COVER SHEET FOR PROPOSAL TO THE NATIONAL SCIENCE FOUNDATION

PROGRAM ANNOUNCEMENT/SOLICITATION NO./CLOSING DATE/if not in response to a program announcement/solicitation enter NSF 04-2

NSF 04-524 **03/03/04**

FOR CONSIDERATION BY NSF ORGANIZATION UNIT(S) (Indicate the most specific unit known, i.e. program, division, etc.)

IIS - CYBER TRUST

FOR NSF USE ONLY

NSF PROPOSAL NUMBER

0430303

DATE RECEIVED	NUMBER OF COPIES	DIVISION ASSIGNED	FUND CODE	DUNS# (Data Universal Numbering System)	FILE LOCATION
				825753379	

EMPLOYER IDENTIFICATION NUMBER (EIN) OR TAXPAYER IDENTIFICATION NUMBER (TIN)

362167048

SHOW PREVIOUS AWARD NO. IF THIS IS
☐ A RENEWAL
☐ AN ACCOMPLISHMENT-BASED RENEWAL

IS THIS PROPOSAL BEING SUBMITTED TO ANOTHER FEDERAL AGENCY? YES ☐ NO ☒ IF YES, LIST ACRONYM(S)

NAME OF ORGANIZATION TO WHICH AWARD SHOULD BE MADE

DePaul University

AWARDEE ORGANIZATION CODE (IF KNOWN)

0016717000

ADDRESS OF AWARDEE ORGANIZATION, INCLUDING 9 DIGIT ZIP CODE

DePaul University
1 East Jackson Boulevard
Chicago, IL. 606042218

NAME OF PERFORMING ORGANIZATION, IF DIFFERENT FROM ABOVE

PERFORMING ORGANIZATION CODE (IF KNOWN)

ADDRESS OF PERFORMING ORGANIZATION, IF DIFFERENT, INCLUDING 9 DIGIT ZIP CODE

IS AWARDEE ORGANIZATION (Check All That Apply) (See GPG II.C For Definitions)
☐ SMALL BUSINESS ☐ MINORITY BUSINESS
☐ FOR-PROFIT ORGANIZATION ☐ WOMAN-OWNED BUSINESS

☐ IF THIS IS A PRELIMINARY PROPOSAL THEN CHECK HERE

TITLE OF PROPOSED PROJECT **Secure Personalization: Building Trustworthy Recommender Systems**

REQUESTED AMOUNT	PROPOSED DURATION (1-60 MONTHS)	REQUESTED STARTING DATE	SHOW RELATED PRELIMINARY PROPOSAL NO. IF APPLICABLE
$ **366,186**	**36** months	**08/01/04**	

CHECK APPROPRIATE BOX(ES) IF THIS PROPOSAL INCLUDES ANY OF THE ITEMS LISTED BELOW

☐ BEGINNING INVESTIGATOR (GPG I.A)
☐ DISCLOSURE OF LOBBYING ACTIVITIES (GPG II.C)
☐ PROPRIETARY & PRIVILEGED INFORMATION (GPG I.B, II.C.1.d)
☐ HISTORIC PLACES (GPG II.C.2.j)
☐ SMALL GRANT FOR EXPLOR. RESEARCH (SGER) (GPG II.D.1)
☐ VERTEBRATE ANIMALS (GPG II.D.5) IACUC App. Date _____
☐ HUMAN SUBJECTS (GPG II.D.6)
Exemption Subsection _____ or IRB App. Date _____
☐ INTERNATIONAL COOPERATIVE ACTIVITIES: COUNTRY/COUNTRIES INVOLVED (GPG II.C.2.g.(iv).(c))
_____
☐ HIGH RESOLUTION GRAPHICS/OTHER GRAPHICS WHERE EXACT COLOR REPRESENTATION IS REQUIRED FOR PROPER INTERPRETATION (GPG I.E.1)

PI/PD DEPARTMENT
School of CTI

PI/PD FAX NUMBER
312-362-6639

PI/PD POSTAL ADDRESS
243 S. Wabash Ave.

Chicago, IL 60604
United States

NAMES (TYPED)	High Degree	Yr of Degree	Telephone Number	Electronic Mail Address
PI/PD NAME **Robin Burke**	**PhD**	**1993**	**312-362-5910**	**rburke@cs.depaul.edu**
CO-PI/PD **Bamshad Mobasher**	**PhD**	**1994**	**312-362-5174**	**mobasher@cs.depaul.edu**
CO-PI/PD				
CO-PI/PD				
CO-PI/PD				

Electronic Signature

Project Summary

Personalization systems are an increasingly important component of electronic commerce systems. Users have come to trust personalization and recommendation software to reduce the burden of navigating large information spaces and choosing from product catalogs. The success of these systems, both for users and implementers, is dependent upon the properties of their recommendations: ideally objective, unbiased and meaningful. However, because recommendation systems are dependent on external information sources, they are vulnerable to attack. If a system generates recommendations collaboratively, that is by user-to-user comparison, hostile users might generate large numbers of biased user profiles for the purpose of disorting the system's recommendations. If on the other hand, the system uses properties associated with products, producers may label their items deceptively to generate more recommendations. **The purpose of the research described in this proposal is to explore the vulnerabilities of recommendation systems, explore techniques for enhancing their robustness, and examine methods by which attacks can be recognized and possibly defeated.**

Intellectual Merit

This project advances our understanding of the trustworthiness of recommender systems, now a crucial component in many e-commerce areas. We will explore the spectrum of possible attacks against recommendation systems, and develop formal models characterizing these attacks and their impacts. We will investigate different metrics for assessing the robustness of recommendation algorithms including accuracy, stability and expected payoff to the attacker. In tandem with this theoretical work, we will conduct empirical investigations using data from a variety of domains. We will test a range of recommendation algorithms including user-based, item-based and model-based collaborative recommenders, and also explore hybrid recommendation by combining collaborative recommendation techniques with content-based and knowledge-based ones. Finally, informed by these results, we will consider how recommender systems can be secured, through improved algorithms but also by detecting attacks and responding appropriately.

Broader Impacts

The results of our research into secure personalization will be valuable to implementers of recommender systems, helping to improve the robustness of e-commerce systems. In doing so, it will have an impact on the many commercial users of recommender systems, all of whom face potential attacks. Implementers will be able to use our results to evaluate recommendation algorithms, not just on their accuracy and efficiency, but also on their security. In addition, as the example of the popular Google search engine shows, a technique that protects against attack may also improve the overall performance of a system by reducing noise and by bringing additional knowledge and information sources to bear, so improved recommendation performance may be an additional benefit of this project.

Most research in computer security focuses on protecting assets inside an organization's security perimeter from unauthorized access and modification. This project examines the problem of security for systems that are designed to be accessed and modified by the general public. How do we protect such a system from the legal but biased inputs of an attacker trying to subvert its functionality? This question shows that the results of our research will also have significant implications for a variety of adaptive information systems that rely on users' input for learning user or group profiles. Such systems include adaptive hypertext systems, personalized e-learning environments, adaptive information filtering systems (e.g., email spam filtering applications), and collaborative group-based environments. Many such systems have open components through which a malicious user or an automated agent can affect the overall system behavior.

Finally, the proposed activity and the resulting systems, will serve as an integral part of several graduate courses in different programs at the host institution, including the newly created Computer, Information and Network Security program which has been certified by the National Committee on Security Standards. This project will also be actively used as a basis for providing research opportunities for undergraduate students.

Project Description

1 Introduction

Recommendation systems are an increasingly important component of electronic commerce systems. Users have come to trust personalization and recommendation software to reduce the burden of navigating large information spaces and choosing from product catalogs. The preservation of this trust is important both for users and site owners and is dependent upon the perception of recommender systems as objective, unbiased and meaningful. However, because recommendation systems are dependent on external sources of information, they are vulnerable to attack. If a system generates recommendations collaboratively, that is by user-to-user comparison, hostile users can generate large numbers of biased user profiles for the purpose of distorting the system's recommendations.If on the other hand, the system uses properties associated with products, producers may label their items deceptively to generate more recommendations.

