Learning the Kernel?

From Multitask Learning to Collaborative Filtering

Alex Smola
Yahoo! Research, Santa Clara, CA
alex.smola.org im:alex.smola skype:smolix

with Deepay Chakrabarty, Wei Chu, Markus Weimer

Outline

Outline

- Three problems - one solution
  - Multitask learning
  - Collaborative filtering
  - Learning the kernel

Outline

- Three problems - one solution
  - Multitask learning
  - Collaborative filtering
  - Learning the kernel
- Factorization approach
  - Applications
  - Extensions (Tucker factors, data integration)
  - Optimization

Outline

- Three problems - one solution
  - Multitask learning
  - Collaborative filtering
  - Learning the kernel
- Factorization approach
  - Applications
  - Extensions (Tucker factors, data integration)
  - Optimization
- Experiments
  - User profiles
  - Webpage categorization with side information

Three problems - one solution

Learning the Kernel

Learning the Kernel

- Empirical risk (how well you do on the data)
  - Classification
  - Regression
  - Ranking
  - Graphical Models (CRF, M3M)

$$R_{\text{emp}}[f] = \frac{1}{m} \sum_{i=1}^{m} l(y_i, f(x_i))$$

Learning the Kernel

- Empirical risk (how well you do on the data)
  - Classification
  - Regression
  - Ranking
  - Graphical Models (CRF, M3M)

$$R_{\text{emp}}[f] = \frac{1}{m}\sum_{i=1}^{m} l(y_i, f(x_i))$$

- Function Regularizer
  - RK **Hilbert Space** norm
  - Sparsity (no kernel here)

$$\Omega[f] = \frac{1}{2}\|f\|_{\mathcal{H}}^2$$

Learning the Kernel

- Empirical risk (how well you do on the data)
  - Classification
  - Regression
  - Ranking
  - Graphical Models (CRF, M3M)

$$R_{\text{emp}}[f] = \frac{1}{m} \sum_{i=1}^{m} l(y_i, f(x_i))$$

- Function Regularizer
  - RK **Hilbert Space** norm
  - Sparsity (no kernel here)

$$\Omega[f] = \frac{1}{2} \|f\|_{\mathcal{H}}^2$$

- Kernel Regularizer
  - Convex combination
  - Wishart regularizer
  - Inverse norm regularizer

$$\Gamma[\mathcal{H}] = \operatorname{tr} K$$

Learning the Kernel

- Convex optimization problem

minimize

$$R_{\text{emp}}[f] = \frac{1}{m} \sum_{i=1}^{m} l(y_i, f(x_i))$$

$$+$$

$$\Omega[f] = \frac{1}{2} \|f\|_{\mathcal{H}}^2$$

$$+$$

$$\Gamma[\mathcal{H}] = \operatorname{tr} K$$

Learning the Kernel

- Convex optimization problem

$$R_{\mathrm{emp}}[f] = \frac{1}{m} \sum_{i=1}^{m} l(y_i, f(x_i))$$

+

$$\Omega[f] = \frac{1}{2} \|f\|_{\mathcal{H}}^2$$

+

$$\Gamma[\mathcal{H}] = \operatorname{tr} K$$

$$\Omega'[f] := \min_{\mathcal{H}} \left[\frac{1}{2} \|f\|_{\mathcal{H}}^2 + \lambda \operatorname{tr} K \right]$$

Banach space norm

Learning the Kernel

- Convex optimization problem
  - Switch to nontrivial Banach space norms
  - Computation efficient via underlying RKHS

$$R_{\text{emp}}[f] = \frac{1}{m}\sum_{i=1}^{m} l(y_i, f(x_i))$$

$$\Omega[f] = \frac{1}{2}\|f\|_{\mathcal{H}}^2$$

+

$$\Omega'[f] := \min_{\mathcal{H}} \left[\frac{1}{2}\|f\|_{\mathcal{H}}^2 + \lambda \operatorname{tr} K\right]$$

$$\Gamma[\mathcal{H}] = \operatorname{tr} K$$

Banach space norm

Multitask Learning

Multitask Learning

- Many tasks, use them jointly to learn a kernel

$$\sum_j R^j_{\mathrm{emp}}[f_j] + \underbrace{\sum_j \lambda_j \Omega[f_j] + \Gamma[\mathcal{H}]}_{\Omega'[f_1, \ldots, f_t]}$$

Multitask Learning

- Many tasks, use them jointly to learn a kernel

$$\sum_j R^j_{\mathrm{emp}}[f_j] + \sum_j \lambda_j \Omega[f_j] \quad + \quad \Gamma[\mathcal{H}]$$

$$\Omega'[f_1, \ldots, f_t]$$

- Argyriou et al. 2008, 2009

$$\underset{K}{\mathrm{minimize}} \sum_j f_j^\top K^{-1} f_j \text{ subject to } K \succeq 0 \text{ and } \operatorname{tr} K \leq 1$$

Multitask Learning

- Many tasks, use them jointly to learn a kernel

$$\sum_j R^j_{\text{emp}}[f_j] + \sum_j \lambda_j \Omega[f_j] + \Gamma[\mathcal{H}]$$

$$\Omega'[f_1, \ldots, f_t]$$

- Argyriou et al. 2008, 2009

$$\underset{K}{\text{minimize}} \sum_j f_j^\top K^{-1} f_j \text{ subject to } K \succeq 0 \text{ and } \operatorname{tr} K \leq 1$$

$$\|[f_1, \ldots f_t]\|_{\text{KyFan}}$$

Chakrabarty et al. 2010

Multitask Learning

- Many tasks, use them jointly to learn a kernel

$$\sum_j R^j_{\mathrm{emp}}[f_j] + \sum_j \lambda_j \Omega[f_j] + \Gamma[\mathcal{H}]$$

$$\Omega'[f_1, \ldots, f_t]$$

- Argyriou et al. 2008, 2009

$$\underset{K}{\operatorname{minimize}} \sum_j f_j^\top K^{-1} f_j \text{ subject to } K \succeq 0 \text{ and } \operatorname{tr} K \leq 1$$

$$\|[f_1, \ldots f_t]\|_{\mathrm{KyFan}}$$

Chakrabarty et al. 2010

This is the nuclear norm of collaborative filtering!

From CF to MTL

From CF to MTL

- Collaborative Filtering (Srebro et al., 2005+)

$$\sum_{(i,j)\in S} l(Y_{ij}, F_{ij}) + \lambda \|F\|_{\text{KyFan}}$$

equivalent formulation

$$\sum_{(i,j)\in S} l(U_i^\top V_j, F_{ij}) + \frac{\lambda}{2}\left[\|U\|^2_{\text{Frob}} + \|V\|^2_{\text{Frob}}\right]$$

From CF to MTL

- Collaborative Filtering (Srebro et al., 2005+)

$$\sum_{(i,j)\in S} l(Y_{ij}, F_{ij}) + \lambda \|F\|_{\mathrm{KyFan}}$$

equivalent formulation

$$\sum_{(i,j)\in S} l(U_i^\top V_j, F_{ij}) + \frac{\lambda}{2}\left[\|U\|_{\mathrm{Frob}}^2 + \|V\|_{\mathrm{Frob}}^2\right]$$

- Multitask Learning

$$\sum_j R_{\mathrm{emp}}[f^j] + \lambda \|[f_1, \ldots, f_t]\|_{\mathrm{KyFan}}$$

equivalent formulation (latent factor model)

