Simultaneous (Co)-clustering and Modeling for Large Scale Data Mining

Joydeep Ghosh
Schlumberger Centennial Chaired Professor
Electrical and Computer Engineering, UT-Austin

Joint work with Meghana Deodhar (ECE)

Background

- Difficult classification/regression problems may involve a heterogeneous population

- Divide and conquer approach
  - Partition the population and model each partition separately

- Advantages (vs. "ensemble approaches")
  - Learning models on more homogenous data
  - Improves accuracy
  - Simpler, more interpretable models

Motivation

- Traditionally partitioning is *a priori*
  - Domain knowledge
  - Clustering algorithm

- … but, *a priori* partitioning may be suboptimal

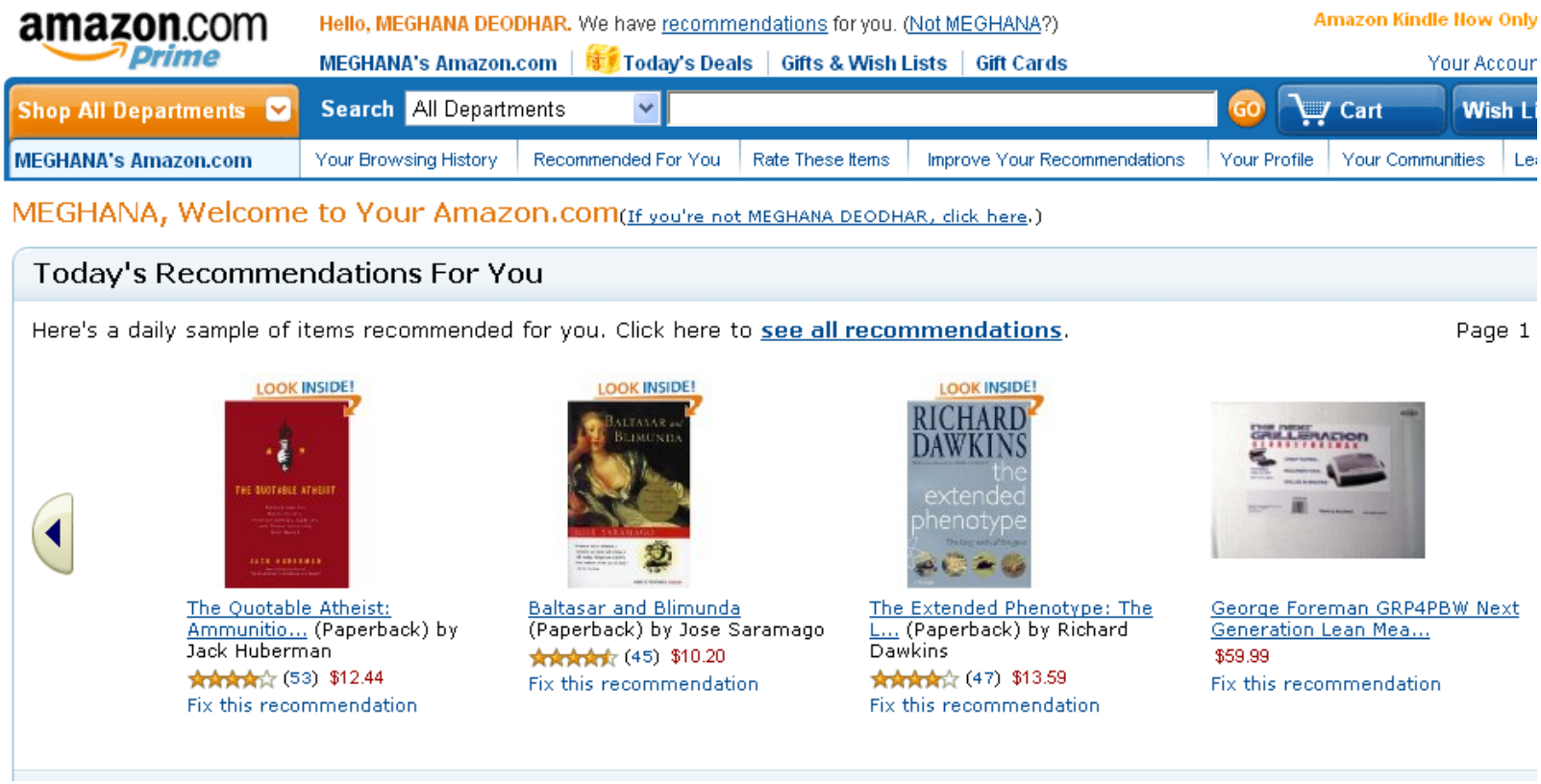
- Solution: Interleaving partitioning and construction of prediction models