Consider a recommender system that identifies books that users might like to read based on a collaborative algorithm. Alice, having built up a profile from previous visits, returns for new recommendations. Unbeknownst to her, Eve has inserted a large number of biased profiles into the system, all of which give high ratings to Z, a book she has written. See Figure 1. If this so-called "push attack" is successful, Alice will get Z as a recommendation, regardless of whether this is really the best suggestion for her or not. She may find the suggestion inappropriate, or worse, she may take the system's advice and then be disappointed by the delivered product.

	A	B	C	D	E	...	Z	
U1	+	–	+	+	–			
U2	–		+		+		–	
U3	–	–		+			–	
U4		+		–	+		–	Actual Profiles
U5	+	–	–	+	–			
U6		+	+	+	–		–	
E1	+	–	+	+	–		+	
E2	–		+		+		+	
E3	–	–		+	–		+	
E4		+		–	+		+	Biased Profiles
E5	+	–	–	+	–		+	
E6		+	+	+	–		+	
Alice	+	–		+	–		?	← Target User

Figure 1: A "push" attack

This is the essence of the problem of secure recommendation. In fact, this particular example has currency: the Associated Press reported in February of 2004 that a security glitch at the online book store Amazon.com had caused the real identities of its book reviewers to be made public, and buyers found that many authors were indeed posting glowing reviews of their own work [36]. How can a recommendation system, which is dependent on user-generated profiles, provide trustworthy advice? Recommender systems that are not protected against attack will have diminished value and will not achieve the goal of providing unbiased guidance.

Our primary objective in this research is to conduct a comprehensive study of the robustness of recommender systems in the face of malicious attacks. Our analysis will be multidimensional; examining the effects of a range of attack types on a comprehensive set of recommendation algorithms including hybrid approaches using different types of user profiles in diverse recommendation domains.

O'Mahoney, et al. [35] conducted some preliminary research on the problem of the robustness of collaborative recommendation. The authors develop a formal framework for analyzing one type of attack (the

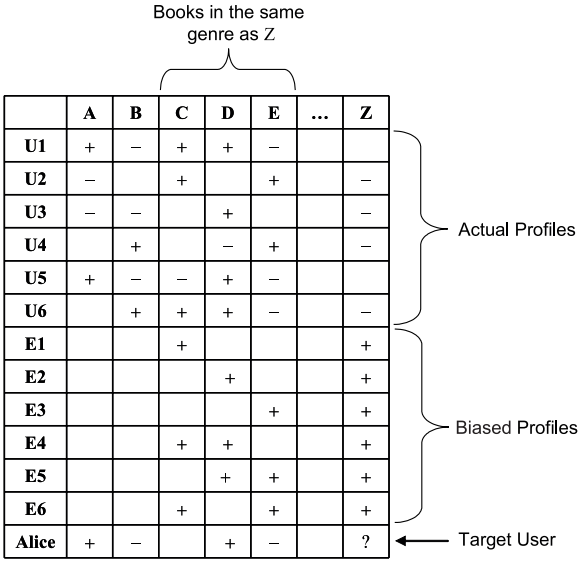

Figure 2: A segmented push attack

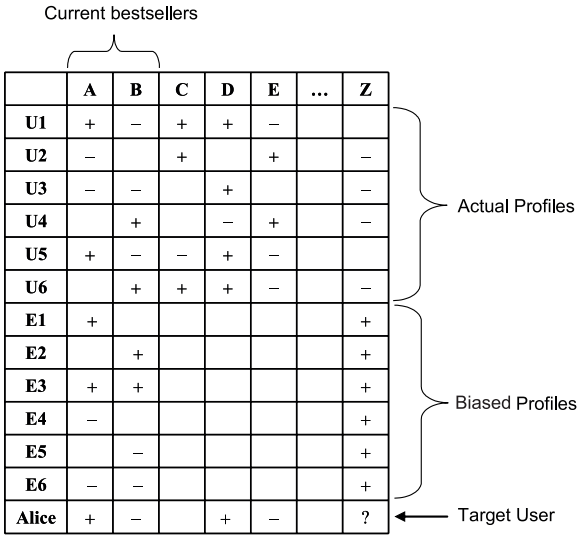

Figure 3: A bandwagon attack

"nuke" attack), one type of bias (class noise) and one type of ratings (explicit binary ratings), and they use this approach to examine the classic nearest-neighbor collaborative recommendation algorithm. We will take this work as our starting point for this project, and expand it along the dimensions outlined above.

1.1 Types of Attacks

[35] defines two styles of attack: "push" and "nuke", defined by their intended effect on a given item in the product space. An attacker may insert profiles to make Z more likely to be recommended ("push") or to make it less likely ("nuke"). Their paper assumes that the bias with respect to class is the only perturbation introduced by the attacker. In other words, Eve's biased profiles look just like regular profiles in terms of their distribution of items and ratings: the only difference is that they give Z a positive rating much more frequently. While this contributes to the analytical clarity of their approach, it is probably not realistic that the attacker will be able to craft biased profiles so thoroughly.

To examine more realistic attacks, we will investigate various forms of attribute bias, in addition to the class bias. One form that such bias may take is through a "segmented" attack. In this attack, the attacker identifies a subset of the products as targets based on a heuristic, for example those books in a related genre, and launches an attack designed to affect all of them. For example, Eve may identify other books that share the same general topic as her book and craft profiles using them. She might decide to discredit them all, in the hopes of having Z rise in a relative way, or she might build push profiles that incorporate them, hoping that users with an interest in the topic will be directed towards Z, as shown in Figure 2. Given the difficulty of acquiring full knowledge of the ratings distribution in the system, such heuristic-based attacks seem to be highly probable.

An attacker may also combine class and attribute noise in a combination push/nuke attack, promoting one product and denigrating others. However, given the need to achieve a match against existing profiles, the simplest incarnation of such an attack (negative profiles with a single positive rating for the pushed product) will probably not be successful. The design of such a combination attack would have to take some of the properties of the recommendation algorithm into account.

Another type of attack uses the application of Zipf's law: a small number of products, best-seller books for example, get the lion's share of attention, and hence are more likely to appear in user profiles. By associating her book with current best-sellers, Eve can ensure that her bogus profiles have a good probability of matching any given user, since so many users will have these items on their profiles. This might be called a "bandwagon

attack". Note that the type of rating given to the popular items is irrelevant: not every user will have a positive opinion of a well-known work, but most will have some opinion. See Figure 3.

There are a variety of other attack types, including various combinations of the aforementioned attacks. One goal of our research is to catalog different types of attacks, including those that may be pertinent to specific algorithms or data types, and to develop models of successful attack profiles, in terms of changes in the distribution of data before and after an attack. Such models will, in turn, be potentially helpful in automatically detecting and countering such attacks.

1.2 Recommendation Algorithms

Nearest-neighbor collaborative filtering is the classic formulation of the collaborative model. A user such as Alice is compared against others known to the system. The most proximal neighbors are selected and their profiles are used as a basis for predicting Alice's ratings for items as yet unrated by her. It is this model on which O'Mahoney and his colleagues base their analysis. However, there are numerous other formulations of the collaborative technique, including model-based techniques in which a model is learned from the user profiles and then applied to Alice's profile. Bayesian networks, association rules, and latent semantic analysis are just a few of the techniques that have been applied to the problem of collaborative recommendation [8, 31, 6]. In this project, we will extend the empirical work done in [35] to examine some selected model-based algorithms and determine if these methods offer improved robustness over instance-based ones.

Another class of recommender system uses algorithms based not on user ratings alone but also information about the items themselves. A content-based recommender, for example, induces a classifier based on a particular user's ratings, and then produces recommendations on that basis [27, 34]. A knowledge-based recommender reasons about the fit between a user's need and the features of available products [11]. These systems are vulnerable to a different form of attack. False profiles do not have any effect, but often such systems depend on external sources for their content information: a book recommender system does not read books, and so must rely on the features available in its catalog. If such features are supplied by authors or publishers and not validated, the data may be biased. This phenomenon is well-known in the web search industry as "search engine spam"[1]: web page authors attach misleading keywords, metadata and other information to their pages in the hopes of being retrieved in response to popular queries.

