PriView

Personalized Media Consumption Meets Privacy against Inference Attacks

Sandilya Bhamidipati and Nadia Fawaz, Technicolor

Branislav Kveton, Adobe Research

Amy Zhang, SET Media

// The PriView video consumption system protects user privacy while recommending relevant content. It distorts a user's video ratings to prevent attackers from inferring user attributes, while maintaining the distorted ratings' usefulness for recommendations. //

WITH THE ADVENT of targeted advertising and the popularity of data mining, users find their privacy threatened. To address this rising concern, researchers have proposed many privacy-preserving mechanisms.[1] Most of these mechanisms have strong theoretical guarantees but often lack practicality. For instance, reaching a sufficient level of privacy often requires sanitizing (distorting) user data to the point where it is no longer usable.

PriView is an interactive system for video consumption that provides privacy transparency and control while maintaining the quality of video recommendations based on user ratings. It shows how information-theoretic privacy can lead to practical policies for protecting user profiles while maintaining the sanitized data's utility.

The Privacy–Utility Framework

We assume a user has two kinds of data. Data vector A should remain private—for example, it could be a user's political views, age, and gender. However, the user wants to release data vector B to a service provider in exchange for some utility, such as the user's ratings of a TV show to get content recommendations based on his or her preferences. Because these two kinds of data are correlated, releasing show ratings might lead to indirectly revealing a user's private data through inference attacks. Surveys have shown that TV audiences can be distinctly characterized.[2]

We consider a local-privacy setting in which the joint probability distribution p_{AB} links A to B. (For more on local privacy and centralized privacy, see the sidebar.) So, an adversary who observes B could infer some information about A. Such an adversary could be an untrusted service provider or a third party with whom service providers might exchange data.

To reduce this inference threat, PriView releases the sanitized data $\hat{B}$, which it generates according to a conditional probabilistic mapping $p_{\hat{B}|B}$ called the *privacy mapping*. This mapping should make it more difficult to perform any statistical inference of A based on the observation of $\hat{B}$, while preserving some utility for $\hat{B}$ by limiting the distortion.

We adopt Flavio du Pin Calmon and Nadia Fawaz's privacy–utility framework.[3] In it, the privacy mapping controls the privacy leakage, modeled as the mutual information $I(A;\hat{B})$ between A and $\hat{B}$, subject to a utility requirement modeled by a constraint on the average distortion

$$E_{B,\hat{B}}\left[d(B,\hat{B})\right].$$

We focus on *perfect privacy*, $I(A;\hat{B})=0$: the mapping $p_{\hat{B}|B}$ renders

0740-7459/15/$31.00 © 2015 IEEE

RELATED WORK IN LOCAL AND CENTRALIZED PRIVACY

Researchers have studied privacy–utility tradeoffs for both local and centralized privacy. In local privacy, users do not trust the entity aggregating the data. So, they hold their data locally and process it according to a privacy-preserving mechanism before releasing it to the aggregator. Local privacy dates back to randomized response in surveys[1] and has been considered in data mining and statistics.[2–7] PriView (see the main article) falls in this category because it assumes that the service provider is untrusted and that users want to protect their private information from inference attacks.

Researchers have also considered local privacy in the context of differential privacy in terms of learning aggregate statistical properties from several users' data.[7–9] In contrast, we devise content recommendations for individual users while maintaining the privacy of users' attributes.

In centralized privacy, a trusted entity aggregates data from users in a database while an untrusted analyst queries the database. The aggregator processes that data through a centralized privacy-preserving mechanism to produce a privatized answer to the query.

Researchers have used information-theoretic frameworks to analyze asymptotic privacy–utility tradeoffs in centralized databases, as the number of data samples grows large.[10,11] Much differential-privacy research has assumed a centralized setting with a trusted database owner and has focused on making the output of an application running on the database differentially private.[8,9,12] In particular, Frank McSherry and Ilya Mironov considered a trusted recommender system that accessed ratings from privacy-conscious users.[12] They addressed the challenge of training a differentially private recommendation algorithm on the basis of those original ratings.

