A random walks perspective on maximizing satisfaction and profit

Matthew Brand[*]

Abstract

We model consumer behavior such as web browsing, shopping, and entertainment choices as random walks on a weighted association graph. The graph is derived from a relational database that links products, consumers, and attributes such as product categories, consumer demographics, market segments, etc. The Markov chain that describes this walk amalgamates consumer behavior over the whole population; individuals are distinguished by their current state in the chain. We develop a geometrization of the chain that furnishes a key similarity measure for information retrieval—the cosine (correlation) angle between two states. Empirically, this proves to be highly predictive of future choices made by individuals, and is useful for recommending and semi-supervised classification. This statistic is obtained through a sparse matrix inversion, and we develop approximation strategies that make this practical for very large Markov chains. These methods also make it practical to compute recommendations to maximize long-term profit.

Keywords: collaborative filtering; random walks; Markov chain; cosine correlations; semi-supervised classification.

1 Introduction

Collaborative filtering seeks to make recommendations to individuals based on the choices made by a population. An extensive literature treats this as a missing value problem, wherein an individual's history of choices is a fragment of a hypothetical vector containing that individual's ratings or rankings of all possible consumables. The goal is to fill in that vector (imputation) or identify the relative ranking of the of the unknown elements. A rich literature has grown up around this problem, with successful demonstrations of Bayesian, nonparametric, and even linear methods; see [1] for a broad survey. All methods essentially match the individual to others who have made similar choices, and use some combination of their experiences to predict future choices.

In this paper we explore the idea of making recommendations on the basis of associations in a relational database. The database may connect categories to products to purchasers to demographics, etc., and we may be interested in finding out what products a customer is likely to buy next, what product categories are preferred by specific demographic groups, or even how to sequence sales pitches to maximize likely profits. We answer these questions by looking at the expected behavior of a random walk on the database's association graph (e.g., see figure 1). The expected travel time between states gives us a distance metric that has a natural transformation into a similarity measure. The random walks view has been highly successful in social networks analysis (e.g., [10, 13]) and web search (e.g., [4, 11, 3]) and in many respects is formally identical to the analysis of electrical resistive networks [5]. We develop a novel measure of similarity based on random walk behavior—the *cosine correlation between states*—and show that it is much more predictive of individual's future choices than classic graph-based dissimilarity measures. A particularly nice feature of the random walks view is that it can naturally incorporate large amounts of contextual information beyond the usual who-liked-what of collaborative filtering, including categorical information. All this comes at a heavy computational price, but we outline approximation strategies that make these computations practical for very large graphs. These also make it practical to compute a classically useful statistic—the expected discounted profit of states, and make recommendations that optimize vendor profit.

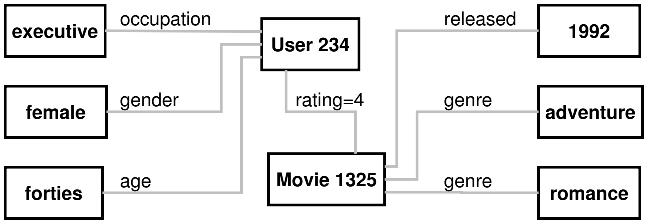

Figure 1: A fragment of an association graph representing a relational database. Affinities between pairs or groups of vertices can be computed from statistics of a random walk on the entire graph.

2 Markov chain statistics

Let $\mathbf{W} \in \mathbb{R}^{N\times N}$ be a sparse nonnegative matrix that specifies the edges of a graph. $\mathbf{W}$ may count events, e.g., W_{ij} is the number of times event j followed event i, or more generally,

[*] Mitsubishi Electric Research Labs, Cambridge MA 02139 USA

we may view $\mathbf{W}$ as an arbitrarily weighted association matrix with $W_{ij} > 0$ iff person i has viewed movie j, or if web page i contains keyword j, etc. We are interested in a random walk on the directed graph specified by $\mathbf{W}$. (If $\mathbf{W}$ is symmetric the graph is undirected.) The row-normalized stochastic matrix $\mathbf{T} = \mathrm{diag}(\mathbf{W1})^{-1}\mathbf{W}$ contains the transition probabilities of the associated Markov chain ($\mathbf{1}$ is a vector of 1's). We assume that the chain is irreducible and has no unreachable or absorbing states; it may be asymmetric and self-transitions are allowed to model repeat purchases. If the statistics in $\mathbf{W}$ are a fair sample of the collective behavior of a population, then a random walk on this Markov chain will mimic, over the short term, the behavior of individuals randomly drawn from this population.

Various statistics of this walk are useful for prediction tasks. The *stationary distribution* $\mathbf{s} \in \mathbb{R}^N$ describes the relative frequencies of visiting each state in an infinitely long random walk, and can be used to flag the most popular products. Formally, $\mathbf{s}^\top \doteq \mathbf{1}^\top \mathbf{T}^\infty$ satisfies $\mathbf{s}^\top = \mathbf{s}^\top \mathbf{T}$ and $\mathbf{s}^\top \mathbf{1} = 1$. If $\mathbf{W}$ is symmetric then $\mathbf{s} = \frac{\mathbf{1}^\top \mathbf{W}}{\mathbf{1}^\top \mathbf{W1}}$; otherwise it may be computed from the recurrence $\mathbf{s}_{i+1}^\top \leftarrow \mathbf{s}_i^\top \mathbf{T}$, $\mathbf{s}_0 = \mathbf{1}/N$. The *recurrence times* $\mathbf{r} \in \mathbb{R}^N : r_i = s_i^{-1}$ describes the expected time between two visits to the same state (and should not be confused with the self-commute time $C_{ii} = 0$ described below). The *expected hitting time* H_{ij} for a random walk starting at state i to hit state j can be computed from

