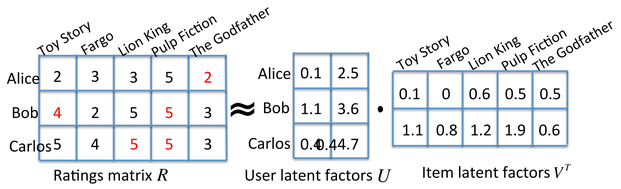
Active Learning and Search on Low-Rank Matrices

Dougal J. Sutherland, **Barnabás Póczos**, and **Jeff Schneider**
Auton Lab, School of Computer Science, Carnegie Mellon University – autonlab.org

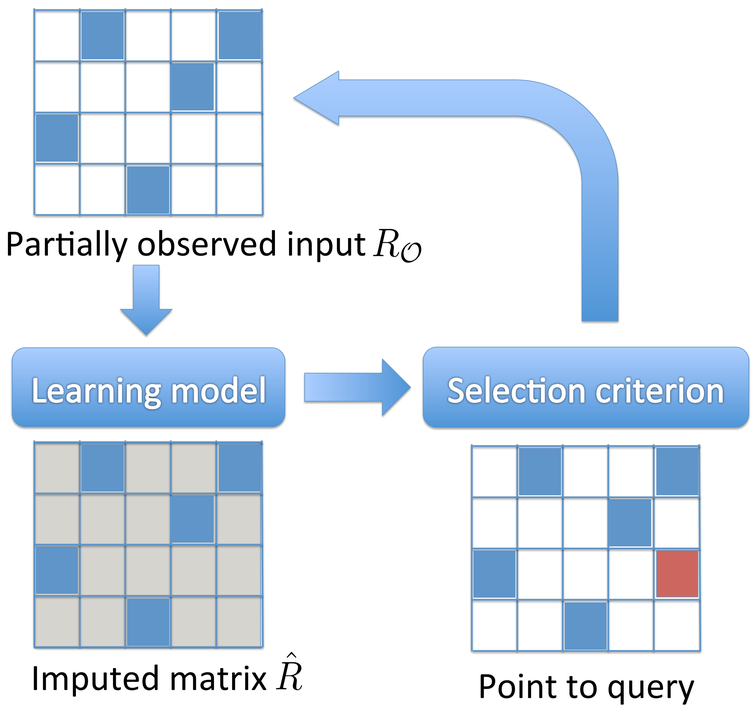
Matrix factorization

Low-rank matrix factorization is a very powerful and popular technique used in recommender systems and a variety of other application areas. Given a partially observed matrix, it does a good job of imputing the other elements of the matrix.



Active matrix factorization

Sometimes, however, we have the ability to obtain a certain element of the matrix (though doing so may be expensive, e.g. in drug discovery). *Active* matrix factorization asks how we can best choose matrix elements to query in order to accomplish our learning goals.



Learning goals

Prediction: minimize prediction error

$$\min \mathbb{E}\left[(R_{ij} - \hat{R}_{ij})^2 \mid (i,j) \notin \mathcal{O}\right]$$

Model: minimize uncertainty in the data model

$$\min H\left[\text{model} \mid R_\mathcal{O}\right]$$

Magnitude Search: find the largest matrix elements

$$\max \sum_{(i,j)\in\mathcal{A}} R_{ij}$$

Search: find many elements from a given class

$$\max \sum_{(i,j)\in\mathcal{A}} \mathbb{1}(R_{ij} \in +)$$

Acknowledgements

This work was funded in part by the National Science Foundation under grant NSF-IIS0911032 and the Department of Energy under grant DESC0002607.

Probabilistic Matrix Factorization (PMF)

Our learning model will be the PMF generative factorization model for matrices of a specific rank [1]:

$$R_{ij} \sim N\left(U_i^T V_j, \sigma^2\right) \quad U_i \sim N\left(0, \sigma_U^2 I_D\right) \quad V_j \sim N\left(0, \sigma_V^2 I_D\right)$$

$$-\ln p(U, V \mid R_\mathcal{O}) = \frac{1}{2\sigma^2}\|I \circ (R - UV^T)\|_F^2 + \frac{1}{2\sigma_U^2}\|U\|_F^2 + \frac{1}{2\sigma_V^2}\|V\|_F^2 + C$$

It's easy to get a point estimate for U and V (though the objective is biconvex). But for active learning, we want uncertainty in the model.