In contrast, we assume that the system already owns a recommendation algorithm that uses ratings from users who aren't privacy conscious. We address the privacy challenges faced by any privacy-conscious user who wants to use this untrusted system.

References

1. S.L. Warner, "Randomized Response: A Survey Technique for Eliminating Evasive Answer Bias," *J. Am. Statistical Assoc.*, vol. 60, no. 309, 1965, pp. 63–66.
2. F. Calmon and N. Fawaz, "Privacy against Statistical Inference," *Proc. 2012 Allerton Conf. Communication, Control, and Computing* (ALLERTON 12), 2012; http://arxiv.org/abs/1210.2123.
3. S. Salamatian et al., "How to Hide the Elephant—or the Donkey—in the Room: Practical Privacy against Statistical Inference for Large Data," *Proc. 2013 IEEE Global Conf. Signal and Information Processing* (GlobalSIP 13), 2013, pp. 269–272.
4. A. Evfimievski, J. Gehrke, and R. Srikant, "Limiting Privacy Breaches in Privacy Preserving Data Mining," *Proc. 22nd ACM SIGMOD-SIGACT-SIGART Symp. Principles of Database Systems* (PODS 03), 2003, pp. 211–222.
5. S. Salamatian et al., "Managing Your Private and Public Data: Bringing Down Inference Attacks against Your Privacy," ArXiv e-prints, 2014; http://arxiv.org/abs/1408.3698.
6. D. Rebollo-Monedero, J. Forne, and J. Domingo-Ferrer, "From t-Closeness-Like Privacy to Postrandomization via Information Theory," *IEEE Trans. Knowledge and Data Eng.*, vol. 22, no. 11, 2010, pp. 1623–1636.
7. J.C. Duchi, M.I. Jordan, and M.J. Wainwright, "Local Privacy and Statistical Minimax Rates," *Proc. IEEE 54th Ann. Symp. Foundations of Computer Science* (FOCS 13), 2013, pp. 429–438.
8. C. Dwork et al., "Calibrating Noise to Sensitivity in Private Data Analysis," *Proc. 3rd Conf. Theory of Cryptography* (TCC 06), 2006, pp. 265–284.
9. C. Dwork, "Differential Privacy," *Automata, Languages and Programming*, LCNS 4052, Springer, 2006, pp. 1–12.
10. H. Yamamoto, "A Source Coding Problem for Sources with Additional Outputs to Keep Secret from the Receiver of Wiretappers," *IEEE Trans. Information Theory*, vol. 29, no. 6, 1983, pp. 918–923.
11. L. Sankar, S.R. Rajagopalan, and H.V. Poor, "Utility-Privacy Tradeoffs in Databases: An Information-Theoretic Approach," *IEEE Trans. Information Forensics and Security*, vol. 8, no. 6, 2013, pp. 838–852.
12. F. McSherry and I. Mironov, "Differentially Private Recommender Systems: Building Privacy into the Net," *Proc. 15th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining* (KDD 09), 2009, pp. 627–636.