Approaches to Interactive Decomposition

- Hard partitioning of input space
  - Change-point detection: Segmenting time series + fitting model per segment
- Soft partitioning of input space for regression
  - Mixture of Experts
    - Hierarchical versions
- Output space partitioning for modeling large number of classes

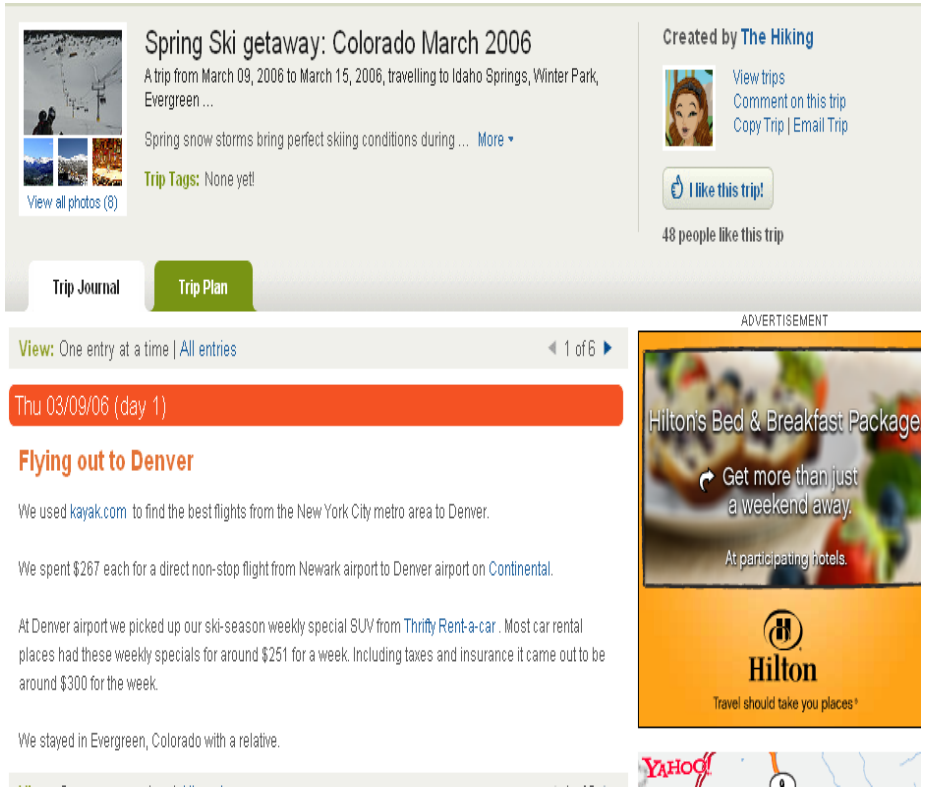
Dyadic Data Applications

- Recommender system: customers, products, ratings

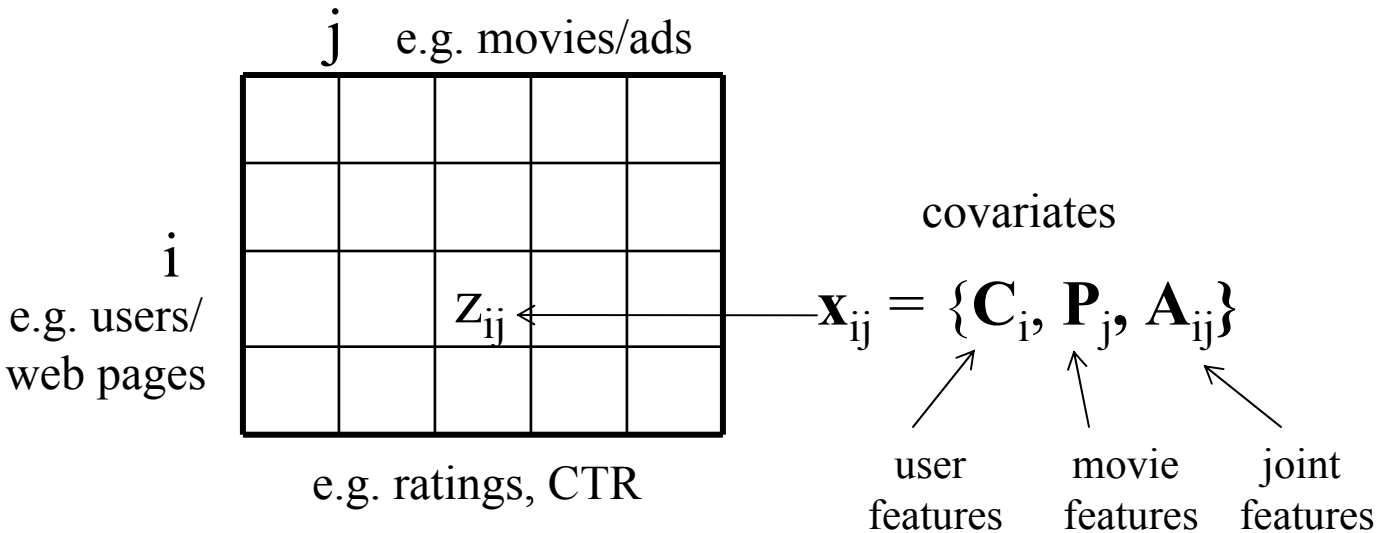


Several Other Applications

- Search advertizing
  - Web pages, ads, click-through rates
  - Users, pages, ads, click-though rates
- Web search
  - Queries, web pages, relevance scores
- Ecological studies
  - Species, sites, presence/absence
- ....



Modeling Large-Scale Dyadic Data



- Can we simultaneously partition (along multiple modes) and predict?
  - Deodhar and Ghosh, KDD07, KDD09
  - Agarwal and Merugu, KDD07

Example – Recommender System

- Problem: predict customer purchase decisions

Product attributes

price

market share

advertisements

products

customers

-1	1	-1	1	1
?	1	1	?	-1
1	1	?	-1	1
?	-1	1	-1	1

age	income	gender	# kids

Customer attributes

Possible Approaches

- Collaborative Filtering
- Classification
  - Logistic regression
- Co-clustering or Biclustering
  - Bregman Co-clustering

Collaborative Filtering

- Collaborative Filtering – technique for reducing information overload
  - Improves access to relevant products and information
  - e.g. Recommender systems that suggest books, films, music, etc.

- Predict how well a user will like an unrated item
  - Based on preferences of a community

- Preference judgments can be explicit or implicit
  - Explicit – numerical ratings for each item
  - Implicit – extracted from purchase records or web logs