Variational PMF

One method to get this is to approximate $p(U, V)$ by some parametric distribution $q(U, V)$, and find the best such approximation by minimizing KL divergence:

$$\begin{aligned} KL(q\|p) &= \int q(U,V) \ln \frac{q(U,V)}{p(U,V \mid R_\mathcal{O})} d\{U,V\} \\ &= -H[q] - \mathbb{E}_q\left[\ln p(U,V \mid R_\mathcal{O})\right] \\ &= -H[q] - C + \frac{1}{2\sigma_U^2}\sum_{i=1}^{N}\sum_{k=1}^{D}\mathbb{E}_q[U_{ik}^2] + \frac{1}{2\sigma_V^2}\sum_{j=1}^{M}\sum_{k=1}^{D}\mathbb{E}_q[V_{jk}^2] \\ &\quad + \frac{1}{2\sigma^2}\sum_{i=1}^{N}\sum_{j=1}^{M}\left(\sum_{k=1}^{D}\sum_{\ell=1}^{D}\mathbb{E}_q[U_{ki}V_{kj}U_{\ell i}V_{\ell j}] - 2R_{ij}\sum_{k=1}^{D}\mathbb{E}_q[U_{ki}V_{kj}] + R_{ij}^2\right) \end{aligned}$$

Some choices for q are

1. a normal distribution on the elements of U and V
2. fully factorized distribution on each element [2]
3. "in between": a matrix normal distribution on their concatenation. We try this approach.

In each case we can minimize $\mathrm{KL}(q\|p)$ through (projected) gradient descent.

Markov chain Monte Carlo

Another option is to sample from $p(U, V)$ using MCMC, and use the samples for inference.

BPMF [3] extends PMF by adding arbitrary means and covariances to the priors on U_i and V_j, with Gaussian-Wishart hyperpriors, and samples via Gibbs. We use Hamiltonian MCMC with the No-U-Turn Sampler [4].

Myopic selection criteria

Prediction: uncertainty sampling

$$\arg\max_{(i,j)} \mathrm{Var}[R_{ij}]$$

Model: ?

Magnitude Search: the largest-mean element

$$\arg\max_{(i,j)} \mathbb{E}[R_{ij}]$$

Search: the element most likely to be positive

$$\arg\max_{(i,j)} \mathbb{P}(R_{ij} \in +)$$

Lookahead selection criteria

We can also define a quality measure f and consider integrate over possible outcomes:

$$\int_x \hat{\mathbb{P}}(R_{ij} = x)\, \mathbb{E}\left[f(q) \mid R_\mathcal{O}, R_{ij} = x\right]$$

Prediction: entropy of predicted matrix

$$f(q) = H[R]$$

Model: entropy of posterior over U and V

$$f(q) = H[U, V]$$

Magnitude search: mean of found elements

$$f(q) = R_{ij} + \max_{(k,l)\in\mathcal{P}-(i,j)} \mathbb{E}[R_{kl}]$$

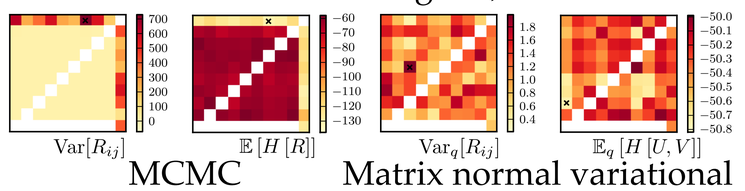
Search: expected number of positives found

$$f(q) = \mathbb{1}(R_{ij} \in +) + \max_{(k,l)\in\mathcal{P}-(i,j)} \mathbb{P}(R_{kl} \in +)$$