An effective response to the problem of biased content is to combine the use of content with the use of collaborative information. Much of the success of the Google search engine[2] can be attributed to its use of an authority measure (effectively a collaboratively-derived weight) in addition to standard content-based metrics of similarity in query processing [9]. This hybrid technique means that a page that is misleadingly labeled is much less likely to be retrieved by Google than by a system that uses only content. Google is therefore an example of a hybrid approach to secure recommendation, defending against a biased content attack through the addition of collaborative information.

Hybrid recommendation, combining multiple recommenders of different types, is therefore a promising approach for securing recommender systems. In previous work, one of the principal investigators has developed a taxonomy of hybrid recommendation and explored the relative accuracy of different hybrid formulations [12, 10]. We propose to use this taxonomy in this project as a roadmap for exploring the robustness of different hybrids and the relative advantage of hybrid recommendation over recommendation techniques used in isolation. With these hybrids, we will also be able to explore attacks that involve biased content as well as those that use biased profiles.

1.3 Types of User Profiles

The research described in [35] makes use of explicit binary ratings data: products are individually and explicitly rated by the user as liked and disliked. Recommender systems today are more likely to make use of implicit ratings, ratings that are inferred from user behavior, rather than explicitly provided by the user.

[1] http://www.google.com/contact/spamreport.html
[2] www.google.com

(See the research reviewed in [25].) Such data sources may have different characteristics than the classic explicit rating scenario. In Web usage mining [15, 41], Web server logs are examined for link traversal and dwell time and continuously-valued ratings derived from this analysis, and as a result, negative ratings are not available from Web usage data. We may infer that if page A got less attention than page B that page B is liked more, but not necessarily that page A was disliked.

Web usage mining techniques, such as clustering and association rule discovery, that rely on offline pattern discovery from user transactions, have been studied as an underlying mechanism for personalization and recommender systems [30, 31, 32]. Such techniques generally provide both a computational advantage as well as better recommendation effectiveness than traditional collaborative techniques, particularly in the context of click-stream data.

In the knowledge-based recommender system Entree, users employ critique-based navigation to home in on a desirable restaurant [11]. EntreeC, a hybrid version of this system employing collaborative filtering, interprets user actions as implicit ratings: the critiques of each unsatisfactory restaurant encountered along the way are negative ratings, and the final endpoint, a positive rating [12]. In this data set, negative ratings far outweigh positive ones.

Any attempt to secure such recommenders against attack must take into account their unique characteristics both in terms of the structure of the profiles and how profiles they are collected. In Entree, for example, an attacker cannot simply call up a list of restaurants and assign them negative ratings. A more sophisticated softbot attack must be mounted in which a software agent interacts with the Entree application in a predefined way. Similarly, an attack against a usage-based web personalization system would have to proceed through a crawling mechanism that downloaded pages and performed the appropriate actions to push or nuke them. Such mechanisms are certainly not beyond the capabilities of attackers, but they may leave characteristic signatures in the interaction traces from which implicit ratings are derived. As part of our research, we will explore how the attacks that we envision may be realized against systems using a variety of profile types and data collection methods, including implicit or explicit ratings, critique-based profile data, and semantically derived profiles (such as those in content-based or knowledge-based recommender systems). In each of these cases, we will investigate how bias may be modeled and detected in such inputs.

1.4 Summary of Proposed Activity

Our proposed research will involve the following main areas of activity:

- We will develop formal models for the analysis of robustness in recommender systems. The models will include frameworks for analyzing recommendation accuracy, prediction stability, and expected payoff, in the face of different types of malicious attacks.

- We will explore and model different attack types, from the simple biased class noise attacks discussed above to push/nuke attacks, segmented attacks, bandwagon attacks, and other combinations, which involve different degrees of attacker knowledge about the domain, about the data distribution, and about the recommendation algorithms. Our models will focus on changes in the distribution of the underlying profile data before and after an attack has taken place. We will consider the profiles of successful attacks and consider how such attacks may be detected and countered.

- We will explore the vulnerabilities of recommender systems to various forms of attack, both through formal modeling of the system's capabilities and possible attack strategies, and through empirical investigation of the stability and accuracy of an array of common recommendation algorithms in the face of different attack types. Specifically, we will study recommendation algorithms for user-based and item-based collaborative filtering; model-based approaches, such as those based on association rules and latent variable models; as well as content- and knowledge-based techniques.

- Finally, we will explore techniques for enhancing the robustness of recommender systems, through improved algorithms, including a variety of hybrid algorithms, and through alternative techniques for gathering and using implicit or explicit profile data.

2 A Framework for Modeling Attacks

Generally speaking, a malicious attack against a recommender system can impact the system in two different, but related ways. The attack may affect the accuracy of recommendations produced by the system. There are a variety of approaches for measuring predictive accuracy, most of which rely on a comparison of correct recommendations produced by the system (as measured with respect to a evaluation data set) to all generated recommendations. An attack against the system may also have an impact on the stability of the system. In essence, stability is a measure of changes in the recommendation sets before and after the attack (regardless of whether the generated recommendations are correct). Accuracy and stability measure the impact of an attack from the system perspective. Another valuable measure we would like to examine is *expected payoff*: to what extent does an attack benefit the attacker? As a starting point we begin with previously studied models. We will then explore various extentions of these models within the scope of this project.

2.1 Accuracy Analysis

In the context of classification, an attack can be viewed as the addition of noise to the training data from which the classification model is derived. The noise can be associated with the target class (the attribute or item which is the subject of attack) or with non-class attributes. Previous work in this area has focused exclusively on models that only take into account the class noise. Our goal is to extend existing framework to enable the analysis of system accuracy also in the context of attribute noise.

As a starting point, we focus on a model developed in [35]. As described above, the model makes a rather strong assumption that the attributes are noise-free. This implies that the attacker has apriori knowledge of the probability distribution over the space of user-item mappings (e.g., ratings matrix in a collaborative filtering context). The model, referred to as the *Biased Class Noise* (BCN) model, is characterized by two parameters: the class bias, μ, and the noise rate, β.

The BCN model assumes that an instance is generated according to the underlying distribution. In $1-\beta$ fraction of the cases, the instance is noise-free and labelled by the target concept. For the remaining portion of the instance space, the instance are labeled as positive with probability $\beta\mu$ and as negative with probability $\beta(1-\mu)$. This model is general enough to represent a variety of attack types. For example, the boundary case where $\mu = 0$, represents the situation in which an attacker is inserting fake profiles all of which negatively rate a target item (i.e., a "nuke" attack). In certain contexts (e.g., Web personalization based on usage data), only positive class labels are available. While this can be seen as a special case of the general model, the restriction may have an impact on the specific lower bounds, as well as on attack strategies.

The BCN model extends the noise-free Probably Approximately Correct (PAC) [19] model of [3] for k-nearest neighbor to handle biased class noise. The intuition behind the latter model is that kNN is accurate if the training instance space is sufficiently dense (contains a sufficient number of *good* (noise-free) instances. The notion of *density* is modeled as a function of the sample size and a distance threshold between the k instances and the rest of the instance space. Specifically, given a d-dimensional instance space X^d, and distance function, *dist* defined over $X^d \times X^d$, a subset $S \subseteq X^d$, is considered (k, α, γ)-*dense*, if except for a subset with probability less than γ, for every $x \in X^d$, there exists at least k distinct points $x_1, \ldots, x_k \in S$ such that $dist(x, x_i) \leq \alpha$, for each i.

In both [3] and [35], the assumption is that d (the number of items in instance space) is fixed (which is generally the case in instance-based learning algorithms). However, the derived lower bounds on density (or sparsity) vary with d. Thus, the model can potentially be used as the basis for situations in which d may vary as a result of the attack. Such a situation may occur in content-based or knowledge-based systems in which the attack may, indeed, involve the addition of items (such as pages, products, or links) as part of the attack profile.

In [35], the aforementioned PAC model is further extended by defining a dual notion of *sparse subsets*. The intuition behind this extension is that, in addition to containing sufficient number of noise-free instances, the sample must also *not* contain more than a specified number of *bad* (noisy) instances. The notion of (k, α, γ)-*sparse* subsets of X^d are, thus, defined analogously to that of dense subsets described above. Together these dual notions enable the extended model to handle the case were instances contain biased class noise.