$$\mathbf{A} \doteq (\mathbf{I} - \mathbf{T} - \mathbf{1f}^\top)^{-1} \tag{2.1}$$

for any vector $\mathbf{f} > \mathbf{0}$ satisfying $\mathbf{f}^\top \mathbf{s} \neq 0$ as

$$H_{ij} = (A_{jj} - A_{ij})/s_j \tag{2.2}$$

and the expected round-trip *commute time* is

$$C_{ij} = C_{ji} = H_{ij} + H_{ji}. \tag{2.3}$$

For the special case of $\mathbf{f} = \mathbf{s}$, $\mathbf{A}$ is the inverse of the *fundamental matrix* and we recover the classic formula for hitting times [2]. The two dissimilarity measures C_{ij} and H_{ij} have been proposed as a basis for making recommendations [6] but they can be dominated by the stationary distribution, often causing the same popular items to be recommended to every consumer, regardless of individual consumer tastes. Adhoc normalizations have been proposed, but none are clearly advantageous. In this regard, it will prove useful to develop an understanding of how the chain embeds in normed spaces.

2.1 Random walk correlations

Here we establish a connection to one of most useful statistics of information retrieval: the *cosine correlation*. In information retrieval, items are often represented by vectors that count various attributes. For example, if we view a document as a sample from a process that generates a particular distribution of words, its attribute vector counts (or log-counts) how many times each (stemmed) word appears. Similar documents employ similar vocabulary, thus the inner product of their attribute vectors is large. However, longer documents sample this distribution more, resulting in more words and larger inner products. In order to be invariant to this sampling artifact, one normalizes the vectors, so that the inner product measures the empirical correlation between any two word distributions. This measure is called the cosine correlation because the normalized inner product is the cosine of the angle between two vectors.

To extend this idea to random walks, we will take two states to be similar if their relations to all other states are similar, just as similar documents have similar relationships to words.

The key idea for formalizing this intuition is a geometrization of the chain's long-term behavior: The square-root commute times are metric, satisfying the triangle inequality $\sqrt{C_{ij}} + \sqrt{C_{jk}} \geq \sqrt{C_{ik}}$, symmetry $\sqrt{C_{ij}} = \sqrt{C_{ji}}$, and identity $\sqrt{C_{ii}} = 0$ [7]. Identifying commute times with squared distances $C_{ij} \sim \|\mathbf{x}_i - \mathbf{x}_j\|^2$ sets the stage for a geometric embedding of a Markov chain in a Euclidean space[1], with each state assigned to a point $\mathbf{x}_i \in \mathbb{R}^N$, and similar states located near to each other. Because raw commute times reflect the stationary distribution, popular states will crowd near the origin regardless of dissimilarity, so raw Euclidean distance is unsuitable for most applications. However, the angle $\theta_{ij} \doteq \angle(\mathbf{x}_i, \mathbf{x}_j)$ between the embedding vectors $\mathbf{x}_i, \mathbf{x}_j$ of states i and j factors out the centrality of popular states. More importantly, its cosine measures the correlation between these two state's travel times to the rest of the graph—how similar their roles are in a random walk. E.g., if two states are perfectly correlated ($\cos\theta_{ij} = 1$), then jumping instantaneously from one to the other would not change the statistics of the random walk over the remaining states.

We need not actually compute the embedding to obtain the cosines. We can convert the matrix of squared distances $\mathbf{C}$ to a matrix of inner products $\mathbf{P}$ by observing that

$$\begin{aligned} C_{ij} &= \|\mathbf{x}_i - \mathbf{x}_j\|^2 & (2.4)\\ &= \mathbf{x}_i^\top \mathbf{x}_i - \mathbf{x}_i^\top \mathbf{x}_j - \mathbf{x}_j^\top \mathbf{x}_i + \mathbf{x}_j^\top \mathbf{x}_j & (2.5) \end{aligned}$$

[1] Both $\sqrt{C_{ij}}$ and C_{ij} are metrics; why prefer $\sqrt{C_{ij}}$? Consider square lattice graphs of varying dimension and uniform transition probabilities. The identification $C_{ij} \sim \|\mathbf{x}_i - \mathbf{x}_j\|$ leads to embeddings of 1D lattice graphs with uniform distances between adjoining states, but higher dimensional lattices are embedded with a pin-cushion radial distortion (corners are pulled away from the origin). Concentrating the graph in the corner spikes makes near-corner vertices have larger distances but smaller angles to central vertices than other non-corner vertices—undesirable because they are not similar to central vertices. The identification $C_{ij} \sim \|\mathbf{x}_i - \mathbf{x}_j\|^2$ leads to embeddings with a lattice-axis-parallel barrelling distortion (straight lines in the lattice are preserved, but the spacing of lattice lines is compressed according to the sigmoid $x \to \sin x$ on $(-\pi, \pi)$; angles properly increase with distance in the graph. Proof: Embedding uses the (nonconstant) eigenvectors of the graph Laplacian which comprise the lowest frequencies of a Fourier basis on the domain of grid spacings.