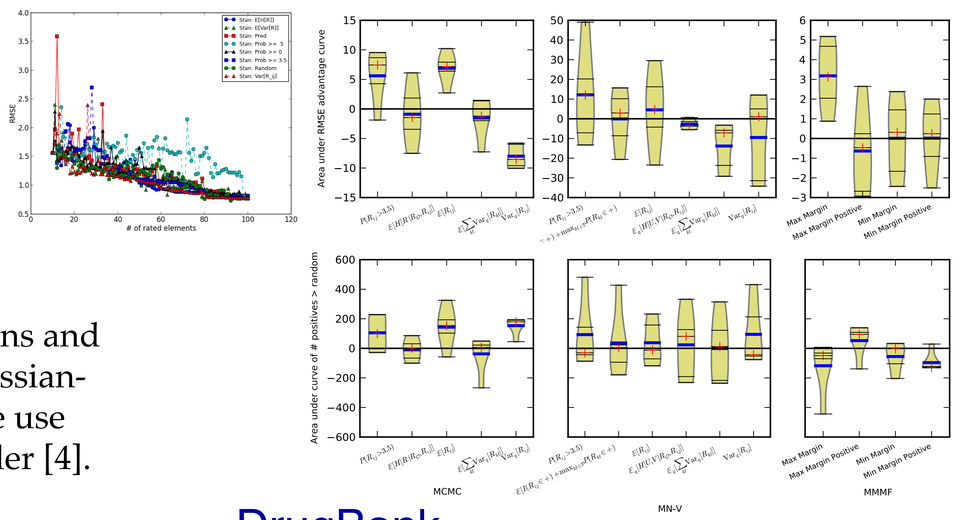
This can easily be extended to more than one lookahead step, and is optimal with full lookahead [6], but cost increases exponentially.

Toy Experiments

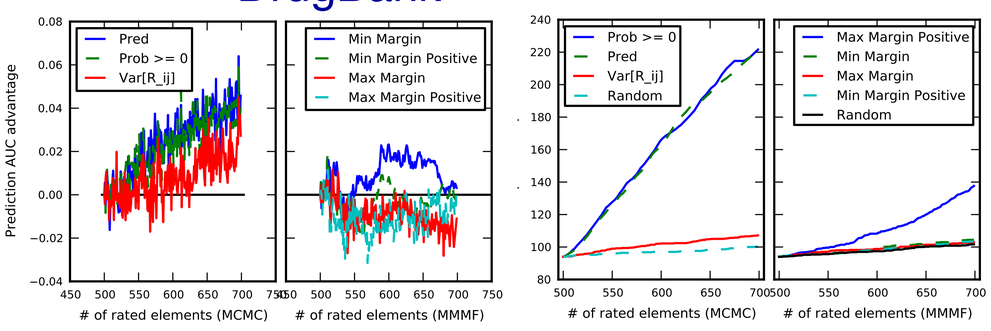
Rank 1 matrix: MCMC is good, variational bad.



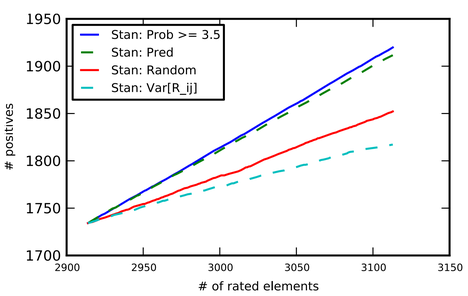
10x10 rank 2, values 1-5:



DrugBank



MovieLens



References

[1] Salakhutdinov & Mnih. Probabilistic matrix factorization. NIPS 2007.
[2] Silva & Carin. Active learning for online Bayesian matrix factorization. KDD 2012.
[3] Salakhutdinov & Mnih. Bayesian probabilistic matrix factorization using Markov chain Monte Carlo. ICML 2008.
[4] Hoffman & Gelman. The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo. JMLR, in press.
[5] Rish & Tesauro. Active collaborative prediction with maximum margin matrix factorization. ISAIM Info. Th. & App. 2007.
[6] Garnett et al. Bayesian Optimal Active Search and Surveying. ICML 2012.