Let $S_r \subset S$ denote the set of instances corresponding to real users, and $S_a = S - S_r$ denote those instances corresponding to the attack. Given the specified size of the attack β (according to the above model), the expected value of $|S_r|$ is $(1-\beta)|S|$ and the expected value of $|S_a|$ is $\beta|S|$. Similarly, let S_g denote the subset of S containing instances that are labelled correctly, and S_b, those that are labelled incorrectly. Note that S_g may contain instances from either of S_r and S_a. In particular, let p be the fraction of positive instances in S, $n = 1 - p$ the fraction of negative instances, and μ the class noise bias. Then, the probability that an arbitrary instance of S is correctly labelled is $\lambda = (1-\beta) + \beta(p\mu + n(1-\mu))$. Given this notation, the expected size of S_g is simply $(1-\lambda)|S|$ (thus the expected size of S_b is $\lambda|S|$).

For the system to make accurate recommendations, the set S_g must be $(k_1, \alpha_1, \gamma_1)$-*dense*, while the set S_b must be $(k_2, \alpha_2, \gamma_2)$-*sparse*, for specified distance and probability thresholds α_i and γ_i, respectively. In the context of instance-based learning algorithms, such as kNN, generally, $k_1 + k_2 = k$, with $k_1 > k_2$. By estimating the probabilities that each of these two conditions are satisfied, given a class C and a distribution D over X^d, [35] derive a lower bound on the probability that the error rate for the learning algorithm is less than a specified error threshold ϵ, i.e., $\Pr(\text{err}(k\text{NN}(S),C,D) < \epsilon)$.

We propose to extend the Biased Class Noise model in several ways in course of our research. Some of the alternate forms of attack that we plan to consider will violate some of the assumptions of the model. For example, a segmented attack will probably not be sparse in the subset of the product space corresponding to the genre under attack. It may be necessary to develop an analysis that partitions the product-user space, and considers the impact of the attack in different regions.

Model-based collaborative algorithms will require new analyses, as they are not dependent on the kNN algorithm. The problem of noise and bias has been well studied in the machine learning algorithms on which model-based recommenders can be based [4, 24, 7, 39]. We will need to extend these noisy learning models to handle the special cases derived from our attack models.

2.2 Stability Analysis

The second dimension along which the impact of an attack can be analyzed is that of *prediction stability*. Prediction stability can be modeled as the mean absolute variation in predictions over all items resulting from an attack. Our stability model will extend the basic framework of [23], developed in the context of database transactions, and further adapted to the context of collaborative filtering in [35].

In the context of memory-based collaborative filtering (CF) in which the task is to predict the votes of a particular user (the *active user*) from a database of user votes, the predicted vote of an active user a on item j is given by:

$$p_{a,j} = \bar{v}_a + \eta \sum_{i=1}^{n} w(a,i)(v_{i,j} - \bar{v}_i),$$

where $\bar{v}_i$ is the mean vote for user i, n is the number of users in the database with non-zero weights $w(a,i)$ and η is a normalization factor.

Following the approach introduced in [35] we model the variation in predictions resulting from an attack in terms of the *prediction error*.

For each undefined user-item pair (a,j), the prediction error is given by $E_p(a,j,T) = p'_{a,j} - p_{a,j}$, where $p_{a,j}$ and $p'_{a,j}$ are the system's predictions on pair (a,j) before and after and attack T, respectively. The prediction stability (PS) over a set A of undefined user-item pairs is then given by

$$PS(A,T,\alpha) = 1 - \frac{1}{|A|} \sum_{(a,j)\in A} \kappa_{a,j}(\alpha),$$

where α is a specified prediction shift threshold. When $\alpha \geq 0$, $\kappa_{a,j}(\alpha) = 1$ if $E_p(a,j,T) \geq \alpha$, and 0 otherwise. On the other hand, when $\alpha < 0$, $\kappa_{a,j}(\alpha) = 1$ if $E_p(a,j,T) \leq \alpha$, and 0 otherwise. Note that a stability value of 1 for any given α indicates no change in pre- and post-attack predictions.

The advantage of this general model for prediction stability is that, by varying α, a comprehensive picture of system stability can be obtained. For example, a PS value of 0.5 at $\alpha = 2$ indicates that 50% of

all predictions were shifted by at least +2 units on a given rating scale, while a PS value of 0.3 at $\alpha = -2$ indicates that 70% of all predictions were shifted by at least -2 units.

The model essentially measures the power of a recommender system to deliver stable predictions to users in the presence of an arbitrary amount of inaccurate data. As such, this model is independent of the "true" ratings for the items over which PS is calculated. This is an important feature of the model, since it allows for a direct comparison of the stability of different recommender systems without regard to accuracy considerations.

While the stability model discussed above was initially conceived in the context of collaborative filtering, we believe that the core elements of the model can be extended to accomodate a variety of other recommendation algorithms. One specific goal of this project is to extend this model for approaches based on item-based collaborative filtering, model-based techniques, such as the association-rule-based recommendations, as well as content-based and hybrid techniques. We will use the extended stability framework to comprehensively evaluate the stability of these algorithms in the face of different types of attacks.

2.3 Expected Payoff Analysis

Accuracy and stability are desirable system characteristics. Another valuable measure we would like to examine is *expected payoff*: to what extent does an attack benefit the attacker? For example, in a push attack, how does the probability that the pushed product is recommended increase as a result of the attack? We can think of payoff in game-theoretic terms. Each recommendation that is made in the attacker's favor represents some utility u. The system prior to the attack delivers some expected utility to the attacker $E(u)$, when the average user asks for a recommendation. $E(u)$ is given by

$$E(u) = \sum_{k=1}^{n} w_k P(k)$$

where k ranges over all possible recommendation outcomes, w_k is the utility associated with that outcome for the attacker, and $P(k)$ is the probability of an outcome. In a collaborative system, $P(k)$ is a function of the profile database D used to generate recommendations. Let D be the system database before an attack and D' afterwards. We are interested in $P(k)$ as conditionally dependent on the database, that is $P(k|D)$. $E(u')$ is then given by

$$E(u') = \sum_{k=1}^{n} w_k P(k|D')$$

The attacker is primarily interested in the gain brought about by the attack, that is $\delta = E(u') - E(u)$. If the attacker is only interested in a single outcome, k_i, for example, the recommendation of a single product in a push attack, then the gain is $\delta = E(u') - E(u) = w_k(P(k_i|D') - P(k_i|D))$. However, some attack profiles, such as a push/nuke combination attack imply a more complex utility computation, involving multiple recommendation outcomes.

The expected payoff measure is more specific than the general stability measure, which looks at the performance for all users and all products, and payoff captures the perspective of the attacker in a way that the other measures do not. An attack on a recommender system for which the payoff is minimal, that is with a small δ, is one that will not be worth employing.

3 Recommendation Techniques

A primary goal of this research is to explore the vulnerabilities of recommender systems to various forms of attack, both through formal modeling of the system's capabilities and possible attack strategies, and through empirical investigation of the stability and accuracy of an array of common recommendation algorithms in the face of different attack types. Specifically, we intend to study recommendation algorithms for user-based and item-based collaborative filtering; model-based approaches, such as those based on association rules and latent variable models; as well as content- or knowledge-based techniques. Furthermore, we will explore

techniques for enhancing the robustness of recommender systems, through improved algorithms, including a variety of hybrid algorithms. In this section we provide a brief overview of the specific approaches on which our study will focus.

3.1 Instance-based Collaborative Filtering

In a collaborative filtering (CF) scenario, generally we start with a list of m users $U = \{u_1, u_2, \ldots, u_m\}$, a list of n items $I = \{i_1, i_2, \ldots, i_n\}$, and a mapping between the users and items. The latter mapping is usually represented as a $m \times n$ matrix M. In the traditional CF scenario the matrix M usually represents user ratings of items, thus the entry $M_{i,j}$ represents a user u_i's rating on item i_j. In this case, the users' judgments or preferences are explicitly given by matrix M. For a given active user (also called the target user) u_k, the task of a CF system is to (1) predict $M_{k,t}$, where u_k has not rated or visited the item i_t (the *target item*) or (2) recommend a set of items that may be interesting to user u_k, those i_t with the highest predicted $M_{k,t}$.