$$= P_{ii} - P_{ij} - P_{ji} + P_{jj}\,. \tag{2.6}$$

Thus, removing the row- and column-averages $P_{ii} = \mathbf{x}_i^\top \mathbf{x}_i$ and $P_{jj} = \mathbf{x}_j^\top \mathbf{x}_j$ from $\mathbf{C}$ by a double-centering

$$-2 \cdot \mathbf{P} = (\mathbf{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^\top)\mathbf{C}(\mathbf{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^\top) \tag{2.7}$$

yields $P_{ij} = \mathbf{x}_i^\top \mathbf{x}_j$ [12]. The cosine correlation is then

$$\cos\theta_{ij} = \frac{\mathbf{x}_i^\top \mathbf{x}_j}{\|\mathbf{x}_i\| \cdot \|\mathbf{x}_j\|} = \frac{\mathbf{x}_i^\top \mathbf{x}_j}{\sqrt{\mathbf{x}_i^\top \mathbf{x}_i} \cdot \sqrt{\mathbf{x}_j^\top \mathbf{x}_j}} = \frac{P_{ij}}{\sqrt{P_{ii}P_{jj}}}. \tag{2.8}$$

In appendix A we will show efficient ways to compute $\mathbf{P}$ directly from sparse $\mathbf{T}$ or $\mathbf{W}$ without computing dense $\mathbf{C}$. One result established there is that for the special case of symmetric, zero-diagonal $\mathbf{W}$, $\mathbf{P}$ simplifies to the pseudo-inverse of the graph Laplacian $\mathrm{diag}(\mathbf{W1}) - \mathbf{W}$.

For an alternate geometric interpretation of the cosine correlations, consider projecting all embedded points onto a unit hypersphere (thereby removing the effect of generic popularity) and denoting the resulting pairwise Euclidean distances as $\overset{\circ}{d}_{ij}$. Then

$$\cos\theta_{ij} = 1 - (\overset{\circ}{d}_{ij})^2/2. \tag{2.9}$$

In this embedding the correlation between two states is negatively proportional to their squared Euclidean distance. Thus summing and averaging correlations is a geometrically meaningful way to measure similarity between two *groups* of states.

In large chains the norm $\|\mathbf{x}_i\| = \sqrt{P_{ii}}$ is usually a close approximation (up to a constant factor) of the recurrence time $r_i = s_i^{-1}$, roughly the inverse "popularity" of a state, so the cosine correlations may be interpreted as a measure of similarity that factors out artifacts of uneven sampling. For example, if two web pages are very popular the expected time to hit either from any page will be low, thus they will have a small mutual commute time. But if they are usually accessed by different groups of people or are connected to different sets of attributes, the angle between them may be large, implying decorrelation or anticorrelation. Similarly, with a movie database described below we find that the horror thriller "Silence of the Lambs" to the children's film "Free Willy" have a smaller than average mutual commute time because both were box-office successes, yet the angle between them is larger than average because there was little overlap in their audiences.

As presented, these calculations require the construction and inversion of a dense $N \times N$ matrix, an $O(N^3)$ proposition that is clearly impractical for large chains. It is also wasteful because most queries will involve submatrices of $\mathbf{P}$ and the cosine matrix. Section A will show how to efficiently estimate the submatrices directly from the sparse Markov chain parameters.

3 Recommending as semi-supervised classification

To make recommendations, we select one or more query states and then rank other states by their summed (or averaged) correlation to the query states. The query states may represent customers, recent purchases, demographic categories, etc.

Recommending in this model is strongly related to the semi-supervised classification problem: The states are embedded in a space as points, one or more points are given class labels, and we seek to compute an affinity (similarity measure) between each unlabelled point and each class. Unlike fully supervised classification, the affinity between a point and the labelled examples is mediated by the distribution of other unlabelled points in the space, because they influence the (locally varying) distance metric over the entire space. Similarly, in a random walk on a graph, the similarity between two states depends on the distribution of all possible paths in the graph.

To make this visually intuitive, we revisit a classification problem recently proposed by Zhou & Schölkopf [14] in the machine learning literature (see figure 2): 80 points are arranged in two normally distributed clusters in the 2D plane, surrounded by an arc of 20 points. An undirected graph is made by connecting every point to its k nearest neighbors (figure 2 left), giving a sparse graph, or to all neighbors within some distance ε (figure 2 right), giving a denser graph. Edge weights are chosen to be a fast-decaying function of Euclidean distance, e.g., $W_{ij} \propto \exp(-d_{ij}^2/2)$. Although connectivity and edge weights are loosely related to Euclidean distance, similarity is mediated entirely by the graph, not its layout on the page. Given three labelled points (one on the arc and one on each cluster) representing two classes, Zhou & Schölkopf ask how the rest should be classified, and propose the similarity measure $((1-\alpha)\mathbf{I} + \alpha\mathbf{N})^{-1}$, with $\mathbf{N} = \mathbf{I} - \mathrm{diag}(\mathbf{W1})^{-1/2}\mathbf{W}\mathrm{diag}(\mathbf{W1})^{-1/2}$ the normalized combinatorial Laplacian, and $0 < \alpha < 1$ a user-specified regularization parameter. This is similar to our framework in the special case of an undirected graph with no self-arcs, but whereas we normalize the pseudo-inverted Laplacian to obtain cosines, they normalize the Laplacian, then regularize to make ordinary inversion feasible[2].

The similarity measure should be relatively insensitive to perturbations of the graph, especially those inflicted by a user varying the graph parameter k or ε. Since these mainly affect the density of edges and thus the stationary distribution, we may expect some classification robustness from cosine correlations. Figure 2 shows two such labellings. Clas-

[2] Zhou & Schölkopf suggest their measure is the cosine associated with commute time norms on a "lazy" random walk, but equations 3.4, 3.8 and 3.9 in their analysis only hold for $\alpha = 1$ (where their inverse is undefined), and neither the inverse nor the pseudo-inverse will yield true cosines unless $\alpha = 0$ (i.e., the graph is ignored). A secondary motivation from calculus on graphs is much more satisfying.