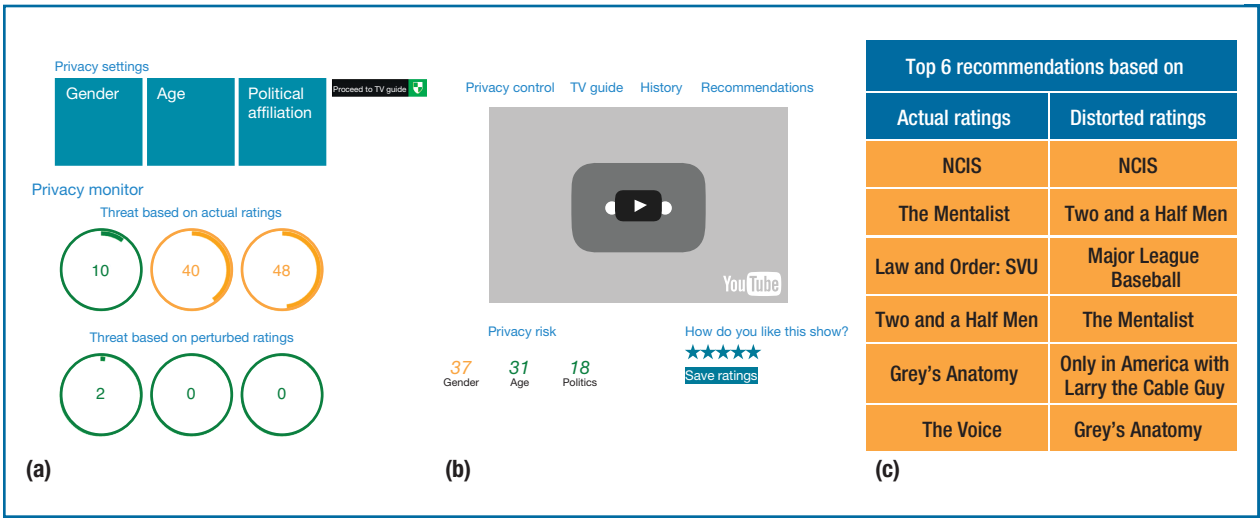

FIGURE 1. PriView overview. (a) The PriView privacy dashboard ("perturbed" means distorted). (b) The PriView show page. (c) An example of top six recommendations (shown as a table here with TV show names instead of images, owing to licensing and copyright issues).

$\hat{B}$ statistically independent from A. Any inference algorithm that tries to infer A from $\hat{B}$ cannot outperform an uninformed random guess.

In the local-privacy setting, perfect privacy is equivalent to statistical independence between A and $\hat{B}$. In other words,

$$p_{\hat{B}|A}\left(\hat{b} \mid a\right) = p_{\hat{B}|A}\left(\hat{b} \mid a'\right) = p_{\hat{B}}\left(\hat{b}\right)$$

for all a, a', and $\hat{b}$, which in turn is equivalent to $\hat{B}$ being locally 0-differential private with respect to A.

The System Architecture

PriView has three components: a user client, a privacy server, and a recommendation server. The client is a Web interface written in HTML5 and JavaScript. It lets users interact with the privacy settings, lets them watch and rate TV shows, and displays recommendations based on the privacy settings and privatized ratings.

The servers are written in Flask, a Python-based micro-Web framework. They serve client requests and store and fetch data from databases. The privacy server also privatizes ratings according to users' privacy settings and sends the privatized ratings to the recommendation server and user client. The recommendation server generates recommendations based on the privatized ratings and sends them to the user client.

Four data collections are stored in MongoDB databases. One stores users' privacy settings and interactions with the content, such as show ratings; another collection stores privacy-mapping data. Both are accessed by the privacy server. A third collection stores the content metadata displayed on the client's Web interface; the fourth stores content profiles for recommendation purposes. These two are accessed by the recommendation server.

PriView can run as a local privacy agent on the user side (for example, as a plug-in), before the release of sanitized ratings to the service provider. It does not require modifications on the service-provider side.

The Dataset

PriView uses the Politics and TV dataset, which gathered data on US viewers' political views and TV preferences in the fall of 2012.[4] The dataset contains entries for 1,218 users, 744 of whom identified as Democrats and 474 of whom identified as Republicans. For each user, the dataset entry is a vector [*age*, *gender*, *state*, *politics*, B_1, …, B_{50}], where $B_i \in \{0, 1, \ldots, 5\}$ is the user's 5-star rating for TV show i. The actual ratings range from 1 to 5, where 5 is the highest rating; 0 means the user didn't rate the show. The ratings were for 50 TV shows in six categories: sitcoms, reality shows, TV series, talk shows, news, and sports.

Functionalities

Here we examine in detail PriView's three main functionalities.