3.1.1 User-Based Collaborative Filtering

In user-based CF algorithms [40, 26, 20], first a set of k nearest neighbors of the target user are computed. This is performed by computing correlations or similarities between user records (rows of the matrix M) and the target user. A variety of similarity or distance measures can be used in the neighbor formation task. The most common are Pearson correlation and vector-based cosine similarity measures. A primary advantage of Pearson correlation is that it uses the covariances of item ratings among users, thus taking into account possible rating styles among different users. This measure produces a correlation coefficient in the range [-1,1], with the boundary values reflecting perfect negative and perfect positive correlations among users, respectively. The cosine similarity measure is often considered a more appropriate measure when dealing with binary data (such as that based on implicit observations in Web user sessions).

Once the neighbors are obtained, a combination function (such as weighted average) is used to combine the neighbors' item ratings (or weights) to produce a prediction value for the target user on unrated (or unvisited) items.

A major problem with this approach is the lack of scalability: the complexity of the system increases linearly as a function of the number of users which, in large-scale e-commerce sites, could reach tens of millions.

3.1.2 Item-Based Collaborative Filtering

In contrast, *item-based* CF algorithms [37, 17] attempt to find k similar items that are co-rated (or visited) by different users similarly. This amounts to performing similarity computations among the columns of matrix M. For a target item, predictions can be generated by taking a weighted average of the target user's item ratings (or weights) on these neighbor items.

The first step in computing the similarity of two items i_p and i_q (column vectors in the data matrix M) is to identify all the users who have rated (or visited) both items. Many measures can be used to compute the similarity between items. In our investigation, when dealing with Web usage data, we will use the standard cosine similarity between two vectors. For ratings data, however, variances in user ratings styles must be taken into account. To offset the difference in rating scales, the data can be normalized to focus on rating variances (deviations from the mean ratings) on co-rated items. For our purposes, when dealing with ratings data, we will adapt the *Adjusted Cosine Similarity* measure introduced by Sarwar. et al. [37] which takes such variances into account. After computing the similarity between items, we select a set of K most similar items to the target item and generate a predicted value for the target item using a weighted sum.

We expect that item-based CF, which has become popular for efficiency reasons, may turn out to be particularly sensitive to focused attacks, such as the segmented and bandwagon attacks.

3.2 Model-based Collaborative Filtering

The instance-based collaborative techniques make predictions from raw user data from the profile database. Model-based collaborative recommendation extracts from the profile database a predictive model, from which recommendations are generated. We will focus on two such model-based techniques: association rules and latent semantic indexing.

3.2.1 Association-Based Approach

Association rules capture the relationships among items based on their patterns of co-occurrence across transactions. In the case of Web transactions, association rules capture relationships among pageviews based on the navigational patterns of users. Most common approaches to association discovery are based on the Apriori algorithm [1, 2] that follows a generate-and-test methodology. This algorithm finds groups of items (pageviews appearing in the preprocessed log) occurring frequently together in many transactions. Such groups of items are referred to as *frequent itemsets*.

Given a transaction T and a set $I = \{I_1, I_2, \ldots, I_k\}$ of frequent itemsets over T, the *support* of an itemset $I_i \in I$ is defined as $\sigma(I_i) = |\{t \in T : I_i \subseteq t\}|/|T|$.

An important property of support in the Apriori algorithm is its downward closure: if an itemset does not satisfy the minimum support criteria, then neither do any of its supersets. This property is essential for pruning the search space during each iteration of the Apriori algorithm. Association rules which satisfy a minimum *confidence* threshold are then generated from the frequent itemsets. An association rule r is an expression of the form $X \Rightarrow Y$ (σ_r, α_r), where X and Y are itemsets, $\sigma_r = \sigma(X \cup Y)$ is the support of $X \cup Y$ representing the probability that X and Y occur together in a transaction. The confidence for the rule r, α_r, is given by $\sigma(X \cup Y)/\sigma(X)$ and represents the conditional probability that Y occurs in a transaction given that X has occured in that transaction.

The result of association rule mining can be used in order to produce a model for recommendation or personalization systems [18, 28, 31, 38]. For this research, we adopt the framework proposed in [31], which was used in the context of personalization based on Web usage data. However, it can also easily be adopted for use in the standard collaborative fitering framework involving item-ratings. Given an active profile of size w and a group of frequent itemsets, we only consider all the frequent itemsets of size $|w| + 1$ containing the current profile. The recommendation value of each candidate is based on the confidence of the corresponding association rule whose consequent is the singleton containing the item to be recommended. The confidence of this rule is also used as the recommendation score for the recommended item.

Model-based algorithms, such as the association-based approach described above may have their own unique vulnerabilities in the face of an attack. For example, the association models are highly sensitive to global minimum support and confidence thresholds. An item that has very high support (appears in many transactions) will almost always have a high confidence in many discovered rules. This may present an opportunity for an attacker to manipulate the results based on apriori knowledge about the popularity of specific items. Liu et al. [29] proposed a mining method with multiple minimum supports that allows users to specify different support values for different items. In this method, the support of an itemset is defined as the minimum support of all items contained in the itemset. The specification of multiple minimum supports allows frequent itemsets to potentially contain rare items which are nevertheless deemed important. Such an approach may decrease the potential impact of a general random attack, but, in turn, may increase the impact on a directed push/nuke attack based on "rare" items with low minimum support threshold. We intend to study the robustness of various extensions of Apriori algorithms, such as the multiple minimum support model, in the face of different attack types.

3.3 Latent Variable Models

Latent Semantic Indexing (LSI) is a dimensionality reduction technique that has been widely used in information retrieval [16]. It has been applied in CF scenarios to reduce the dimensionality of the item space increasing density and thereby creating a space in which a given profile has more neighbors. LSI uses singular value decomposition (SVD) as its underlying matrix factorization algorithm. The reduced orthogonal

dimensions resulting from SVD are less noisy than the original data and capture the latent associations between the users and products. This reduction in noise may have benefits for robustness that our research will explore.

In our case, we perform SVD on the user-item matrix $M_{m\times n}$ by decomposing it into three matrices: $S_{m\times n} = U_{m\times r} \bullet \Sigma_{r\times r} \bullet V_{r\times n}$, where U and V are two orthogonal matrices; r is the rank of matrix S, and Σ is a diagonal matrix of size $r \times r$, where its diagonal entries contain all singular values of matrix S and are stored in decreasing order. One advantage of SVD is that it provides the best lower rank approximation of the original matrix S [5]. We can reduce the diagonal matrix Σ into a lower-rank diagonal matrix Σ_k by only keeping k ($k < r$) largest values. Accordingly we reduce U to U_k , V to V_k. Then the matrix $S_k = U_k \bullet \Sigma_k \bullet V_k$ is the rank-k approximation of the original matrix S.

To use this approach for recommendations, first the compute the square-root of the matrix Σ_k to obtain $\Sigma_k^{1/2}$ from which two resultant matrices, $U_k \bullet \Sigma_k^{1/2}$ and $\Sigma_k^{1/2} \bullet V_k'$ are computed. To generate predictions for user C_i and an item p, we compute the dot product of the ith row of $U_k \bullet \Sigma_k^{1/2}$ and the pth column of $\Sigma_k^{1/2} \bullet V_k'$ and then add back the user's average $\overline{C}$: $pred(C_i, p) = \overline{C} + U_k \bullet \Sigma_k^{1/2} \bullet [C_i]^T \bullet \Sigma_k^{1/2} \bullet V_k' \bullet [P]$.

In addition to the above SVD-based model, we will also study other variations of latent variable models that have been applied to collaborative recommender systems. Specifically, we will examine the robustness of systems based on Probabilistic Latent Semantic Analysis (PLSA) [21, 14, 22] and systems based on Principal Factor Analysis [13].