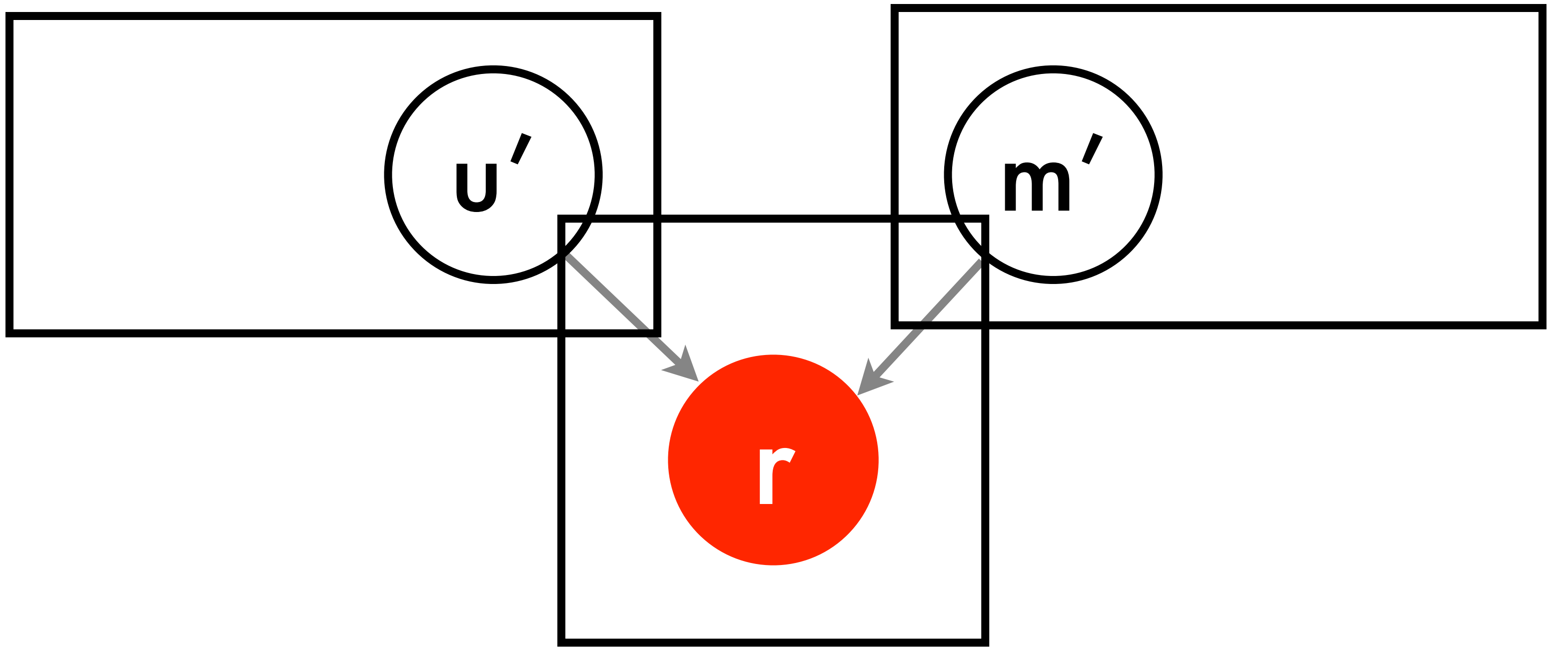
$$\sum_j R_{\mathrm{emp}}^j[U_j^\top f] + \frac{\lambda}{2}\left[\|U\|_{\mathrm{Frob}}^2 + \|f\|_{\mathrm{Frob}}^2\right]$$

