Patents/Patent_5754939.pdf

Hey, J. (1989), System and Method of Predicting Subjective Reactions, United States Patent 4,870,579, US Patent and Trademark Office.

Hey, J. (1991), System and Method for Recommending Items, United States Patent 4,996,642, US Patent and Trademark Office

Hill, W., Stead, L. et. al. (1995), Recommending and Evaluating Choices in a Virtual Community of Use, in *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*

Jacobi, J. and Benson, E. (2000), System and methods for collaborative recommendations, United States Patent 6,064,980, US Patent and Trademark Office. http://164.195.100.11/netacgi/nph-Parser?Sect1=PTO1&Sect2=HITOFF&d=PALL&p=16064980

Jannarkar, S. (2003), Patent Company sues Amazon, News.com, http://news.com.com/2100-1017-1027343.html?part=dht&tag=ntop

Johnson, S. (1998), The soul encoded: software that thinks it knows you.(intelligent agents offer advice), *Harper's Magazine*; 9/1/1998

Karypis, G. (2001), Evaluation of Item-Based Top-N Recommendation Algorithms, *Proceedings of the 2001 ACM CIKM International Conference on Information and Knowledge Management*, Atlanta, Georgia, USA, November 5-10, 2001. ACM 2001, ISBN 1-58113-436-3

King-Ip Lin, Congjun Yang (2001), The ANN-tree: An index for efficient approximate nearest neighbor search , *Proceedings of the Seventh International Conference on Database Systems for Advanced Applications*, April.

Kitts, B., Freed, D. and Vrieze, M. (2000), Cross-sell, *Proceedings of the 6th International ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, Boston, MA. MIT Press.

Konstan, J., Miller, B. et. al. (1997), Applying collaborative filtering to usenet news, *Communications of the ACM*, Vol. 40, pp. 77-87. March

Lewis, D. (1998), Naive Bayes at Forty: The Independence Assumption in Information Retrieval, *Proceedings of the Tenth European Conference on Machine Learning*.

Li, R. (2003), Empirical Analysis of Predictive Algorithms for Recommender Systems, *First Instructional Conference on Machine Learning.http://www.cs.rutgers.edu/~mlittman/courses/ml03/iCML03/schedule.html*

Little, R. and Rubin, D., *Statistical Analysis with Missing Data*, John Wiley and Sons, NY. (1987).

Linden, G., Smith, B. and York, J. (2003), Amazon.com Recommendations: Item-to-item Collaborative Filtering, *IEEE Internet Computing*, IEEE Computer Society, January-February.

Maltz, D. and Ehrlich, K. (1995), Pointing the way: Active collaborative filtering, *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pp. 202-209.

Mobasher, B., Dai, H., Luo, T., Nakagawa, M. and Witshire, J. (2000), Discovery of aggregate usage profiles for web personalization. In *Proceedings of the WebKDD Workshop, at the Sixth International Conference on Knowledge Discovery and Data Mining*.

O'Brien, J. (1983), interview with Morris Hansen, former Associate Director for Research and Development at the US Census Bureau, *US Census Bureau Oral Histories*, June 22, 1983, http://www.census.gov/prod/2003pubs/oh-hanse.pdf

Oh, H. and Scheuren, F., Estimating the Variance impact of Missing CPS Income Data, *Proceedings of the Survey Research Methods Section, American Statistical Association*, pp. 408-415. (1980).

Paige, E. (1994), Blockbuster experiments with computerized title suggestions. (Blockbuster Entertainment tests movie-recommendation system), *Video Business*, October 28, http://static.highbeam.com/v/videobusiness/october281994/blockbusterexperimentswithcomputerizedtitlesuggest/

Papagelis, M. and Dimitris P. (2005), Qualitative Analysis of User-based and Item-based Prediction Algorithms for Recommendation Agents, *Journal of Engineering Applications of Artificial Intelligence* Vol 18, No. 4, June.

Peng, C.-Y. J., Harwell, M., Liou, S.-M., & Ehman, L. H. (2005). Advances in missing data methods and implications for educational research. In S. Sawilowsky (ed.), *Analyzing real data*. Washington, DC: American Educational Research Association.

Pennock, D., Horvitz, e., Lawrence, S. and Giles, C. (2000), Collaborative filtering by personality diagnosis: A hybrid memory- and model-based approach, in *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, pp. 473-480.