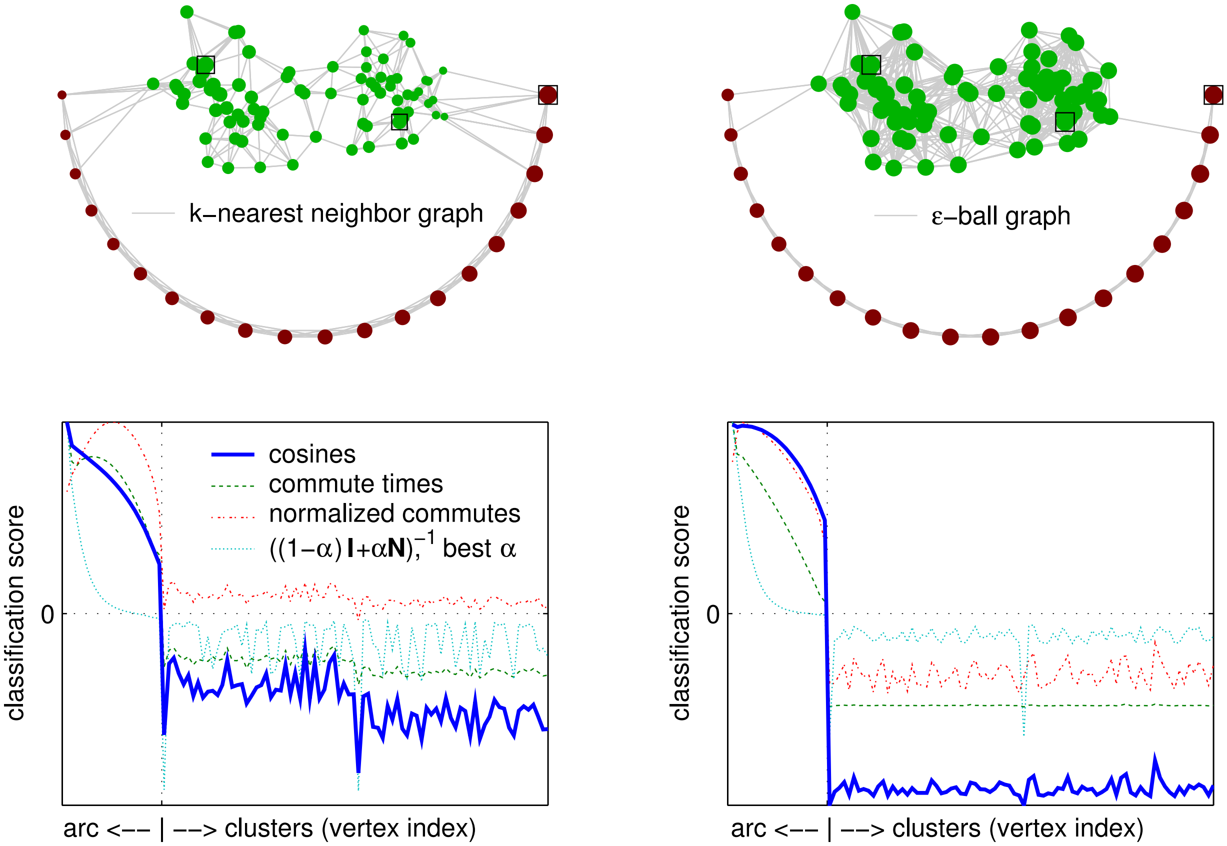

Figure 2: TOP: Classification of graph vertices according to random walk cosine correlations with labelled (boxed) vertices. Size and color of each vertex dot indicates the magnitude and sign of its classification score. BOTTOM: Point-by-point classification scores using various graph-based similarity and dissimilarity matrices. Each horizontal ordinate represents a point; the vertical ordinate is its classification score. Points scoring > 0 are classified as belonging to the arc. Classification scores are connected into lines only for the purpose of visual grouping. The cosine matrix offers the widest classification margin and most stability to small changes in the graph.

sification scores, depicted by the size and color of the graph vertices, are simply the difference between the recommendation score for two classes. The left and right panels illustrate how the classification varies when the criteria for adding edges to the graph changes. We experimented with different values for k and ε, and found that cosine correlations and commute times both perform well, in the sense of giving an intuitively correct classification that is relatively stable as the density of edges in the graph is varied . However, cosines offer a considerably wider classification margin and thus more robustness to graph perturbations. Normalized commute times, the Zhou & Schölkopf measure, hitting times, reverse hitting times, and their normalized variants (not shown) classify poorly to well on denser graphs but quite poorly on sparser graphs. The Zhou & Schölkopf measure in particular has a small margin because it is designed to vary smoothly over the graph. From this small informal experiment we may expect cosine correlations to give consistent recommendations under small variations in the association graph; this is borne out below in large cross-validation experiments.

4 Expected profit

While the consumer is interested in finding the next most interesting product, the vendor wants to recommend products that are also profitable. Assuming that most customers will make more than one purchase in the future and that customers' purchase decisions are independent of vendor profit margins, decision theory tells us that the optimal strategy is to recommend the product (state) with the greatest expected profit, discounted over time. That is, the vendor wants to nudge a consumer into a state from which a random walk will pass through highly profitable states (hence retail strategies such as "loss leaders"). Moreover, these states should be traversed early in the walk, because money is worth more now than it is in the indefinite future.