Collaborative Filtering

products

customers

-1	1	-1	1	1
?	1	1	?	-1
1	1	?	-1	1
?	-1	1	-1	1

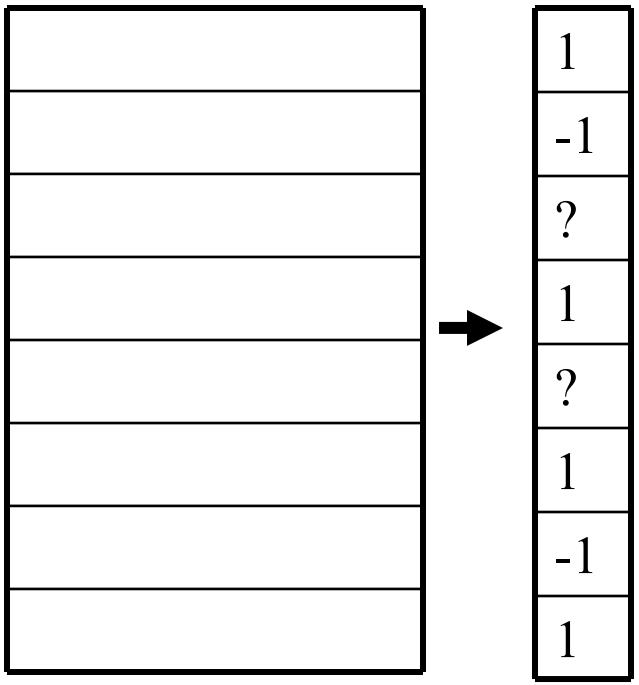
- Find a neighborhood of similar customers
  - Based on known choices
- Predict current purchase decision using preference of neighborhood
- Ignores customer/product attributes

Low rank matrix factorization methods
also focus on the "Z" matrix but exploit more global properties

Single Classification Model

Feature vector
- customer, product attributes

Target variable
- Matrix entries



- May not be adequate to capture heterogeneity
- Does not use neighborhood information
  - Similarity of customers/products

Co-clustering

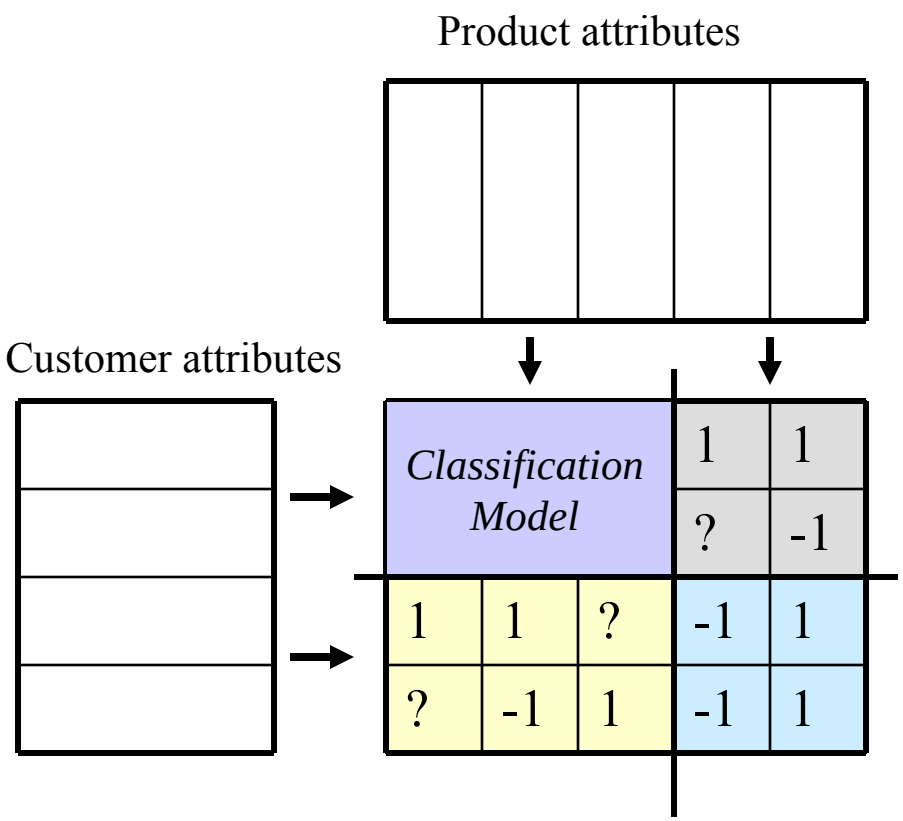
Product clusters

Customer clusters

-1	1	-1	1	1
?	1	1	?	-1
1	1	?	-1	1
?	-1	1	-1	1

- Identifies neighborhood of similar customers and products
- Predicts unknown choice using known entries within the co-cluster
- Ignores customer and product attributes