Polikar, R., Udpa, L., Udpa, S., and Honavar, V. (2001). Learn++: An Incremental Learning Algorithm for Multi-Layer Perceptron Networks. *IEEE Transactions on Systems, Man, and Cybernetics*. Vol. 31, No. 4. pp. 497-508.

Popescul, A., Ungar, L., Pennock, D. and Lawrence, S. (2001), *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence*, (UAI 2001), pp. 437-444, Morgan Kaufmann, San Francisco.

Press, W., Flannery, B., Teukolsky, S., and Vetterling, W. (1988), *Numerical Recipes in C*, Cambridge University Press, NY.

Rashtchy, S. and Avilio, J. (2003), *The Golden Search: Dynamics of the Online Search Market and the Scope of Opportunity*, US Bancorp Piper Jaffray Equity Research. Available by request from US Bancorp.

Resnick, P. and Varian, (1997), Recommender Systems, *Communications of the ACM*, Vol. 40, No. 3, pp. 56-58.

Resnick P, Iacovou N., Suchak M., Bergstrom, and Riedl J. (1994), "GroupLens: An open architecture for collaborative filtering of netnews", *Proceedings of ACM 1994 Conference on Computer Supported Cooperative Work*, Chapel Hill, NC: ACM, 175-186.

Riedl, J., Soul Mate Searches (2000), *Proceedings of the Personalization Summit 2000*, Boston. April.

Robinson, G., (1998) Automated collaborative filtering system, US Patent and Trademark Office, http://164.195.100.11 .

Robinson, G. and Ellsworth, M. (1998), Automated Collaborative Filtering System, United States Patent 5,790,426, US Patent and Trademark Office.

Robinson, G. and Elssworth, M. (1999), Automated collaborative Filtering System, United States Patent 5884282, US Patent and Trademark Office.

Robinson, G. (2003), A Discussion of the Pinpoint vs. Amazon Case, Personal web page, http://radio.weblogs.com/0101454/stories/2003/07/19/aDiscussionOfThePinpointVs.html

Robertson, S. (2004) Understanding inverse document frequency: on theoretical arguments for IDF. *Journal of Documentation* Vol. 60, 503-520.

Rubin, D., (1996) Multiple imputation after 18+ years, *Journal of the American Statistical Association*, Vol. 91, pp. 434, 473-489.

Salton, G. and McGill, M. (1983), *Introduction to Modern Information Retrieval*, McGraw-Hill.

Salton, G. (1989), *Automatic Text Processing: The Transformation, Analysis and Retrieval of Information by Computer*, Addison-Wesley.

Sarwar, B., Karypis, G., Konstan, J. and Reidl, J. (2001), Item-based collaborative filtering recommendation algorithms, *World Wide Web*, pp. 285-295.

Schafer, J., *Analysis of Incomplete Multivariate Data*, Chapman and Hall, London. (1997).

Schafer, J. and Olsen, M. (1997) Multiple imputation for multivariate missing-data problems: a data analyst's perspective, *Multivariate Behavioural Research*, Vol. 33, pp. 545-571. also available from http://www.stat.psu.edu/~jls.

Schafer, J. (1999) Multiple imputation: a primer, *Statistical Methods in Medical Research*, Vol. 8, pp. 3-15.

Shardanand, U. and Maes, P. (1995), Social information filtering: algorithms for automating "word of mouth" *Proceedings of ACM Conference on Human Factors in Computing Systems (CHI95)*, pp. 210-217.

Spärck Jones, K. (1972) Exhaustivity and specificity. *Journal of Documentation* 28, 11-21, reprinted in Vol. 60, pp. 493-502, 2004, Letter and reply, *Journal of Documentation* 28, 164-165, 1972

Terveen, L., et. al. (1997), PHOAKS: A system for sharing recommendations, *Communications of the ACM*, Vol. 40, pp. 59-62

Ungar, L. and Foster, D. (1998), Clustering methods for collaborative filtering, in *Workshop on Recommendation Systems at the 15th National Conference on Artificial Intelligence*.

Wallenius, K. (1963) *Biased Sampling: The Non-central Hypergeometric Probability Distribution*. Ph.D. thesis, Stanford University.

Wyner, G. (1998), Collaborative Filtering: Research or IT?, *Marketing Research*, Fall98, Vol. 10, Issue 3, pp. 35-38.