Let $\mathbf{p} \in \mathbb{R}^N$ be a vector of profit (or loss) for each state, and $e^{-\beta}, \beta > 0$ be a discount factor that determines the time value of future profits. The expected discounted profit v_i of the i^{th} state is the averaged profit of every state reachable from it, discounted for the time of arrival. In vector form:

$$\mathbf{v} = \mathbf{p} + e^{-\beta}\mathbf{T}\mathbf{p} + e^{-2\beta}\mathbf{T}^2\mathbf{p} + \cdots . \tag{4.10}$$

Using the identity $\sum_{i=0}^{\infty} \mathbf{X}^i = (\mathbf{I} - \mathbf{X})^{-1}$ for matrices of less than unit spectral radius ($\lambda_{\max}(\mathbf{X}) < 1$), we rearrange the series into a sparse linear system:

$$\mathbf{v} = \Big(\sum_{t=0}^{\infty} e^{-\beta t}\mathbf{T}^t\Big)\mathbf{p} = (\mathbf{I} - e^{-\beta}\mathbf{T})^{-1}\mathbf{p}. \tag{4.11}$$

The most profitable recommendation for a consumer in state i is thus the state j in the neighborhood of i that has the largest expected discounted profit: $j = \arg\max_{j \in \mathcal{N}(i)} T_{ij} v_j$. If the chain is structured so that states representing saleable products are k steps away from the current state, then the appropriate term is $\arg\max_{j \in \mathcal{N}(i)} T_{ij}^k v_j$.

5 Experiments

The MovieLens database [8] contains ratings on a scale of 1-5 for 1682 movies by 943 individuals. The data is a snapshot of what movies the university community considered worth seeing in 1997. Viewers rated 20-737 movies (average=106); movies received 1-583 ratings (average=60). The ratings table is 93.7% empty, which we interpret to mean that most viewers have not seen most movies. Movies are also tagged with nonexclusive memberships in 19 genres; individuals have 2 possible genders, 21 possible vocations, and 8 overlapping age groups. We constructed an $N = 2675$ state Markov chain with $W_{ij} = 1$ for each of these connections, except for movie ratings, which were copied directly into $\mathbf{W}$ on the principle that more highly rated movies are more likely choices. $\mathbf{W}$ is very sparse with less than 3% nonzero values. To evaluate the many measures of similarity and dissimilarity described above, we compared their performance in the following tasks.

5.1 Recommendation to maximize satisfaction We performed extensive cross-validation experiments to determine which statistic can best predict one part of the data from the rest. In each trial we randomly partitioned the data into a test set containing 10 ratings from each viewer, and a training set containing the remainder of the data. The goal is to "predict" each viewer's held-out movies. A Markov chain was constructed from the training set and a variety of similarity (e.g., cosine correlation) and dissimilarity (e.g., commute times) matrices were computed. Sorting the rows of these matrices gives a predicted per-viewer ranking of all movies. To score each measure's prediction quality, we took as recommendations the 10 top-ranked movies that were not in the training set, and summed the viewer's held-out ratings of each recommended movie. A cumulative score of zero means that the viewer did not elect to rate (or presumably, see) any of the recommendations. A cumulative score of 50 would mean that the viewer did indeed see all 10 recommendations and gave all the highest possible rating. When the data's average rating and sparsity is considered, an omniscient oracle could score no better than 35.3 on average; random guessing will score 2.2 on average. We performed 2500 trials with different random partitions and averaged scores over all viewers and trials. Figure 3 shows that cosine correlation is almost twice as successful as any other measure, with an average score slightly over 7. We also looked at how the predictors ranked the held-out movies: If a viewer had three held-out movies that the predictor ranked 5th, 17th, and 205th in her personalized recommendation list, then that predictor would be assessed a penalty of $5 + 17 + 205 = 227$. Cosine correlation had the smallest average penalty, roughly 1/4 the average penalty of commute times, the next best predictor.

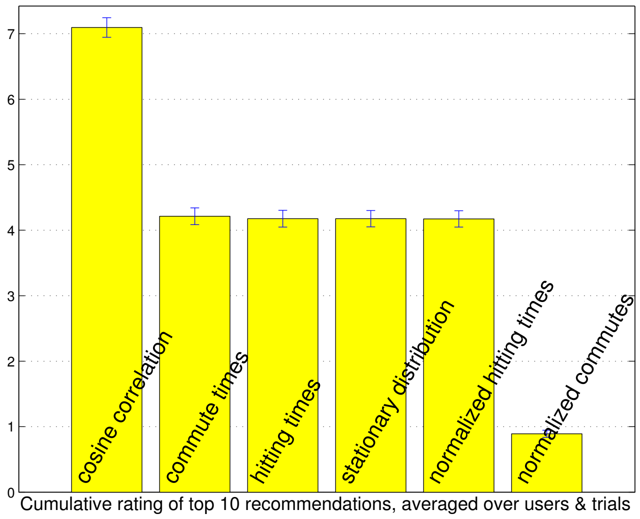

Figure 3: Cosine correlation is almost twice as effective as all other measures for predicting what movies a viewer will see and like.

Both sets of experiments were repeated with all ratings flattened to 1 ($W_{ij} \in \{0,1\}$), yielding almost identical comparative results. When ratings are not flattened, all methods show a bias for highly rated movies.