Transparency

The privacy dashboard (see Figure 1a) shows users' privacy settings and the privacy monitor. Users do not need to reveal their age, gender, or political views; they need to reveal only whether they consider any of these features sensitive information they want to keep private. The privacy monitor shows the inference threat for each private attribute from users' actual TV show ratings and from the sanitized ratings. So, users can compare the risks if they don't activate privacy protection with the risks after the privacy mapping sanitizes the ratings.

To model the inference threat for each private attribute from a particular rating vector representing a user's history of ratings, we employ a privacy risk metric on a scale of 0 to 100. For private attribute A and the specific vector of ratings $B = b$, the privacy risk is

$$Risk(A,b) = 100 * \left(1 - \frac{H(A \mid B = b)}{H(A)}\right)^{+}. \quad (1)$$

$H(A) = -\Sigma_a p_A(a) \log p_A(a)$ denotes the entropy of A distributed according to $p_A(a)$ and represents the inherent uncertainty of A. Similarly,

$$H(A \mid B = b) = -\Sigma_a p_{A|B}(a \mid b) \log p_{A|B}(a \mid b)$$

denotes the remaining entropy of A given the observation $B = b$ and represents the remaining uncertainty of A.

Intuitively, $Risk(A, b)$ measures the percentage by which the uncertainty of A decreases owing to the observation of $B = b$, relative to the original uncertainty before observing B. $Risk(A, b) = 0$ means that $B = b$ provides no information about A; $Risk(A, b) = 100$ implies that no uncertainty is left about A from observing $B = b$. The privacy risk based on a user's actual rating vector $B = b$ is $Risk(A, b)$, whereas the privacy risk based on the sanitized ratings $\hat{B} = \hat{b}$ is $Risk(A,\hat{b})$, which we obtain by replacing $B = b$ in Equation 1 with $\hat{B} = \hat{b}$.

After selecting privacy settings, users can move to the TV guide (not shown here) and choose a show to watch. As Figure 1b illustrates, users can rate shows on each show page. Before users rate a show, a privacy risk tool reminds them of the privacy risk based on their history of actual ratings. When users hover above the rating stars for a new show, for each possible rating of one to five stars, the privacy risk tool dynamically updates its numbers to inform users of how the privacy risk would evolve if they added a particular rating. This tool shows the risk based on actual ratings before sanitization. Once users choose and submit a rating, the privacy-preserving mechanism sanitizes the rating vector. The privacy dashboard lets users verify that the privacy risk after ratings distortion is 0 for the attributes they selected as private.

Control

As we mentioned before, users can select which attributes they want to remain private. The system implements a privacy-preserving mechanism for releasing user ratings to a service provider. This mechanism ensures perfect privacy against the statistical inference of private features[3] while minimizing the released data's distortion. Finally, a history log (not shown here) lets users see their true ratings and the sanitized ratings.

Challenges. While implementing the privacy–utility framework, we encountered the following technical challenges that required adapting it.

The first challenge was scalability. Designing the privacy mapping $P_{\hat{B}|B}$ requires characterizing the value of

$$p_{\hat{B}|B}(\hat{b} \mid b)$$

for all possible pairs

$$(b,\hat{b}) \in \mathbf{B} \times \hat{\mathbf{B}}$$

or solving the convex optimization problem over $|\mathbf{B}||\hat{\mathbf{B}}|$ variables. When $\mathbf{B} = \hat{\mathbf{B}}$ and the size of the alphabet $|\mathbf{B}| = 6^{50}$ is large, solving the convex optimization over $|\mathbf{B}|^2$ variables might be intractable. Salman Salamatian and his colleagues proposed quantization to reduce the number of optimization variables from $|\mathbf{B}|^2$ to K^2, where K denotes the number of quantization levels.[4] The choice of K is a tradeoff between the optimization's size and the additional distortion introduced by quantization.