Factorization

Factorization

Collaborative Filtering

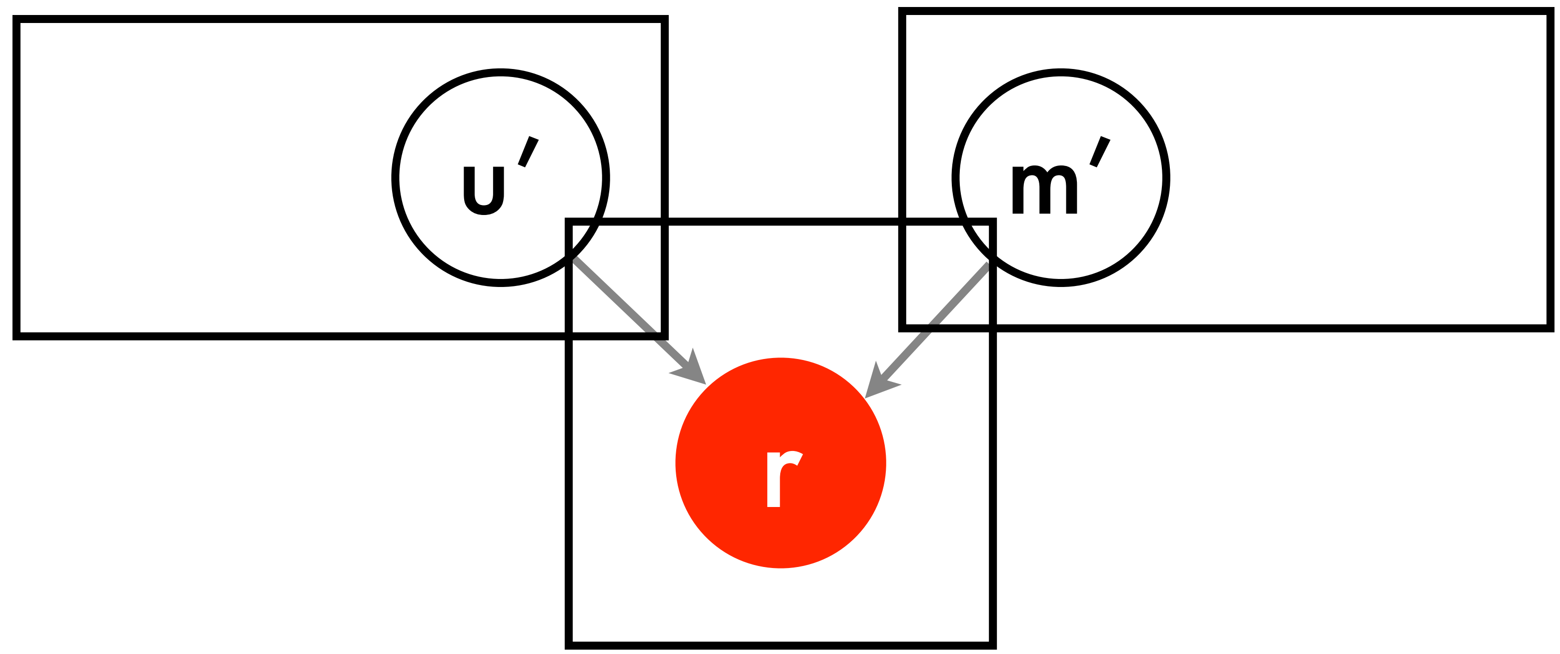
$$f(i,j) = u_i^\top m_j$$



Factorization

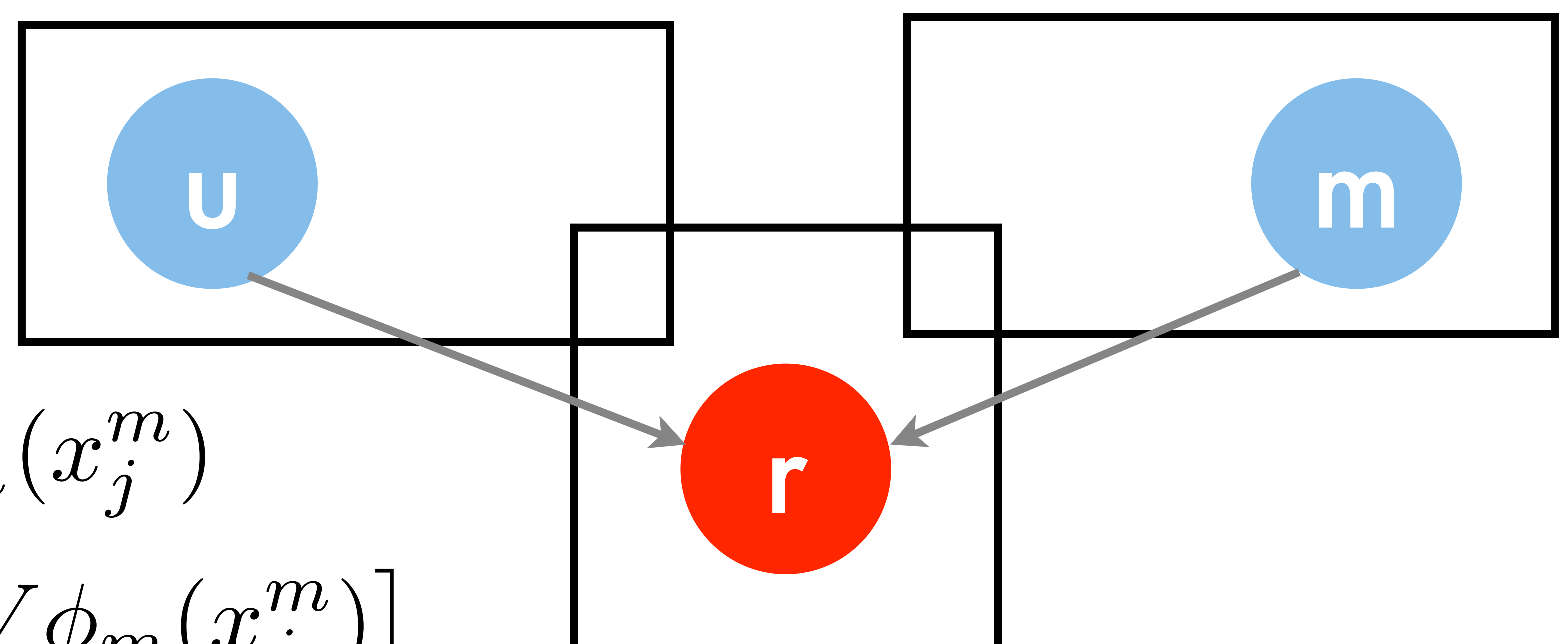
Collaborative Filtering

$$f(i,j) = u_i^\top m_j$$



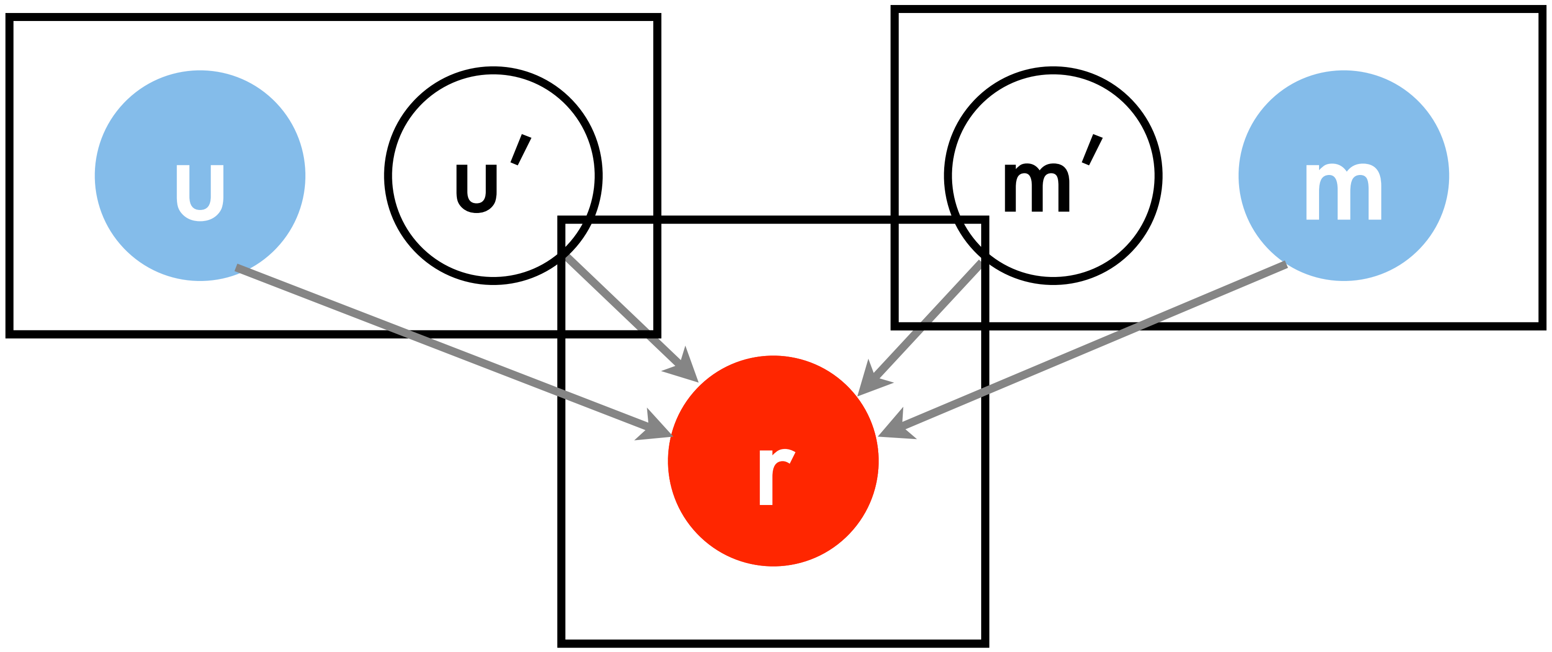
Feature based filtering (e.g. ranking)

$$f(i,j) = \phi_u(x_i^u)^\top M \phi_m(x_j^m)$$
$$= \left[U\phi_u(x_i^u)\right]^\top \left[V\phi_m(x_j^m)\right]$$