Simultaneous Co-clustering and Classification



- Exploits neighborhood information **and** attributes
- Iteratively clusters along both axes and fits predictive model in each co-cluster
- Common framework for solving classification and regression problems

Problem Definition (Regression)

- $Z_{m \times n}$ matrix of "customers" and "products"
  - Matrix entries are real numbers (e.g. ratings)

- Assumption: matrix entry is a linear combination of customer and product attributes ($\mathbf{C}_i$ and $\mathbf{P}_j$) + noise

$$z_{ij} = \boldsymbol{\beta}^T \mathbf{x}_{ij}$$

  - Model parameters $\boldsymbol{\beta}^T = [\beta_0, \boldsymbol{\beta}_c^T, \boldsymbol{\beta}_p^T]$
  - Attribute vector $\mathbf{x}_{ij}^T = [1, \mathbf{C}_i^T, \mathbf{P}_j^T]$

- Aim: Simultaneously cluster customers and products into a grid of co-clusters, such that the values within each co-cluster are predicted by the same regression model

Regression Example

p

0	4	3	1	3	2

c

1	4	5	10	9	8	8
7	33	17	23.2	36	39	19
4	21	11	17.1	26	24	15
2	3.1	4.5	8.5	6.6	4.5	6.1
0	1	2.2	5	5.2	4	4
2	9.5	4	13.5	14	13	10
3	16	9.1	14.9	19	17.5	12
3	4	5	8.7	8	6.8	6.8

Regression Example

- After rearranging rows and cols.

p

	3	2	1	4	0	3

	2	8.5	6.1	4.5	6.6	3.1	4.5
	0	5	4	2.2	5.2	1	4
	3	8.7	6.8	5	8	4	6.8
c	1	10	8	5	9	4	8
	2	13.5	10	4	14	9.5	13
	7	23.2	19	17	36	33	39
	4	17.1	15	11	26	21	24
	3	14.9	12	9.1	19	16	17.5

Regression Example

c + 2p

p

3	2	1	4	0	3

c

2	**8**	**6**	**4**	6.6	3.1	4.5
0	**6**	**4**	**2**	5.2	1	4
3	**9**	**7**	**5**	8	4	6.8
1	10	8	5	9	4	8
2	13.5	10	4	14	9.5	13
7	23.2	19	17	36	33	39
4	17.1	15	11	26	21	24
3	14.9	12	9.1	19	16	17.5

Regression Example

c + 2p

1 + c + p

p

3	2	1	4	0	3

c						
2	8.5	6.1	4.5	6.6	3.1	4.5
0	5	4	2.2	5.2	1	4
3	8.7	6.8	5	8	4	6.8
1	10	8	5	9	4	8
2	13.5	10	4	14	9.5	13
7	23.2	19	17	36	33	39
4	17.1	15	11	26	21	24
3	14.9	12	9.1	19	16	17.5

2c + 3p

5c + p

Reconstruction Errors

Reconstructed with a single linear model

$z = 1.2 + 3.6c + 1.5p$

-2	-2.9	-3	-5.4	-2.9	-6
1.7	2.2	1.9	0.4	2.2	0.7
-5.3	-5.8	-6.1	-7.5	-5.6	-7.2
3.1	2.6	1.1	0.6	1.6	1.1
3	1	-3.5	2	3.5	2.5
-5.2	-7.9	-8.4	6.1	9.1	10.6
-0.5	-1.1	-3.6	6.9	7.8	6.4
0.9	-0.6	-2	3.5	6.4	3.5

MSE = 21.8

Reconstructed with simultaneous co-clustering and regression

-1.1	-0.8	0.2	-0.7	-0.7	-1.9
-1.6	0	1	2.3	2.5	2.2
-4.3	-3.5	-2.6	-8.6	-8.2	-8.7
1.3	2	1.7	1.5	0.9	1.6
2.5	1.8	-1.4	2	1.8	2
1.2	0	0.9	1	2	5
-0.5	0	-1	0	-2	-1
0.9	1	0.8	2	3	1.5