Quantization assumes that B lies in a metric space. Directly applying quantization on the original rating vector B, where unrated shows have a 0 rating, would make our model perceive unrated shows as strongly disliked by the user, when they might actually be unknown.[4] To circumvent this issue, we first transform B into the completed rating vector B_c using low-rank *matrix factorization* (MF), a standard collaborative-filtering technique. We then feed B_c to the quantization module, which maps it to a cluster center C. For quantization, we use K-means clustering with $K = 75$ cluster centers, where our choice of K was guided empirically. C is then fed to the privacy optimization algorithm, which outputs $\hat{B}$. The privacy-mapping algorithm (see Figure 2) follows the Markov chain $A \to B \to B_c \to C \to \hat{B}$.

The second challenge was estimating the prior distribution. Computing $Risk(A, b)$ and finding the privacy mapping as the solution to the privacy convex optimization discussed by Calmon and Fawaz[3] rely on the fundamental assumption that $p_{A,B}$ is known and can be fed as input to the algorithm. In practice, the true distribution might not be known but could be estimated from a sample dataset. Such a dataset could come from a set of users who don't have privacy concerns and publicly disclose both A and B. However, it might contain a small number of samples or be incomplete, which makes estimating the prior distribution challenging.

Salamatian and his colleagues studied in detail the case of a mismatched estimate of the prior distribution and its impact on the privacy–utility tradeoff.[5] Using the completion and quantization step, we adapted Calmon and Fawaz's framework to use the prior distribution between the private data and quantized completed data in the algorithm in Figure 2. We estimate the distribution using kernel density estimation, with a Gaussian kernel of width $\sigma = 9.5$.

Input: prior $p_{A,C}$

Solve: convex optimization

$$\text{minimize } E_{C,\hat{B}}\left[d\left(C,\hat{B}\right)\right]$$

$$p_{\hat{B}|C}$$

$$\text{subject to } I\left(A;\hat{B}\right) \leq \epsilon$$

$$p_{\hat{B}|C} \in \text{Simplex}$$

Remap: $p_{\hat{B}|B} \leftarrow p_{\hat{B}|C(B)}$

Output: mapping $p_{\hat{B}|B}$

FIGURE 2. The privacy-mapping algorithm, which follows the Markov chain $A \rightarrow B \rightarrow B_C \rightarrow C \rightarrow \hat{B}$.

Evaluation. In Figure 2, ϵ bounds the amount of information about A leaked by $\hat{B}$ and thus represents the level of the user's privacy requirements. Varying ϵ lets us study the tradeoff between privacy requirements and distortion. K-means quantization introduces a distortion of 1.08 per rating and yields mutual information $I(A; C) = 0.2$. With 0.14 additional distortion, the privacy mapping achieves perfect privacy, $I\left(A;\hat{B}\right) = 0$, for an end-to-end distortion of 1.22.

PriView focuses on perfect privacy and thus on ϵ close to 0. As we mentioned before, at perfect privacy, any inference algorithm that tries to infer A from $\hat{B}$ can perform only as well as an uninformed random guess. Intuitively, $\hat{B}$ is statistically independent from A; thus, the privacy mapping statistically "erases" any information about A from $\hat{B}$. An inference algorithm that tries to infer A from $\hat{B}$ can perform only as well as an uninformed inference algorithm that would try to infer A without knowledge of $\hat{B}$.

Figure 3 shows a receiver-operating-characteristic (ROC) curve illustrating the performance of a logistic-regression classifier that tried to infer a user's political views from

- the original rating vector (the blue curve),
- a binarized version of the vector in which ratings of ≥4 were mapped to 1 (the viewer liked the show) and ratings of ≤3 were mapped to 0 (the viewer disliked the show),
- vectors with an average distortion of ≤1 (the pink curve), and
- vectors with an average distortion of ≤2 (the red curve).

An ROC curve plots the true-positive rate against the false-positive rate of a binary classifier at various thresholds.