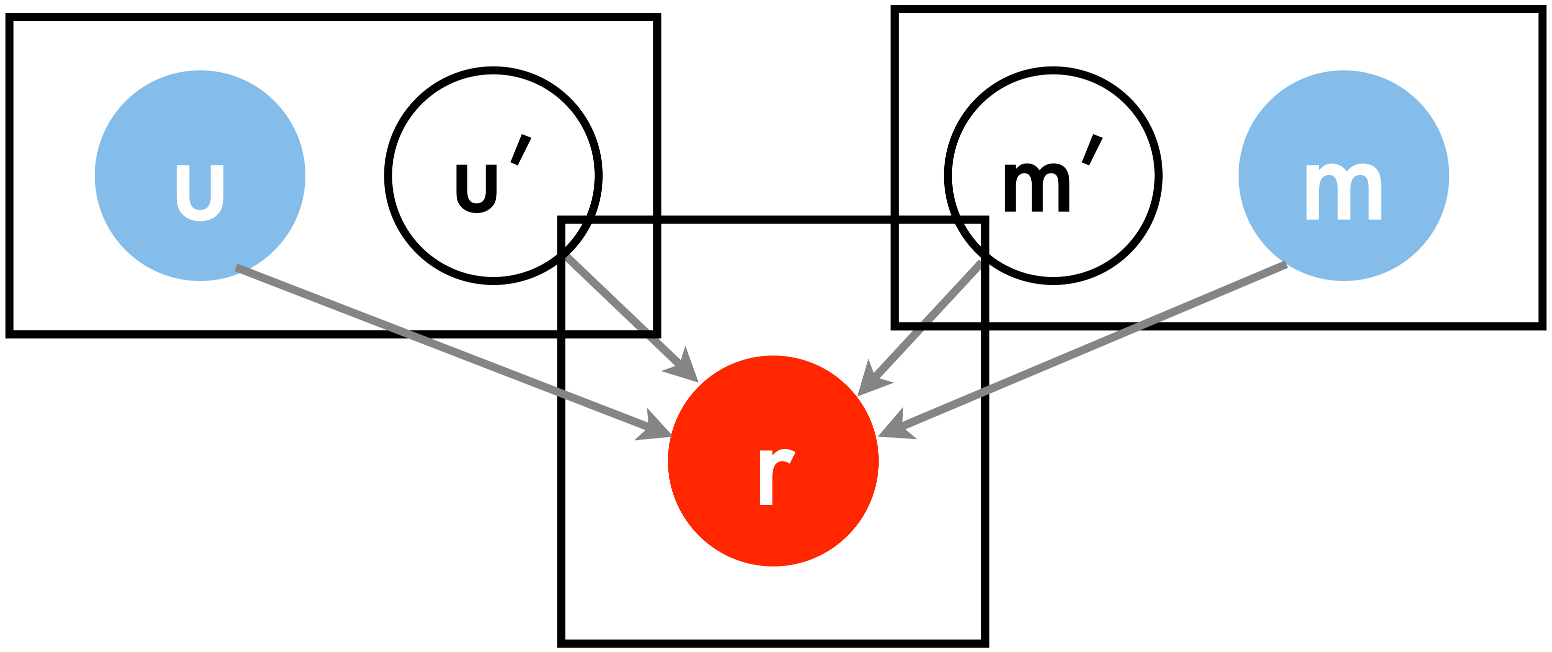
Factorization

Joint Model



Factorization

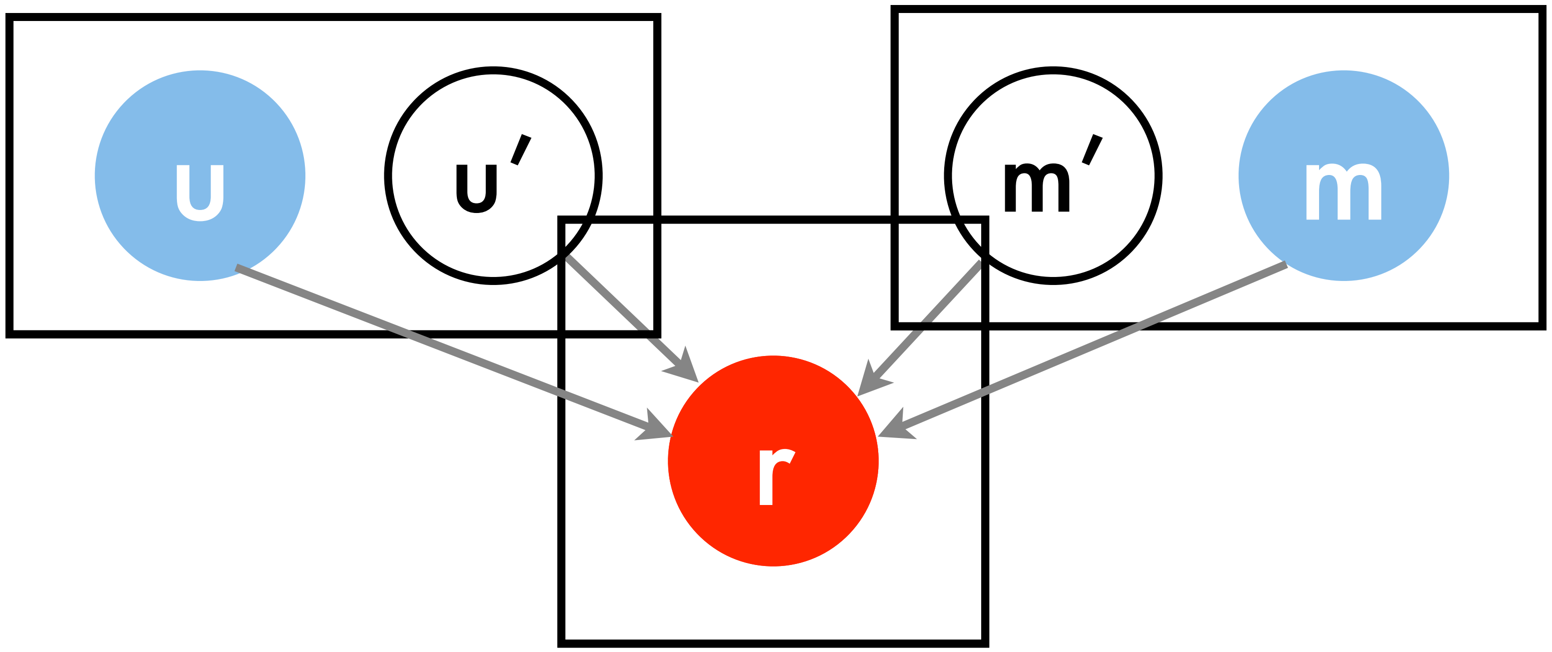
Joint Model



$$f(i,j) = \left[U\phi_u(x_i^u) + u_i + b_u\right]^\top \left[V\phi_m(x_j^m) + m_j + b_m\right]$$

Factorization

Joint Model

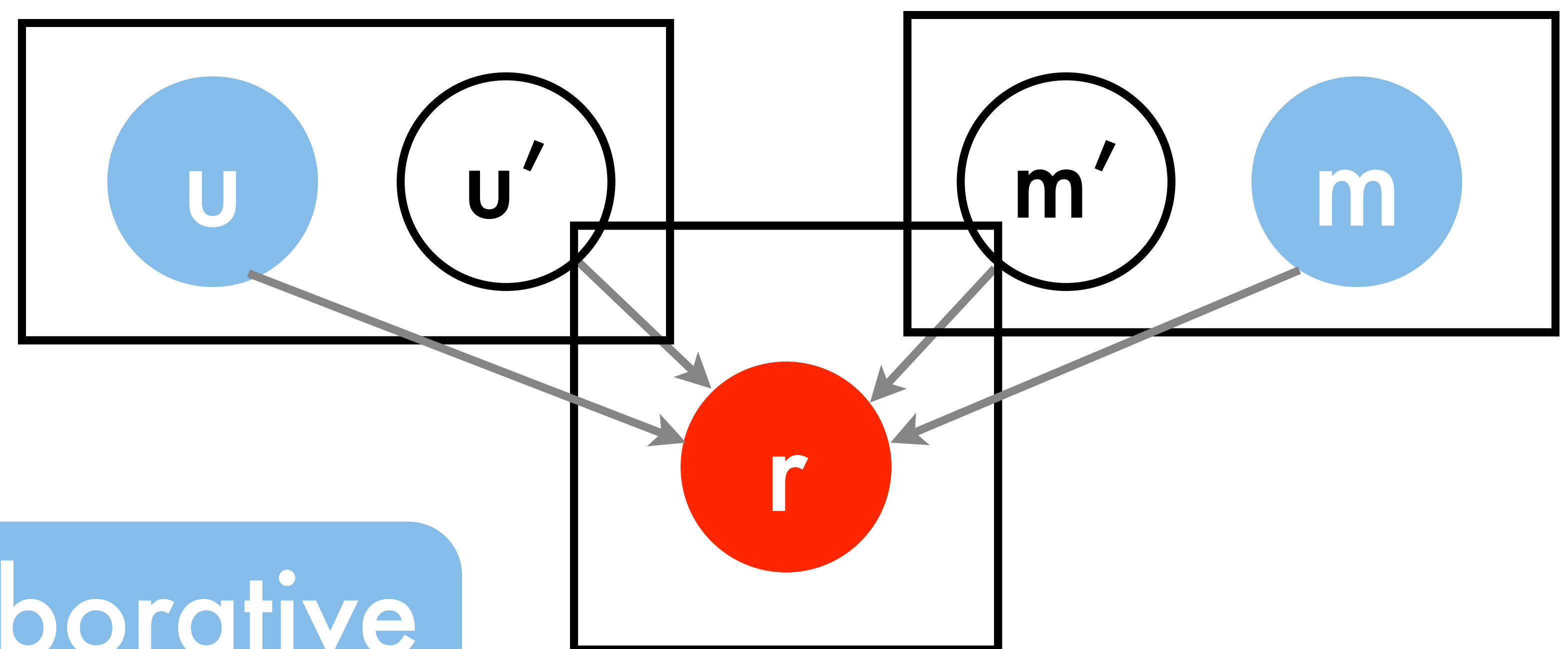


multitask

$$f(i,j) = \left[U\phi_u(x_i^u) + u_i + b_u\right]^\top \left[V\phi_m(x_j^m) + m_j + b_m\right]$$