3.4 Hybrid Recommender Systems

Hybrid recommender systems combine multiple recommendation techniques with the aim of improved performance, particularly in boundary cases, such as new items unrated by other users and inaccessible to a pure collaborative approach. However, as the Google example cited above shows, hybrids also hold promise for improved robustness, since they base their recommendations on multiple data sources. The survey of hybrid recommender systems mentioned above identified seven different hybridization techniques of which six are relevant to this project [12]:[3]

Weighted The score of different recommendation techniques are combined numerically.

Switching The system chooses among recommendation techniques and applies the selected one.

Feature Combination Features derived from different knowledge sources are combined together and given to a single recommendation algorithm.

Cascade Recommenders are given strict priority, with the lower priority ones breaking ties in the scoring of the higher ones.

Feature Augmentation One recommendation technique is used to compute a feature or set of features, which is then part of the input to the next technique.

Meta-level In a meta-level hybrid, one recommendation technique is applied and produces some sort of model, which is then the input used by the next technique. This differs from feature augmentation where the first recommender contributes some features. In the meta-level hybrid, the second recommender only sees the model produced by the first recommender.

To build a hybrid system involving a collaborative recommender, there must be a second recommendation component with which to combine it. For most of the recommendation domains we will be exploring (see Evaluation section below), we have associated content data with which to build a content-based recommender. We will also explore hybrids involving a knowledge-based recommendation component where appropriate knowledge bases are available. Item-based collaborative filtering, in particular, lends itself well to a feature combination approach using semantic information [33].

Our evaluation of hybrid recommender systems will examine three questions:

[3]The seventh technique, the "Mixed" hybrid, displays the recommendations of different components side-by-side. There is no functional synergy between the techniques and hence no opportunity for increased robustness.

1. Does the addition of a content-based or knowledge-based component improve the resistance of the overall system to attack?

2. What hybridization techniques achieve the best combination of attack resistance and overall accuracy?

3. Can we quantify the "Google effect"? That is, under what circumstances does the addition of a collaborative component improve the resistance of a content-based system to a biased data (search engine spam) attack?

4 Evaluation

While formal analyses of the problem of robustness is important to our overall understanding of the problem, such analyses inevitably depend on assumptions that may not be fully satisfied in actual applications. Therefore, it is crucial that we perform empirical research as well, to investigate the vulnerabilities of recommender systems against attacks of different types. Our basic approach will be to construct recommender systems of various types in different domains of application, and examine their performance under attack.

4.1 Data Sets

We plan to conduct our research using three different types of profile data, in five different data sets. This will enable us to examine the differences between recommendation contexts and their consequences for secure recommendation.

Ratings-Oriented Data

We will use two data sets for evaluating algorithms in the traditional collaborative context of ratings data. *EachMovie* data set consists of 2.8 million explicit numeric ratings from approximately 73,000 users on 1628 movies.[4]. A similar data set is the *MovieLens* data set containing 100,000 explicit numeric ratings on 1682 movies from 943 users. Each user has rated 20 or more movies with a rating scale of 1 to 5.

Content information associated with these movies will be extracted using our own wrapper agent that extracts movie instances from the Internet Movie Database [5] based on a simple movie ontology. Specifically, each movie instance is populated with semantic attributes, including movie title, release year, director(s), cast, genre, and plot. The availability of semantic information for movies, in addition to ratings information, will allow us to conduct experiments on hybrid algorithms using a content-based recommendation component.

Web Usage Data

For our experiments with Web usage data involving implicit ratings and user profiles, we will use data set derived from the Web server logs of DePaul University's School of Computer Science, Telecommunication, and Information systems. After data preprocessing, we have identified 21,299 user sessions and 692 Web pageviews, where each user session consists of at least 6 pageviews. In this data set we use the time spent on each pageview as the weight associated with that pageview in the given user session. Since the pages on this site are heterogenous, content information consists only of the terms that appear on each page.

Hybrid Semantic-Usage Data

More semantic information for the integration of Web usage mining with semantic information is found in the real estate data set, based on the raw Web usage data from the server logs of a local affiliate of a national real estate company. The portion of the Web usage data during the period of analysis contains approximately 24,000 user sessions from 3800 unique users. In the processed data matrix, each row represents a user vector with properties as dimensions and visit frequencies as the corresponding dimension values.

[4]Dataset available from http://research.compaq.com/SRC/eachmovie/

[5]www.imdb.com

To automatically extract semantic information about the properties, we use a reference ontology for the class "property". Using a wrapper agent, the attribute values for each property instance are extracted directly from pages related to that property on the Web site. The extracted attributes include price, number of rooms, number of bathrooms, garage size (cars), style, exterior, lot size, school district, zip code, etc.

Critique-Based Profile Data

Our experiments on critique-oriented profiles and knowledge-based recommender systems will be based on the *Entree Chicago* recommender system. This data records the interactions of users with the Entree Chicago system, and contains interactive sessions evaluating over 1,000 restaurants generated by approximately 50,000 users.[6] The sessions are in the form of interactive dialogs, in which the system proposes restaurants and the user responds with semantic critiques.

4.2 Methodology

We will conduct our experiments to mirror attack conditions. The profiles will be partitioned into training data and test data using 10-fold cross-validation. We will evaluate the accuracy of the system prior to the attack, by training the recommendation system with the training data and testing its accuracy against the test data. Then, we will generate biased profiles based on an attack scenario and repeat the evaluation.

A general approach for the evaluation of accuracy in this context would be to divide the training set into a real set R and an attack set A. A portion of the attack set is then modified according to the data distribution associated with the attack profile models which will be developed as part of this research. For example, in the case of the class noise model discussed in Section 2.1, a *noisy* set N is drawn from A containing $\beta|R|$ instances. In this case, the class attributes instances in N is set to positive with a probability μ and to negative with probability $1 - \mu$. The algorithms are then trained on $R \cup N$, and the accuracy of the model is measured using standard approaches such as Mean Absolute Error or Precision and Recall. A similar approach can be adopted for cases where various types of attribute noise are also part of the attack profile.

In the case of stability analysis, we are interested in measuring changes in predictions or recommendations before and after the attack. Thus the standard measures of prediction accuracy such as MAE will not be sufficient. One possible approach would be to adopt a *leave-one-out* testing methodology. In this case, a single user-item pair is removed from the data set for which a prediction is generated using the trained model. The prediction for this pair is then compared before and after the attack and the Prediction Stability measure PS, described in Section 2.2, is computed for different values of the α threshold.

5 Detection and Response

O'Mahoney and colleagues demonstrate that attacks against recommender systems can be successful and the Amazon and Google examples demonstrate that such attacks are real-world problems. We do not expect to find recommendation algorithms that are completely invulnerable to attack. Therefore, in addition to good algorithms, e-commerce sites using recommender systems will need to detect and respond to attacks.

Detection of recommender system attacks is difficult. Collaborative systems are supposed to be influenced by their users, and are supposed to change their recommendations in response to new input. Is a sudden jump in frequency of recommendation of Eve's book Z a function of it being a good read with lots of admirers, or are these admirers really bogus profiles inserted by Eve? To tell these two situations apart will require examination of the attack pattern itself, and will be a function of the attack type.

For example, consider a segmented push attack. The attack will consist of many profiles, all of which contain a positive rating for Z and also positive ratings for books in related genres. These profiles will cluster together in a corner of the recommendation space, weighting it towards Z. There may be genuine users whose profiles show such single-minded concentration on a single topic, but there is unlikely to be many such users. The problem of attack identification will therefore be one of generating metrics that can distinguish between the biased cluster and other clusters of genuine users. Our research will examine each of the attack profiles

[6]Dataset available from the UCI KDD archive: http://kdd.ics.uci.edu/databases/entree/entree.html

and consider how these attacks would manifest themselves in detectable ways. The concept of payoff may be useful to us here. It is not unlikely that a larger number of genuine profiles added to a recommender system will affect its stability and its accuracy, when measured by the previous body of users. However, it is very unlikely that a large number of genuine profiles would all affect the payoff for a single item in exactly the same way. These profiles may be considered suspicious. We will develop and evaluate such heuristics for attack detection.