Consistent with results reported in [6], we found that commute times are slightly more informative than hitting times. That paper advocated commute times and demon-

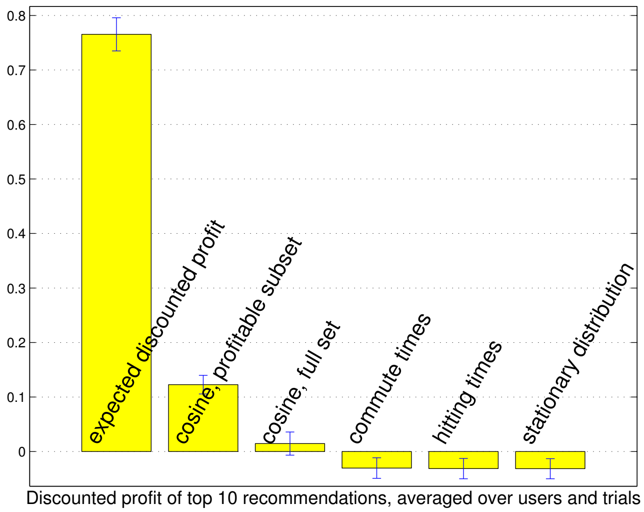

Figure 4: Making recommendations that maximize long-term profit is a much more successful strategy than recommending strictly profitable films, and profit-blind recommendations make no profit at all.

stationary distribution	correlated to 'male'	correlated to 'female'
Star Wars	Star Wars	The English Patient
Fargo	Contact	Contact
Return of the Jedi	Fargo	Titanic
Contact	Return of the Jedi	Jerry Maguire
Raiders of the Lost Ark	Air Force One	Conspiracy Theory
The Godfather	Scream	Sense and Sensibility
Toy Story	Toy Story	The Full Monty
Silence of the Lambs	Liar Liar	L.A. Confidential
Scream	The Godfather	Good Will Hunting

Table 1: Top recommendations made from the stationary distribution and by correlation to '`male`' and '`female`' states.

strated that they outperform *k*-nearest neighbors, Katz' influence measure [10], Djikstra shortest-path distances, and cosine coefficient computed from attribute vectors (*not* to be confused with cosine correlations of the random walk). However, we found that recommending from commute times is only slightly better than simply recommending the most popular movies identified by the stationary distribution, most likely because the commute and hitting times are both dominated by the stationary distribution. We tried several ways of pre-normalizing the data matrix **W** or post-normalizing the hitting/commute times to ameliorate this problem but failed to improve their predictions substantially and usually worsened performance, presumably because most normalizations are not consistent with the geometry of these statistics. For example, because commute times are small where stationary probabilities are large, [14, eq. (3.6-7)] proposed post-normalizing the commute times by the recurrence time (i.e., $C_{ij}/\sqrt{r_i r_j} = C_{ij} \cdot \sqrt{s_i s_j}$); we found this promoted unpopular movies so strongly that recommendation scores averaged worse than chance. The most successful normalization, after cosine correlations, was obtained by projecting the transition matrix to the closest doubly stochastic matrix prior to computing commute times, which makes the stationary distribution uniform (when such a projection exists).

5.2 Recommendations to maximize profit We repeated the experiments above with a slightly different scoring protocol: Before trials, each movie was randomly assigned a unique profit (or loss) p_j from a unit normal distribution. During trials, ten recommendations are given to the viewer in sequence; if the i^{th} recommendation is in the viewer's held-out list, the viewer accepts the movie and we receive a time-discounted profit of $e^{-i\beta} p_j$, with $e^{-\beta} = 0.9$. The goal is to maximize the profit over the entire sequence. In addition to the profit-blind predictors evaluated above, we considered a short-term profit maximizer—a cosine correlation predictor that only recommends movies with positive profits—and the long-term profit maximizer of section 4. This works by first suggesting the movie in a local graph neighborhood of the viewer's state that has maximum expected discounted profit. If the user declines, it suggests the next most profitable movie in the same neighborhood. If the user accepts, the state shifts to that of the accepted movie and the next suggestion comes from the graph neighborhood of that state. (This is one of many ways in which the state could be updated.) Figure 4 shows that the expected discounted profit maximizer strongly outperforms the greedy short-term maximizer, and that profit-blind recommenders effectively make no profit at all. (They show slight losses only because the random pricing happened to make some of the more popular movies unprofitable.)

5.3 Market analysis Recommendations can be made from any state in the chain, making it possible to identify products that are particularly successful with a consumer demographic, or customers that are particularly loyal to specific product categories. For example, the MovieLens data has `male` and `female` attributes that are indirectly linked to all movies through the viewers, and thus we may ask which movies are preferentially watched by men or women. Ranking movies by their commute times or expected hitting times from these states turns out to be uninformative, as the ranking is almost identical to the stationary distribution ranking. (This is understandable for men because the database is mostly male.) However, ranking by cosine correlation produces two very different lists, with males preferring ac-

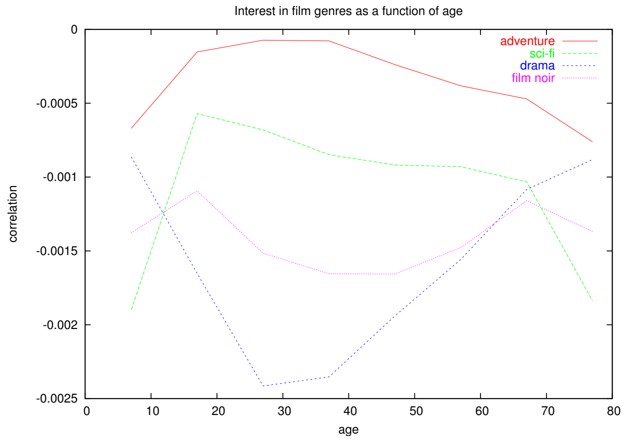

Figure 5: Correlation of age to genre preferences is weak but clearly shows that interest in sci-fi peaks in the teens and twenties. Soon after, interest in adventure peaks and interest in drama begins to climb.

tion and sci-fi movies and females preferring romances and dramas. Table 1 lists the top ten recommendations for each gender.

By the same method, we can ask which genres are preferentially watched by people of particular occupations and/or age groups. Figure 5 shows that age is indeed weakly predictive of genre preferences.