Factorization

Joint Model



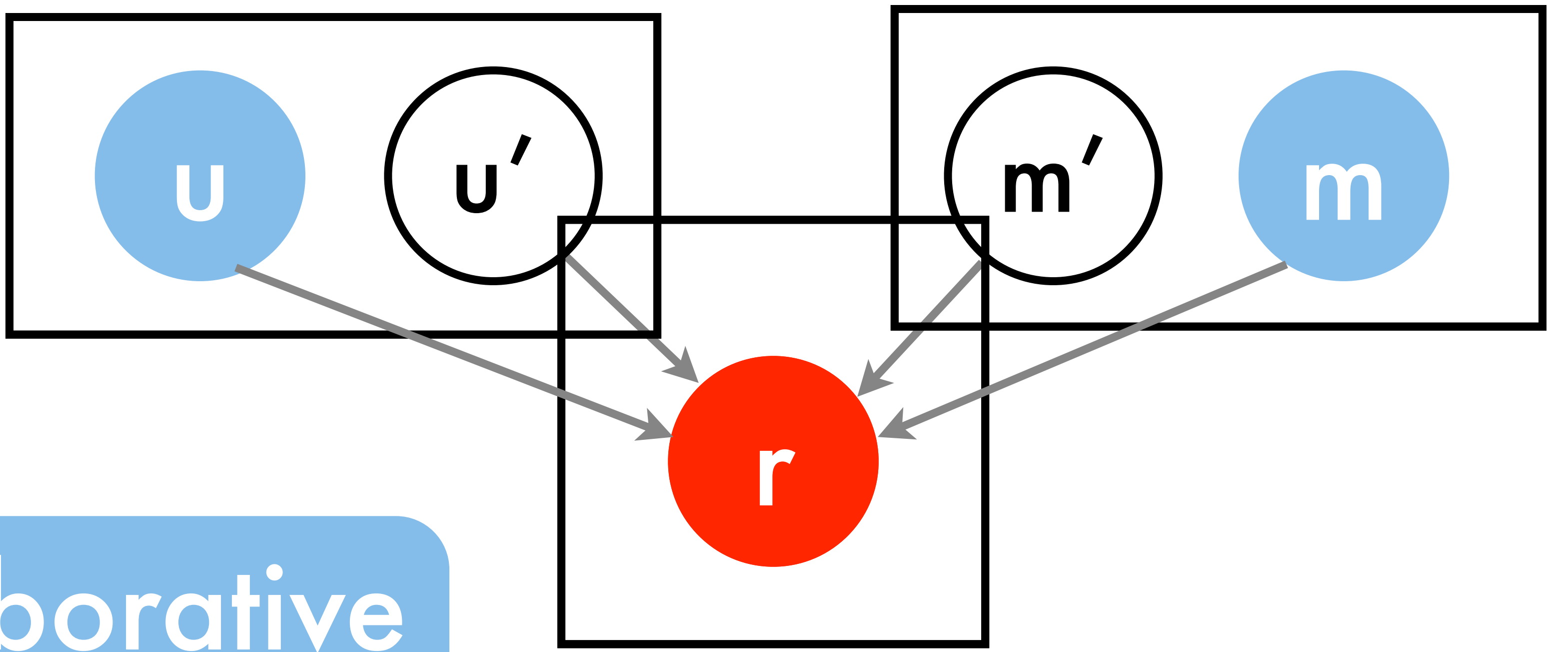
collaborative filtering

multitask

$$f(i,j) = \left[U\phi_u(x_i^u) + u_i + b_u\right]^\top \left[V\phi_m(x_j^m) + m_j + b_m\right]$$

Factorization

Joint Model



collaborative filtering

multitask

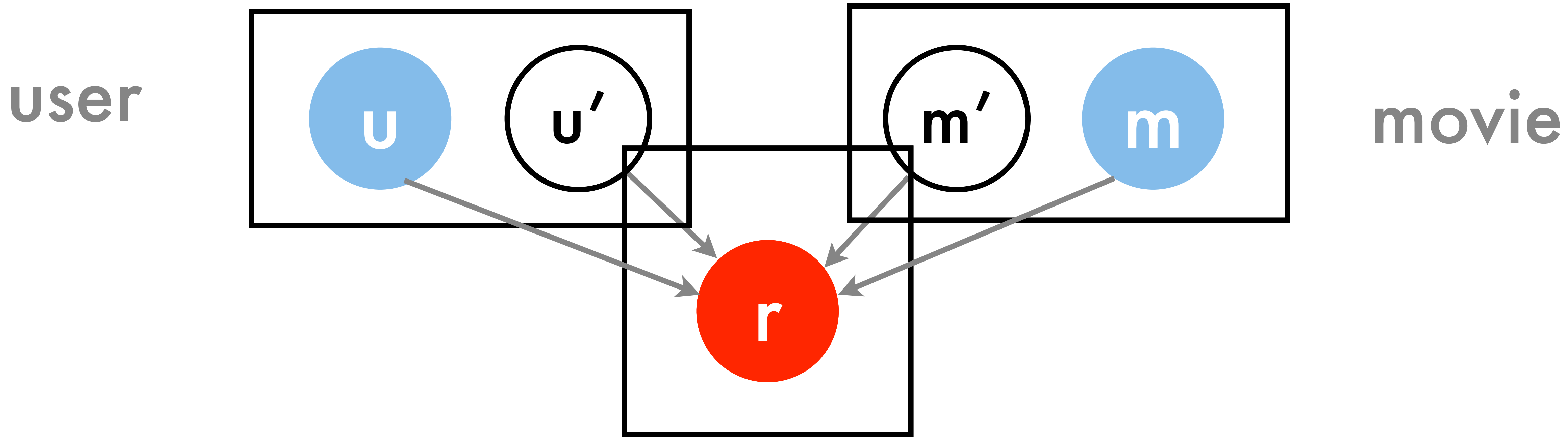
$$f(i,j) = \left[U\phi_u(x_i^u) + u_i + b_u\right]^\top \left[V\phi_m(x_j^m) + m_j + b_m\right]$$

bias

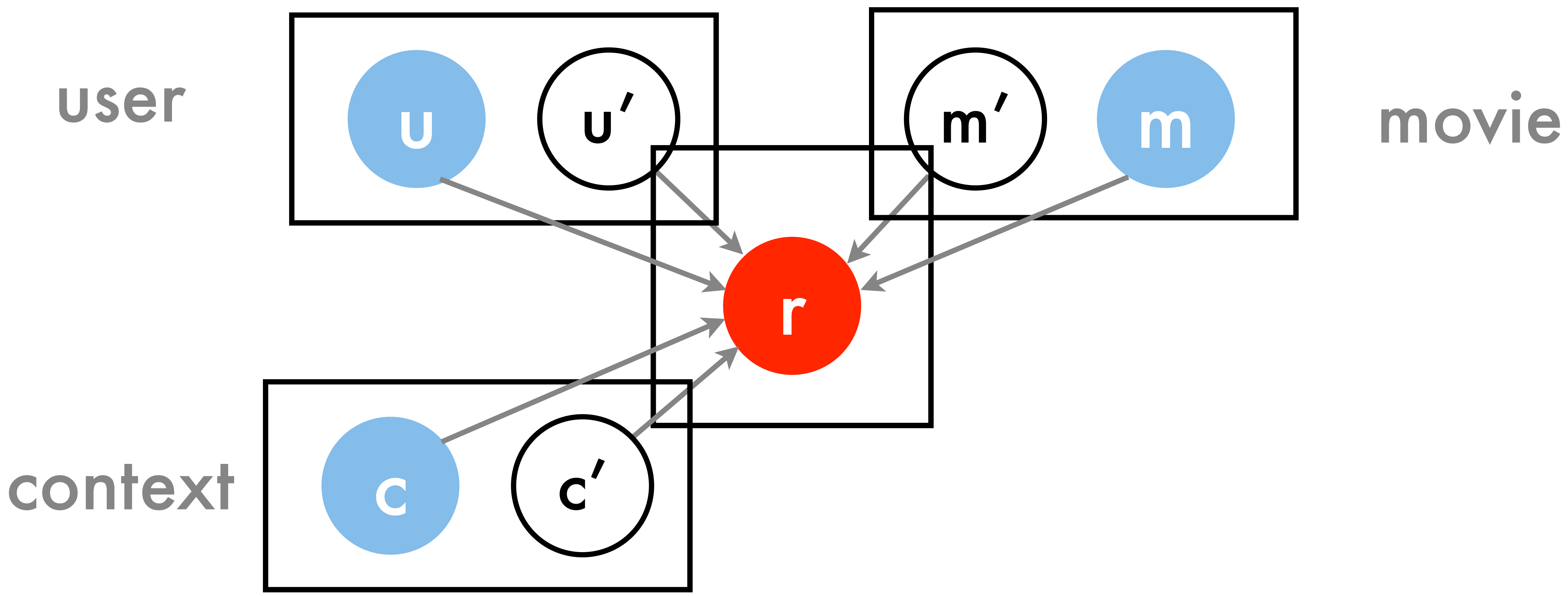
Applications

- Collaborative filtering
  (features, IDs)
- Ranking
  (queries, webpages)
- Multitask learning
  (many categories for webpages)
- Dataset integration
  (different ontologies on same domain)
- Time-series prediction
  (stock values are correlated, balance sheet)