We used 10-fold cross validation to plot the false-positive rate (Democrats falsely classified as Republicans) against the true-positive rate (Republicans correctly classified). The blue curve illustrates the privacy risk of inferring a user's political views from the original rating vectors. The green curve is close to the blue curve and shows that merely binarizing the ratings will not ensure privacy. The red curve is close to the red diagonal line, which represents an uninformed random guess. This proves that with a distortion of ≤2, the privacy-preserving mechanism ensures perfect privacy against logistic regression of political views from the sanitized ratings. Additional inference attacks with other common classifiers, including naive Bayes and support vector machines, produced similar results.

Personalized Recommendations

PriView recommends video content on the basis of the users' released ratings. A natural question is whether the recommendations' relevance can be preserved when they're based on sanitized ratings. PriView's recommendations page lets users compare the top six TV show recommendations based on their actual ratings and on

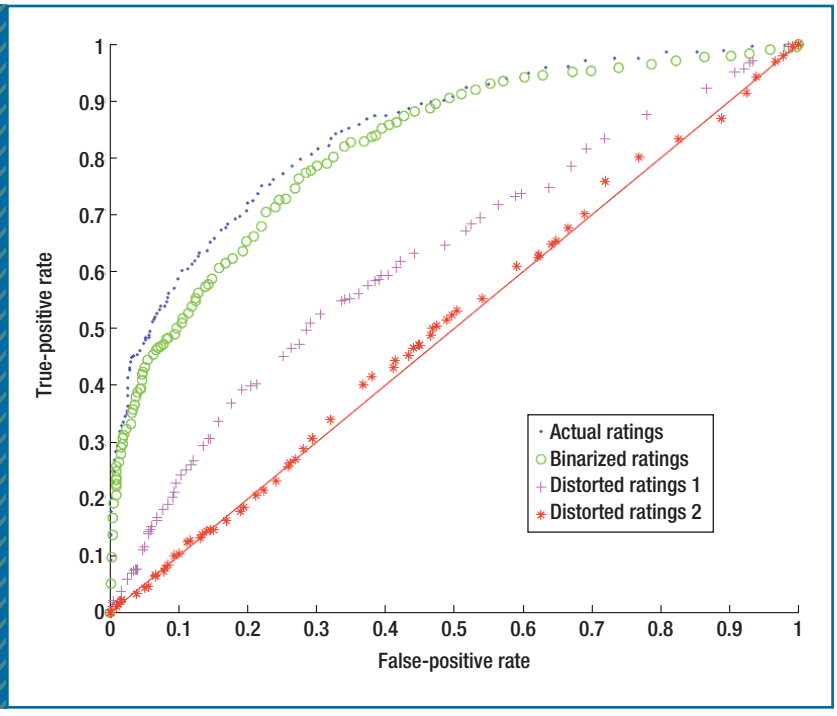

FIGURE 3. A receiver-operating-characteristic (ROC) curve illustrating the performance of a logistic-regression classifier that tried to infer a user's political views from various versions of online movie ratings. With a distortion of ≤2, our privacy-preserving mechanism ensured perfect privacy.

their sanitized ratings (see Figure 1c). PriView's recommendation engine uses low-rank MF to predict missing show ratings from ratings provided by the user for other shows.[6] We trained the MF recommender engine by alternating regularized least squares.[6] Figure 1c shows an overlap of four of the six recommendations without and with privacy, illustrating that PriView maintained utility while protecting user privacy.

We conducted further testing to illustrate that PriView can eliminate the privacy threat from $\hat{B}$ for A with little effect on recommendation quality. We used five-fold cross validation to split our dataset into a training set containing 80 percent of the data and a test set containing the remaining 20 percent. Using that data, we tested the MF recommender engine with and without privacy to compare the relevance of recommendations in these two cases. In each test set, we randomly removed and tried to predict 10 percent of the ratings and calculated the corresponding RMSE.

The RMSE for prediction based on the actual ratings ranged from 1.24 to 1.34. When political views were protected, the RMSE based on the sanitized ratings ranged from 1.34 to 1.42. When gender was protected, the RMSE based on the sanitized ratings ranged from 1.34 to 1.43. So, the RMSE for rating prediction did not degrade much with privacy protection.