6 Conclusion

The random walks view of association graphs is a very natural way to study affinity relations in a relational database, providing a way to make use of extensive contextual information such as demographics and product categories in collaborative filtering tasks. We derived a novel measure of similarity—the cosine correlation of two states in a random walk—and showed that it is highly predictive for recommendation and semi-supervised classification tasks. Cross-validation experiments indicate that correlation-based rankings are more predictive and robust to perturbations of the graph's edge set than rankings based on commute times, hitting times, normalized Laplacians, and related graph-based dissimilarity measures. This is very encouraging because recommendations ought to be stable with respect to random omissions in the database, a challenge presented by most data-collection scenarios. We also sketched some efficient approximation methods for very large graphs; a forthcoming paper will detail very fast exact methods based on a modified sparse L-U decomposition.

Acknowledgments Thanks to anonymous readers and reviewers for helpful comments and pointers to [6, 14].

References

[1] Gediminas Adomavicius and Alexander Tuzhilin. Recommendation technologies: Survey of current methods and possible extensions. MISRC working paper 03-29, http://misrc.umn.edu/workingpapers/abstracts/0329.aspx, May 2003.

[2] D. Aldous and J. Fill. Reversible Markov chains and random walks on graphs. Manuscript, http://www.stat.berkeley.edu/users/aldous/RWG/book.html, In prep.

[3] P. Baldi, P. Frasconi, and P. Smyth. *Modeling the Internet and the Web: Probabilistic Methods and Algorithms*. John Wiley and Sons, 2003.

[4] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *Proc. 7th International World Wide Web Conference*, 1998.

[5] P.G. Doyle and J.L. Snell. *Random Walks and Electric Networks*. Mathematical Association of America, 1984.

[6] Francois Fouss, Alain Pirotte, Jean-Michel Renders, and Marco Saerens. A novel way of computing dissimilarities between nodes of a graph, with application to collaborative filtering. In *Proc., ECML workshop on Statistical Approaches for Web Mining*, 2004.

[7] F. Gobel and A. Jagers. Random walks on graphs. *Stochastic Processes and their Applications*, 2:311–336, 1974.

[8] J. Herlocker, J. Konstan, A. Borchers, and J. Riedl. An algorithmic framework for performing collaborative filtering. In *Proc. 1999 Conference on Research and Development in Information Retrieval*, 1999.

[9] Ngoc-Diep Ho and Paul Van Dooren. On the pseudo-inverse of the Laplacian of a bipartite graph. In *Proc. AML'04*, 2004.

[10] L. Katz. A new status index derived from sociometric analysis. *Psychometrika*, 18(1):39–43, 1953.

[11] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999.

[12] J. B. Kruskal and M. Wish. *Multidimensional Scaling*. Sage Publications, Beverly Hills, CA, 1978.

[13] J. Scott. *Social Network Analysis: A Handbook*. Sage Publications, London, 2000.

[14] D. Zhou and B. Schölkopf. Learning from labeled and unlabeled data using random walks. 2004.

A Computational strategies

For chains with $N \gg 10^3$ states, it is currently impractical to compute the full matrix of commute times or even a large matrix inversion of the form $(\mathbf{I}-\mathbf{X})^{-1} \in \mathbb{R}^{N\times N}$. To get around $O(N^2)$ memory and $O(N^3)$ time costs, we exploit the fact that most computations have the form $(\mathbf{I}-\mathbf{X})^{-1}\mathbf{G}$ where $\mathbf{X}$ is sparse and $\mathbf{G}$ has only a few columns. For many queries, only a subset of states are being compared ($\mathbf{G}$ is sparse as well), making only a subset of columns of the inverse necessary. These can be computed via the series expansions

$$(1.12) \qquad (\mathbf{I}-\mathbf{X})^{-1} = \sum_{i=0}^{\infty} \mathbf{X}^i = \prod_{i=0}^{\infty}(\mathbf{I}+\mathbf{X}^{2^i}),$$

which can be truncated to yield good approximations for fast-mixing sparse chains. In particular, an n-term sum of the additive series (middle) can be evaluated via $2\log_2 n$ sparse matrix multiplies via the multiplicative expansion (right). For any one column of the inverse this reduces to sparse matrix-vector products.

One problem is that these series only converge for matrices of less than unit spectral radius ($\lambda_{\max}(\mathbf{X}) < 1$). For inverses that do not conform, the associated series expansions will have a divergent component that can be incrementally removed to obtain the numerically correct result. For example, in the case of hitting times, we have $\mathbf{X} = \mathbf{T} + \mathbf{1s}^\top$ which has spectral radius 2. By expanding the additive series one can see that unwanted multiples of $\mathbf{1s}^\top$ accumulate very quickly in the sum. Instead, we construct an iteration that removes them as they arise:

$$\begin{aligned}
&(1.13) \qquad & \mathbf{A}_0 &\leftarrow \mathbf{I} - \mathbf{1s}^\top \\
&(1.14) \qquad & \mathbf{B}_0 &\leftarrow \mathbf{T} \\
&(1.15) \qquad & \mathbf{A}_{i+1} &\leftarrow \mathbf{A}_i + \mathbf{B}_i - \mathbf{1s}^\top \\
&(1.16) \qquad & \mathbf{B}_{i+1} &\leftarrow \mathbf{TB}_i\,,
\end{aligned}$$

which converges to

$$(1.17) \qquad \mathbf{A}_{i\to\infty} \to (\mathbf{I} - \mathbf{T} - \mathbf{1s}^\top)^{-1} + \mathbf{1s}^\top.$$