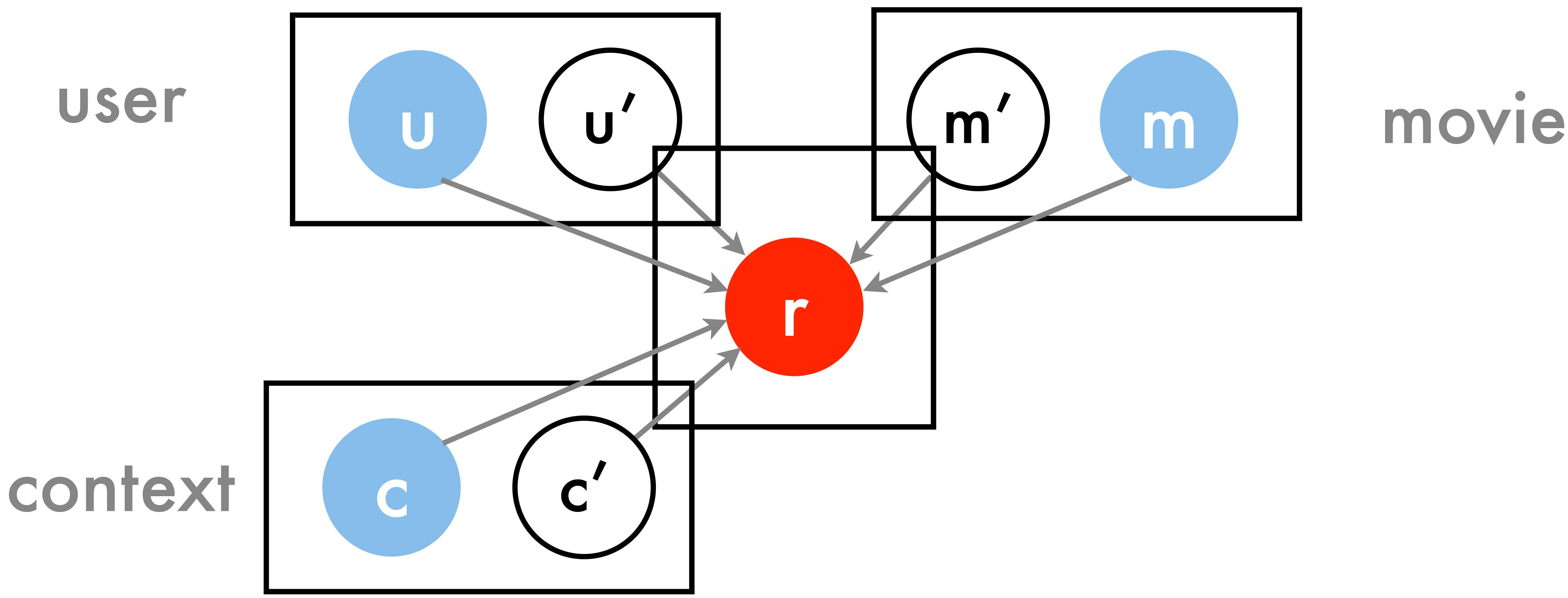
Tensor Factorization



Tensor Factorization

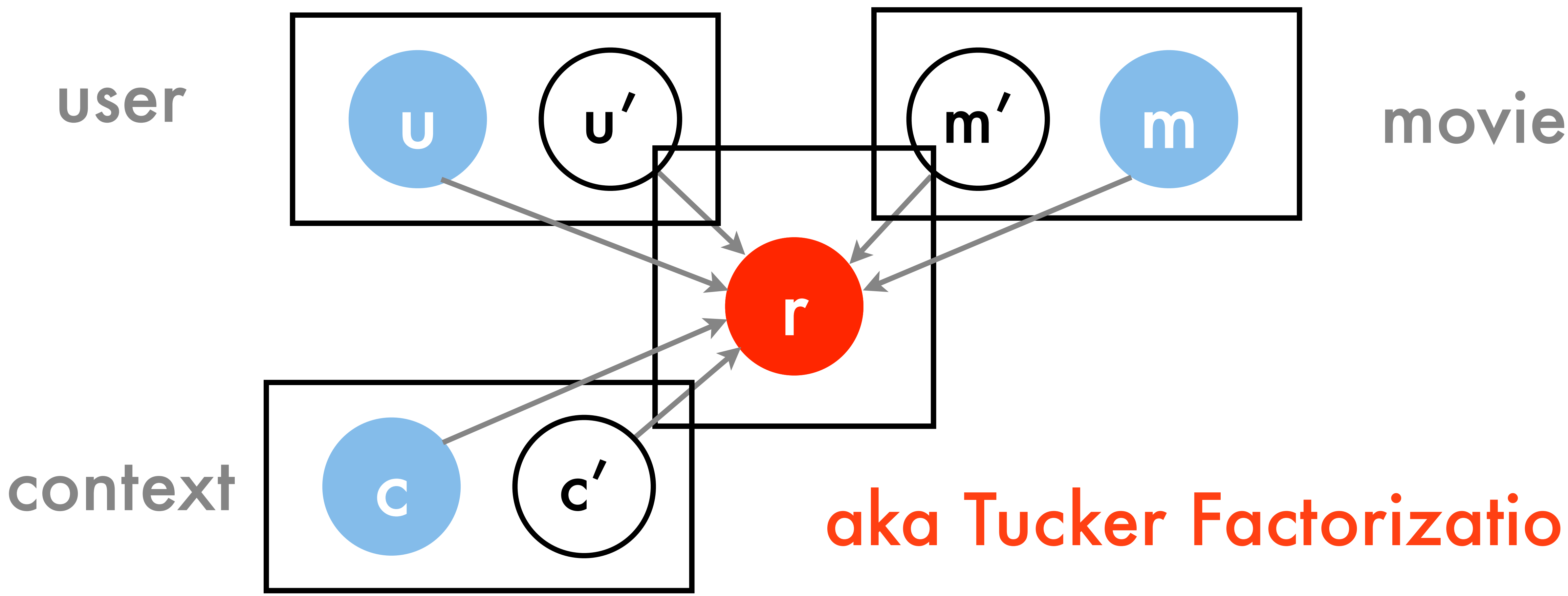


Tensor Factorization



$$f(i,j,k) = \left[U\phi_u(x_i^u) + u_i + b_u\right]^\top \left[V\phi_m(x_j^m) + m_j + b_m\right] +$$
$$\left[U\phi_u(x_i^u) + u_i + b_u\right]^\top \left[W\phi_c(x_k^c) + c_k + b_c\right]$$

Tensor Factorization



aka Tucker Factorization

$$f(i,j,k) = \left[U\phi_u(x_i^u) + u_i + b_u\right]^\top \left[V\phi_m(x_j^m) + m_j + b_m\right] +$$
$$\left[U\phi_u(x_i^u) + u_i + b_u\right]^\top \left[W\phi_c(x_k^c) + c_k + b_c\right]$$