Responding to an attack may not simply be a matter of removing the offending profiles. It may be impossible to distinguish between genuine and bogus profiles on an individual basis, even if a cluster of profiles can be determined to be biased. We will consider how a recommender system may compensate for bias once it is detected, and evaluate these compensation mechanisms in the context of different attacks.

6 Impact of the Proposed Activity

6.1 Broader Impact

The scope and rapid expansion of the universe of Internet-accessible information and the penetration of the World Wide Web into all areas of human activity have made the problem of information access one of universal importance. Tasks from scientific discovery to national security all require extracting useful needles from information haystacks. One increasingly-used technique for lifting the burden of information access from users is automated recommendation. Users of e-commerce sites in particular have come to rely on systems that can independently filter information and arrive at useful recommendations. With the increased use of such systems comes the very real risk that the information base on which they are based, namely user profiles, can be distorted by malicious users.

Confronting this risk can lead to substantial gains in performance and utility. One of the key advantages of the highly-successful Google search engine has been its robustness in the face of so-called "search engine spam," against which many of its competitors were relatively helpless. The broad benefit of this research will therefore be to enhance our understanding of the risks and threats associated with recommender systems, and to examine how those risks can be mitigated. This will be of benefit to all users of advanced information access systems, as well as the implementers of recommender systems.

The results of our research will also have significant implications for a variety of adaptive information systems that rely on users' input for learning user or group profiles. Such systems include adaptive hypertext systems, personalized e-learning environments, adaptive information filtering systems (e.g., email spam filtering applications), and collaborative group-based environments. Many such systems have open components through which a malicious user or an automated agent can affect the overall system behavior.

Results of the research will be published in appropriate journals, and presented at relevant conferences, including major AI conferences such as IJCAI and AAAI; conferences related to electronic commerce, such as the ACM and IEEE E-Commerce conferences; and conferences with primary focus on information assurance and security, such as the ACM Conference on Computer and Communications Security.

6.2 Educational Impact

The School of Computer Science, Telecommunications and Information Systems (CTI) at DePaul University is the largest school of Information Technology in the United States. CTI offers eleven graduate and seven undergraduate programs, making it the school with the most diverse curriculum of any other graduate program in a computer-science-related discipline. CTI offers a master's degree in Computer, Information and Network Security certified by the National Committee on Security Standards, and has begun integrating security topics into many of its undergraduate courses. This research will contribute to CTI's growing strength in the security area. The investigators on this grant are full-time teaching faculty and will remain so during the grant period – no release time from teaching responsibilities is requested as part of this proposal.

Dr. Burke teaches CTI's course in electronic commerce security, and recommender systems provide a prime example of a subtle application vulnerability. Examples from this research will be used to supplement the teaching of this course. Dr. Mobasher is responsible for CTI's graduate courses in information retrieval

and Web mining, the teaching of which will be informed greatly by the conduct of this project. These courses (as well as some others) will draw upon many facets of our proposed work for design and development projects. Furthermore, many students must complete a research or implementation project for various CTI programs as part of a capstone course. We hope that the proposed research activities will satisfy some of this demand and, at the same time, ensure the future success of the project as a whole.

We will actively seek participation of undergraduate students to the research project by encouraging them to apply for the undergraduate research opportunity grants available from the university. Currently, we are seeking undergraduate research support from a variety of sources, including from the local industry, as well as from NSF through a multi-year "Research Experiences for Undergraduates" grant. We are also working with programs for minority students to attempt to recruit qualified students to work with us during the summer. DePaul University, as a whole, has been extremely successful in the recruitment and retention of minority students and women, resulting in one of the most diverse student populations in higher education. We hope to promote undergraduate interest, especially among underrepresented groups, in attending graduate schools and in pursuing research and teaching careers.

In addition, this project will provide tuition and stipend support and invaluable experience for two graduate students in CTI's PhD program, and funding for their travel expenses to conferences to present their work and gain exposure to the wider computer science community.

7 Plan of Work

Year 1

- We will develop formal models for the analysis of robustness in recommender systems. The models will include frameworks for analyzing recommendation accuracy, prediction stability and expected payoff. We will apply these models to simple attacks such as push, nuke and random attacks.
- We will implement our framework for evaluation, including the approriate evaluation methodologies for the analysis of stability, accuracy and payoff based on the formal models.
- We will build our user-based, item-based and model-based collaborative recommendation algorithms, and conduct experiments examining the robustness of these algorithms, in terms of accuracy, stability and payoff, using our data sets.

Year 2

- We will continue to develop our formal models, addressing more complex attacks including combined push/nuke attacks, segmented attacks and bandwagon attacks. We will also examine attacks against systems that use implicit ratings, extending our models as necessary.
- We will examine the impact of different attacks on recommender systems that use latent variable models including singular-value decomposition, probabilistic latent semantic analysis, and principle factor analysis.
- We will develop content-based and (where possible) knowledge-based recommenders for our data sources, and conduct additional experiments on hybrid recommendation algorithms involving these recommenders and the collaborative ones explored in Year 1.

Year 3

- We will examine enhancements to our original recommendation algorithms geared towards increased robustness and determine possible trade-offs between accuracy and robustness.
- We will develop a systematic framework for characterizing different attack profiles and attack strategies with different underlying recommendation algorithms. Based on this framework, we will develop and evaluate techniques for detecting attacks.
- We will develop and evaluate response strategies to various attack types for different combinations of attack profiles and recommendation algorithms.

8 Results of Prior NSF Support

Collaborative Research: A Computational Framework for Agent-based Contracting (IIS-0084234); Award amount: $133897; PI: Bamshad Mobasher; Award duration: August 31, 2000 - August 31, 2004.

The overall goal of this project has been to investigate theoretical foundations and computational methods to support automated negotiation of contracts for tasks that have scheduling constraints in the context of a multi-agent framework, called MAGNET. The project is a collaborative effort with Maria Gini at the University of Minnesota (with separate NSF funding for the Minnesota team).

The results of this research have contributed to the development of general computational models for multi-agent contracting. In particular, we have investiagted and developed several techniques for bid evaluation with time constraints in the context of a multi-agent contracting environment involving self-interested supplier and customer agents. In particular, we have implemented the following bid-evaluation search engines: a highly-modular simulated annealing version, a Mixed Integer Programming version, and a IDA* versions to deal with reverse-auction problems having precedence constraints among items.

A major focus of the project has been the development and release of the MAGNET testbed and a simulation environment for agent-based contracting, as presented in the original proposal. The testbed software has been fully implemented and the final release includes a complete API and associated open source software for the customer agent part of MAGNET together with a basic framework for the market and a simple supplier agent for simulation purposes. The testbed supports a number of measurements for evaluating search performance, including search effort, anytime performance, and solution quality, along with counts of solved, unsolved, and known unsolvable problems encountered.

The work on MAGNET is currently ongoing. The DePaul team has developed a prototype for a MAGNET server which provides support for supplier-customer interactions within the MAGNET framework. The current version of the server is implemented in Enterprise JavaBeans, and we use Apache SOAP for communication between agents and the server. Currenlty we are focused on developing another version of the MAGNET server based on the Web Services model using WSDL. We hope to release this version of the server as an extension of the original release of the testbed.

In addition to research and development activities, the MAGNET project has resulted in numerous contributions on the educational front. In particular, various aspects of the MAGNET framework have been used as an integrated part of several graduate courses at DePaul University, including courses in the Electronic Commerce technology and the Distributed Systems programs. These courses include a new course on Agent-Based Markets for Electronic Commerce. Both at DePaul and at Minnesota, the MAGNET project has contributed to the educational and research activities of more than a dozen Masters and PhD students, as well as several undergraduate students. The funding has also supported several minority students both at the PhD level (DePaul) and the Masters level (Minnesota).