Note that this is easily adapted to compute an arbitrary subset of the columns of $\mathbf{A}_i$ and $\mathbf{B}_i$, making it economical to compute submatrices of $\mathbf{H}$. Because sparse chains tend to mix quickly, $\mathbf{B}_i$ rapidly converges to the stationary distribution $\mathbf{1s}^\top$, and we often find that $\mathbf{A}_i$ is a good approximation even for $i < N$. We can construct a much faster converging recursion for the multiplicative series:

$$\begin{aligned}
&(1.18) \qquad & \mathbf{A}_0 &\leftarrow \mathbf{I} - \mathbf{1s}^\top \\
&(1.19) \qquad & \mathbf{B}_0 &\leftarrow \mathbf{T} \\
&(1.20) \qquad & \mathbf{A}_{i+1} &\leftarrow \mathbf{A}_i + \mathbf{A}_i\mathbf{B}_i \\
&(1.21) \qquad & \mathbf{B}_{i+1} &\leftarrow \mathbf{B}_i^2.
\end{aligned}$$

This converges exponentially faster but requires computation of the entire $\mathbf{B}_i$. In both iterations, one can substitute $\mathbf{1}/N$ for $\mathbf{s}$; this merely shifts the column averages, which are removed in the final calculation

$$(1.22) \qquad \mathbf{H} \leftarrow (\mathbf{1}\mathrm{diag}(\mathbf{A}_i)^\top - \mathbf{A}_i)\mathrm{diag}(\mathbf{r}).$$

The recurrence times $r_i = s_i^{-1}$ can be obtained from the converged $\mathbf{B}_i = \mathbf{1s}^\top$.

It is possible to compute the inner product matrix $\mathbf{P}$ directly from the Markov chain parameters. The identity

$$(1.23) \qquad \mathbf{P} = (\mathbf{Q} + \mathbf{Q}^\top)/2$$

with

$$\begin{aligned}
\mathbf{Q} - \frac{1}{iN}\mathbf{11}^\top &= (\mathbf{I} - \mathbf{T} - \frac{i}{N}\mathbf{r1}^\top)^{-1}\mathrm{diag}(\mathbf{r}) \\
(1.24) \qquad &= (\mathrm{diag}(\mathbf{s}) - \mathrm{diag}(\mathbf{s})\mathbf{T} - \frac{i}{N}\mathbf{11}^\top)^{-1},\ 0 < i \leq N
\end{aligned}$$

can be verified by expansion and substitution. For a submatrix of $\mathbf{P}$, one need only compute the corresponding columns of $\mathbf{Q}$ using appropriate variants of the iterations above.

Once again, if $\mathbf{s}$ (and thus $\mathbf{r}$) are unknown prior to the iterations, one can make the substitution $\mathbf{s} \to \mathbf{1}/N$; at convergence the resulting $\mathbf{A}' = \mathbf{A}_i - \frac{1}{N}\mathbf{11}^\top$, $\mathbf{s} = \mathbf{1}^\top\mathbf{B}_i/\mathrm{cols}(\mathbf{B}_i)$, $r_i = s_i^{-1}$ satisfy

$$(1.25) \qquad \mathbf{A}' - \frac{1}{N}(\mathbf{A}'\mathbf{r} - \mathbf{1})\mathbf{s}^\top = (\mathbf{I} - \mathbf{T} - \frac{1}{N}\mathbf{r1}^\top)^{-1}$$

and

$$(1.26) \qquad \mathbf{Q} = \mathbf{A}'\mathrm{diag}(\mathbf{r})(\mathbf{I} - \frac{1}{N}\mathbf{11}^\top).$$

However, one pays a price for not pre-computing the stationary distribution $\mathbf{s}$: The last two equalities require full rows of $\mathbf{A}_i$, which defeats our goal of economically computing submatrices $\mathbf{P}$.

Such partial computations are quite feasible for undirected graphs with no self-loops: When $\mathbf{W}$ is symmetric and zero-diagonal, $\mathbf{Q}$ (equation 1.24) simplifies to the Laplacian kernel

$$(1.27) \qquad \mathbf{Q} = \mathbf{P} = (\mathbf{1}^\top\mathbf{W1})\cdot(\mathrm{diag}(\mathbf{W1}) - \mathbf{W})^+,$$

a pseudo-inverse because the Laplacian $\mathrm{diag}(\mathbf{W1}) - \mathbf{W}$ has a null eigenvalue. In our setting, the Laplacian has a sparse block structure that allows the pseudo-inverse to computed via smaller singular value decompositions of the blocks [9], but even this can be prohibitive. We avoid expensive pseudo-inversion entirely by shifting the null eigenvalue to 1, inverting via series expansion, and then shifting the eigenvalue back to zero. These operations are collected together in the equality

$$\begin{aligned}
\frac{1}{\mathbf{1}^\top\mathbf{W1}}\mathbf{P} &= \mathbf{D}((\mathbf{I} - \{\mathbf{D}(\mathbf{W} - \frac{i}{N}\mathbf{11}^\top)\mathbf{D}\})^{-1}\mathbf{D} \\
(1.28) \qquad & \quad -\frac{1}{iN}\mathbf{11}^\top,
\end{aligned}$$

where $\mathbf{D} \doteq \mathrm{diag}(\mathbf{W1})^{-1/2}$ and $i > 0$. By construction, the term in braces $\{\cdot\}$ has spectral radius < 1 for $i \leq 1$, thus any subset of columns of the inverse (and of $\mathbf{P}$) can be computed via straightforward additive iteration.

One advantage of couching these calculations in terms of sparse matrix inversion is that new data, such as a series of purchases by a customer, can be incorporated into the model via lightweight computations using the Sherman-Woodbury-Morrison formula for low-rank updates of the inverse.