Optimization

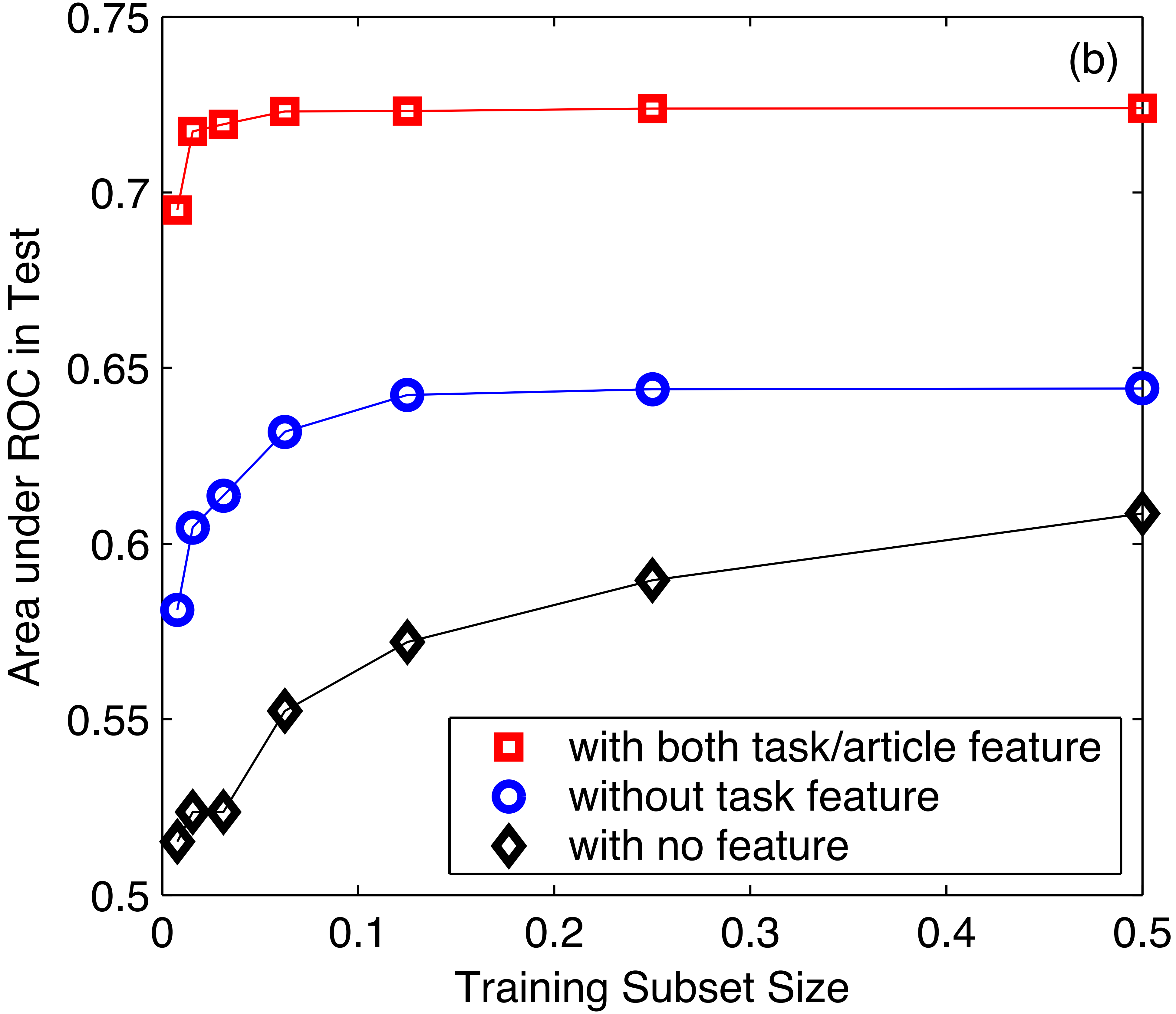
- Stochastic gradient descent on (i,j) pairs one at a time ... idiot-proof simple. But nonconvex.

User profiles

Datasets

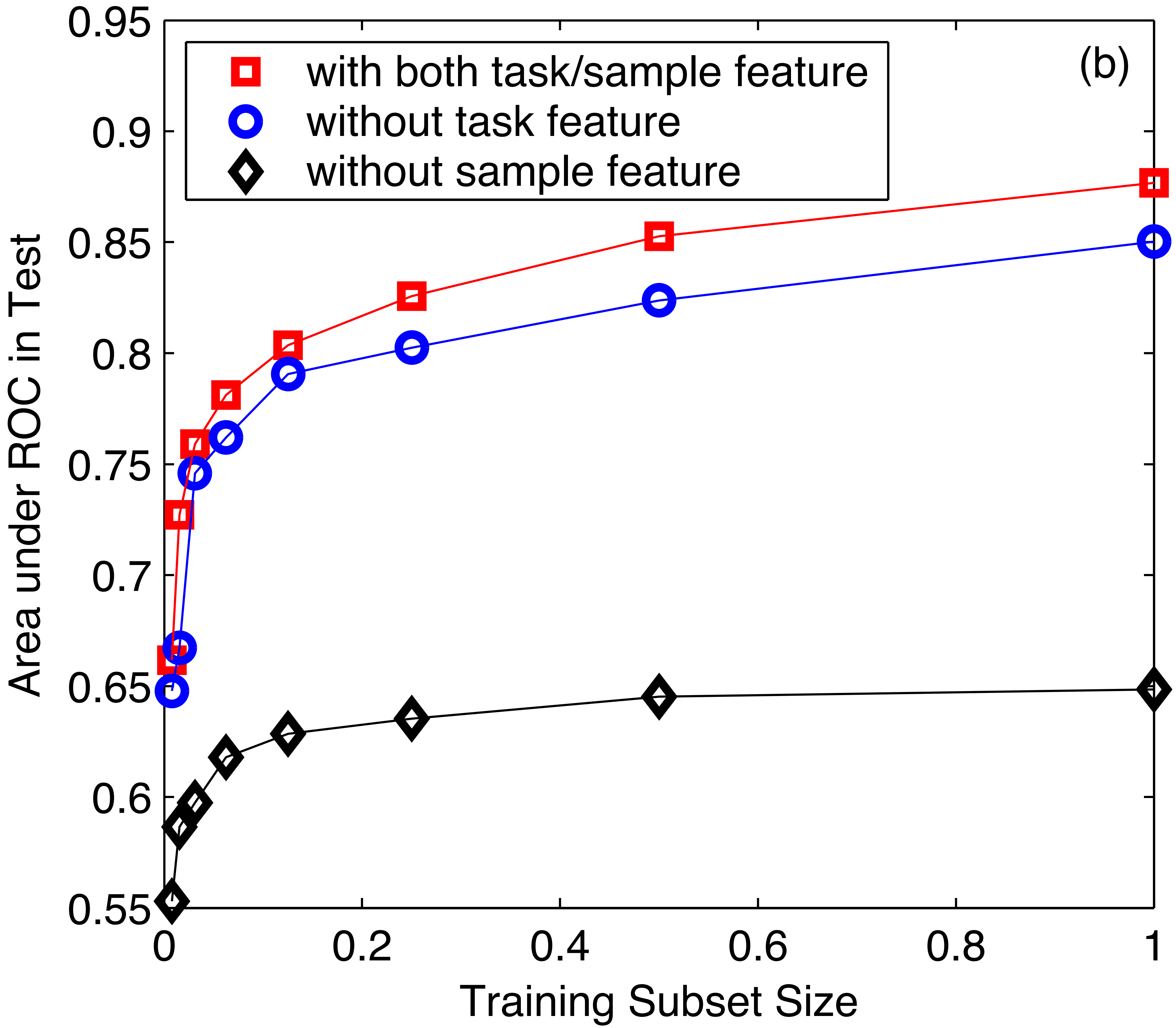
- Click prediction
  - 3M users (d=1000), 150k documents (d=100)
  - 43M (user, document) pairs
- Page classification
  - 2.8M documents, 82 classification problems
  - Trivial problem features (e.g. "SpamOrNot-USMarket")