Those results were for the case of perfect privacy. If the privacy requirements were less stringent, such as

$$I\left(A;\hat{B}\right)\leq\epsilon$$

for some $\epsilon > 0$, the RMSE for prediction with privacy protection would be even closer to the RMSE without privacy. Using a more advanced and optimized recommendation engine instead of the standard MF recommendation engine would yield better rating predictions both without and with privacy protection.

PriView can interface with online video services and with TV and video-on-demand services. It could also be extended to other media content, such as music, books, and news, and to other products, services, or locations that are rated online by users. PriView can be extended to locations outside the US, provided data for them is available.

In addition, PriView could be adapted to protect privacy in the context of social networks. Users could be informed of the privacy risks of actions such as liking a page or comment or adding friends before taking those actions, and could control those risks. In such a context, data distortion could amount to simply avoiding some actions or preventing the release of some data.

Further extensions also include broadening the set of private attributes that users deem sensitive and analyzing the temporal dynamics of privacy and utility in a real-time setting with a system such as PriView. We're looking at extending PriView to cases in which adversaries can access side information, such as additional information about users, before users submit ratings. Future research will include studying how sanitized rating data affects the recommendation engine's quality if the engine was trained on sanitized data.

Acknowledgments

Technicolor owns the PriView intellectual property. This work was performed while all four authors were employed by Technicolor.

References

1. B.-C.M. Fung et al., "Privacy Preserving Data Publishing: A Survey of Recent Developments," *ACM Computing Surveys*, vol. 42, no. 4, 2010, article 14.
2. J. Fetto, "Top TV Shows for Reaching Key Voters," Experian Marketing Services, Aug. 2012; www.experian.com/blogs/marketing-forward/2012/08/28/top-tv-shows-for-reaching-key-voters.
3. F. Calmon and N. Fawaz, "Privacy against Statistical Inference," *Allerton Conf. Communication, Control, and Computing* (ALLERTON 12), 2012; http://arxiv.org/abs/1210.2123.
4. S. Salamatian et al., "How to Hide the Elephant—or the Donkey—in the Room: Practical Privacy against Statistical Inference for Large Data," *Proc. 2013 IEEE Global Conf. Signal and Information Processing* (GlobalSIP 13), 2013, pp. 269–272.
5. S. Salamatian et al., "Managing Your Private and Public Data: Bringing Down Inference Attacks against Your Privacy," to appear in *IEEE J. Selected Topics in Signal Processing*, Oct. 2015; http://arxiv.org/abs/1408.3698.
6. Y. Koren, R. Bell, and C. Volinsky, "Matrix Factorization Techniques for Recommender Systems," *Computer*, vol. 42, no. 8, 2009, pp. 30–37.

ABOUT THE AUTHORS

SANDILYA BHAMIDIPATI is a systems architect at Technicolor. He leads engineering efforts in content discovery, recommendation systems, user analytics, and user privacy. His research interests include applied machine learning and data mining for large-scale systems. Bhamidipati received an MS in computer science from Rutgers University. Contact him at sandilya.bhamidipati@technicolor.com.

NADIA FAWAZ is a senior researcher at Technicolor. Her research interests include data privacy and personalization. Fawaz received a PhD in electrical engineering from École Nationale Supérieure des Télécommunications de Paris and EURECOM, France. She's a member of IEEE and ACM. Contact her at nadia.fawaz@technicolor.com.

BRANISLAV KVETON is a machine-learning scientist at Adobe Research. Most of his recent work focuses on online learning of structured problems such as graphs, submodularity, matroids, polymatroids, and reinforcement learning. Kveton received a PhD in intelligent systems from the University of Pittsburgh. Contact him at kveton@adobe.com.

AMY ZHANG is a machine-learning engineer at SET Media. Her research interests include deep learning, graphical models, and inference. Zhang received an MEng in electrical engineering and computer science from MIT. Contact her at amy@set.tv.