The following are selected publications related to the MAGNET project:

- A. Babanov, J. Collins, and M. Gini. "Scheduling tasks with precedence constraints to solicit desirable bid combinations." In *Proc. of the Second Int'l Conf. on Autonomous Agents and Multi-Agent Systems*, Melbourne, Australia, July 2003
- J. Collins, W. Ketter, M. Gini, B. Mobasher, "A multi-agent negotiation testbed for contracting tasks with temporal and precedence constraints", *Int'l Journal of Electronic Commerce*, 7(1):35–57, 2002.
- S. Botman, M. Hoogendoorn, V. Bud, A. Jaiswal, S. Hawkins, Y. Kryzhnyaya, J. Pearce, A. Schoolcraft, E. Sigvartsen, J. Collins, and M. Gini. "Design of supplier agents for an auction-based market." In *AAMAS-02 Workshop on Agent-Oriented Information Systems*, July 2002.
- John Collins, Cory Bilot, Maria Gini, and Bamshad Mobasher, "Decision Processes in Agent-Based Automated Contracting", IEEE Internet Computing, p. 61, vol. 5 (2), 2001.
- J. Collins and M. Gini. "A testbed for multi-agent automated contracting." In *IJCAI-2001 Workshop on Artificial Intelligence and Manufacturing*, August 2001.
- J. Collins, C. Bilot, M. Gini, and B. Mobasher. "Mixed-initiative decision support in agent-based automated contracting." In *Proc. of the Fourth Int'l Conf. on Autonomous Agents*, June 2000.

References

[1] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *Proceedings of the 20th International Conference on Very Large Data Bases (VLDB'94)*, Santiago, Chile, September 1994.

[2] R. Agrawal and R. Srikant. Mining sequential patterns. In *Proceedings of the International Conference on Data Engineering (ICDE'95)*, Taipei, Taiwan, March 1995.

[3] M. Albert and D. Aha. Analyses of instance-based learning algorithms. In *Proc. 9th Nat. Conf. Artificial Intelligence*. AAAI, Morgan Kaufmann, 1991.

[4] D. Angluin and P. Laird. Learning from noisy examples. *Machine Learning*, 2(4):343–370, 1988.

[5] M.W. Berry, S.T. Dumais, and G.W. O'Brien. Using linear algebra for intelligent information retrieval. *SIAM Review*, 37:573–595, 1995.

[6] D. Billsus and M. Pazzani. User modeling for adaptive news access. *User-Modeling and User-Adapted Interaction*, 10(2-3):147–180, 2000.

[7] D. Billsus and M.J. Pazzani. Learning collaborative information filters. In *Proceedings of the International Conference on Machine Learning*, Madison, WI, 1998.

[8] J. S. Breese, D. Heckerman C., and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the 14th Annual Conference on Uncertainty in Artificial Intelligence*, pages 43–52, 1998.

[9] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117, 1998.

[10] R. Burke. Hybrid recommender systems: Comparative studies. in preparation.

[11] R. Burke. Knowledge-based recommender systems. In A. Kent, editor, *Encyclopedia of Library and Information Systems*, volume 69. Marcel Dekker, New York, 2000.

[12] R. Burke. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4):331–370, 2002.

[13] J. Canny. Collaborative filtering with privacy. In *Proceedings of the IEEE Symposium on Security and Privacy*, Berkeley, CA, May 2002.

[14] D. Cohn and T. Hofmann. The missing link: A probabilistic model of document content and hypertext connectivity. In Thomas G. Dietterich Todd K. Leen and Volker Tresp, editors, *Advances in Neural Information Processing Systems 13*. MIT Press, 2001.

[15] R. Cooley, B. Mobasher, and J. Srivastava. Data preparation for mining world wide web browsing patterns. *Journal of Knowledge and Information Systems*, 1(1), 1999.

[16] S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, and R. Hashman. Indexing by latent semantic indexing. *Journal of the American Society for Information Science*, 41(6), 1990.

[17] M. Deshpande and G. Karypis. Item-based top-n recommendation algorithms. *ACM Transactions on Information Systems*, 22(1):1–34, 2004.

[18] X. Fu, J. Budzik, and K. J. Hammond. Mining navigation history for recommendation. In *Proceedings of the 2000 International Conference on Intelligent User Interfaces*, New Orleans, LA, January 2000. ACM Press.

[19] D. Haussler. Probably approximately correct learning. In *Proceedings of the 8th National Conf. on Artificial Intelligence*, pages 1101–1108. AAAI, Morgan Kaufmann, 1990.

[20] J. Herlocker, J. Konstan, A. Borchers, and J. Riedl. An algorithmic framework for performing collaborative filtering. In *Proceedings of the 22nd ACM Conference on Research and Development in Information Retrieval (SIGIR'99)*, Berkeley, CA, August 1999.

[21] T. Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 42(1):177–196, 2001.

[22] T. Hofmann. Collaborative filtering via gaussian probabilistic latent semantic analysis. In *Proceedings of the 26th International Conference on Research and Development in Information Retrieval (SIGIR03)*, Totonoto, Canada, July 2003.

[23] C.N. Hsu and C.A. Knoblock. Estimating the robustness of discovered knowledge. In *Proceedings of the 1st Int'l Conference on Knowledge Discovery and Data Mining*, Montral, Canada, 1995.

[24] M. Kearns and M. Li. Learning in the presence of malicious errors. In *Proceedings of the 20th ACM Symposium on Theory of Computing*, pages 267–280, 1988.

[25] D. Kelly and J. Teevan. Implicit feedback for inferring user preference: A bibliography. *ACM SIGIR Forum*, 37(2):–, Fall 2003.

[26] J. Konstan, B. Miller, D. Maltz, J. Herlocker, L. Gordon, and J. Riedl. Grouplens: Applying collaborative filtering to usenet news. *Communications of the ACM*, 40(3), 1997.

[27] K. Lang. Newsweeder: Learning to filter news. In *Proceedings of the 12th International Conference on Machine Learning*, pages 331–339, 1995.

[28] W. Lin, S. A. Alvarez, and C. Ruiz. Efficient adaptive-support association rule mining for recommender systems. *Data Mining and Knowledge Discovery*, 6:83–105, 2002.

[29] B. Liu, W. Hsu, and Y. Ma. Association rules with multiple minimum supports. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'99, poster)*, San Diego, CA, August 1999.

[30] B. Mobasher, R. Cooley, and J. Srivastava. Automatic personalization based on web usage mining. *Communications of the ACM*, 43(8):142–151, 2000.

[31] B. Mobasher, H. Dai, T. Luo, and M. Nakagawa. Effective personalization based on association rule discovery from web usage data. In *Proceedings of the 3rd ACM Workshop on Web Information and Data Management (WIDM01)*, Atlanta, Georgia, November 2001.

[32] B. Mobasher, H. Dai, T. Luo, and M. Nakagawa. Discovery and evaluation of aggregate usage profiles for web personalization. *Data Mining and Knowledge Discovery*, 6:61–82, 2002.

[33] B. Mobasher, X. Jin, and Y. Zhou. Semantically enhanced collaborative filtering on the web. In *Proceedings of First European Web Mining Forum*, Lecture Notes in Artificial Intelligence. Springer, To appear in 2004.

[34] R. J. Mooney and L. Roy. Content-based book recommending using learning for text categorization. In *SIGIR '99 Workshop on Recommender Systems: Algorithms and Evaluation*, Berkeley, CA, 1999. ACM SIGIR.

[35] M. O'Mahony, N. Hurley, N. Kushmerick, and . Silvestre. Collaborative recommendation: A robustness analysis. *ACM Transactions on Internet Technology*, in press.

[36] Associated Press. Computer glitch identifies anonymous amazon reviewers. eWeek, Feb. 14 2004.

[37] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International WWW Conference*, Hong Kong, May 2001.

[38] B. M. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Analysis of recommender algorithms for e-commerce. In *Proceedings of the 2nd ACM E-Commerce Conference (EC'00)*, Minneapolis, MN, October 2000.

[39] R.A. Servedio. Smooth boosting and learning with malicious noise. *Journal of Machine Learning Research*, 4:633–648, 2003.

[40] U. Shardanand and P. Maes. Social information filtering: Algorithms for automating 'word of mouth'. In *Proceedings of the Computer-Human Interaction Conference (CHI95)*, Denver, CO, May 1995.

[41] J. Srivastava, R. Cooley, M. Deshpande, and P. Tan. Web usage mining: Discovery and applications of usage patterns from web data. *SIGKDD Explorations*, 1(2):12–23, 2000.