Click model



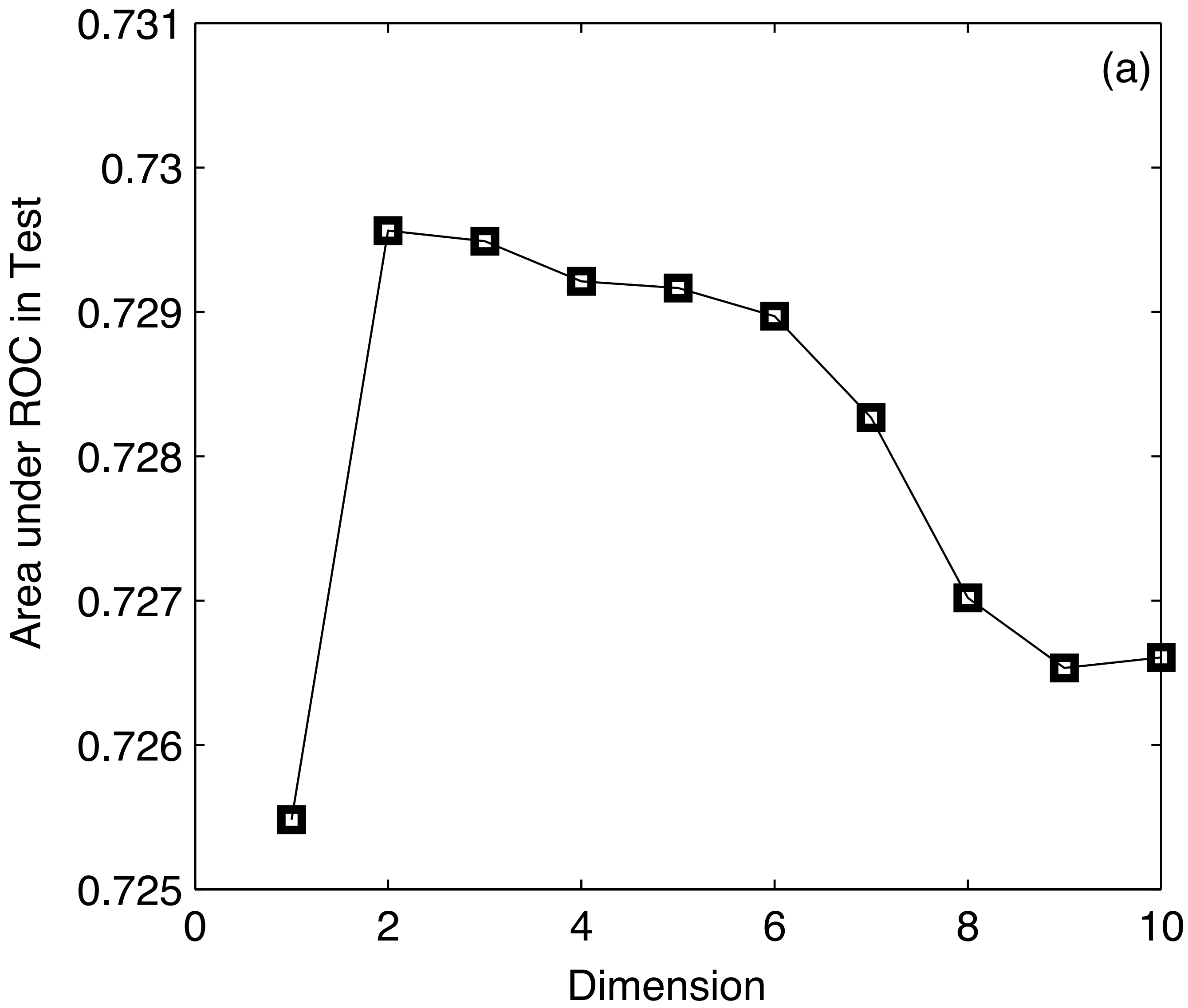
0.75
0.7
0.65
0.6
0.55
0.5
0
0.1
0.2
0.3
0.4
0.5
(b)
Area under ROC in Test
Training Subset Size
with both task/article feature
without task feature
with no feature

Page classification

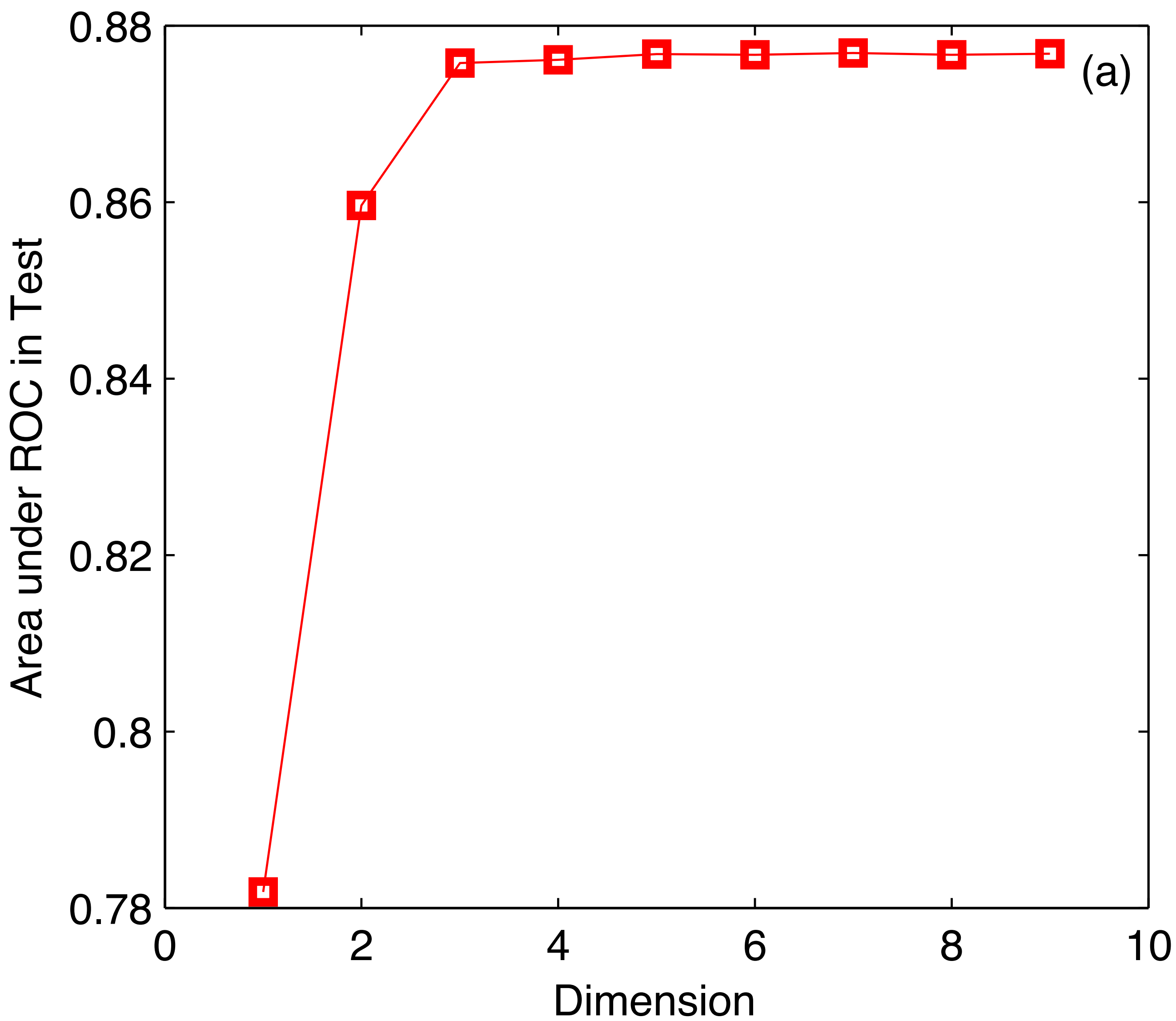


Effect of dimensionality

clicks



classification problems



Summary

Summary

- Three problems - one solution
  - Multitask learning
  - Collaborative filtering
  - Learning the kernel

Summary

- Three problems - one solution
  - Multitask learning
  - Collaborative filtering
  - Learning the kernel
- Factorization approach
  - Applications
  - Extensions (Tucker factors, data integration)
  - Optimization

Summary

- Three problems - one solution
  - Multitask learning
  - Collaborative filtering
  - Learning the kernel
- Factorization approach
  - Applications
  - Extensions (Tucker factors, data integration)
  - Optimization
- Experiments
  - User profiles
  - Webpage categorization with side information