MSE = 7.9

Note: Reduced Parameter approaches available

Objective Function

- ρ: mapping from m rows to **k row clusters**
- γ: mapping from n columns to ***l* column clusters**
  - **Total of k * *l* regression models.**
- Weight (w_{uv}) associated with each matrix entry
  - 1 - known entry, 0 - missing
- Find co-clustering (ρ, γ) and models ($\boldsymbol{\beta}$'s) that minimize the total squared error

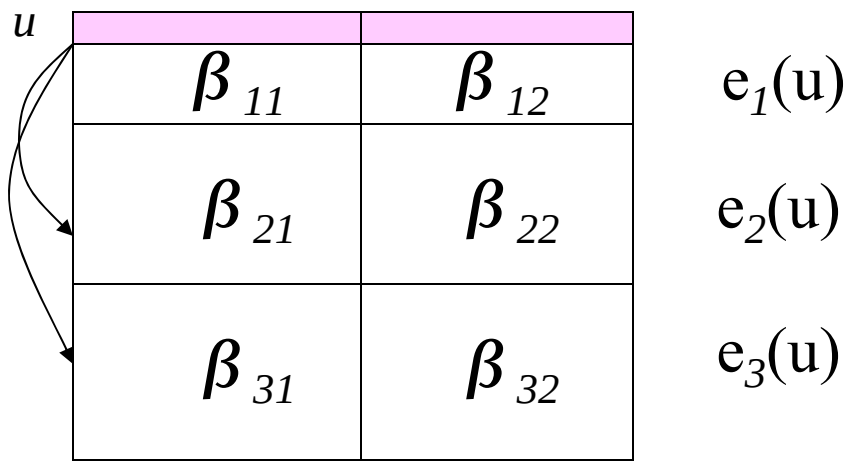
$$\sum_{u,v} w_{uv}(z_{uv} - \hat{z}_{uv})^2$$

$$\hat{z}_{uv} = \boldsymbol{\beta}_{\rho(u)\gamma(v)}{}^T \mathbf{x}_{uv}$$

Row and Column Cluster Updates

- Objective function is a sum of row/column errors
  - Assign each row to a row cluster that minimizes the row error
  - Row cluster assignment for row u

$$\rho^{new}(u) = \arg\min_g \sum_{v=1}^{n} w_{uv}(z_{uv} - \hat{z}_{uv})^2$$



SCOAL Meta -Algorithm

Input: Data: Z, Weights: W, Attributes: C, P
Output: co-clustering (ρ, γ), models { $\boldsymbol{\beta}$ s}

Initialize ρ, γ
Iterate until convergence

- Re-estimate model for each co-cluster
- Re-estimate the co-clusters
  - Update row clusters – assign each row to closest row cluster
  - Update col clusters – assign each col to closest col cluster

Return ρ, γ, { $\boldsymbol{\beta}$ s}

Guaranteed to converge to locally optimal solution

Simultaneous Co-clustering and Classification

- Elements of Z are class labels (2 class problem)
- Logistic regression model relating attributes to the class label
  - Log odds modeled as a linear combination of the attributes (= $\boldsymbol{\beta}^{\mathrm{T}}\mathbf{x}_{ij}$)
- Find co-clustering (ρ, γ) and models ($\boldsymbol{\beta}$'s) that minimize the total log loss (-ve log likelihood)

$$\sum_{u,v} w_{uv} \ln(1+\exp(-z_{uv}\boldsymbol{\beta}^{T}_{\rho(u)\gamma(v)}\mathbf{x}_{uv}))$$

Model Selection (M-SCOAL)

- Cross validation procedure to select k and l
  - Minimize prediction error on the validation set
- Top down bisecting greedy algorithm

Run SCOAL with k=1, l=1

Repeat

1. Split row/column cluster with highest error
2. Initialize SCOAL with current partitioning
3. Accept split if validation error reduces

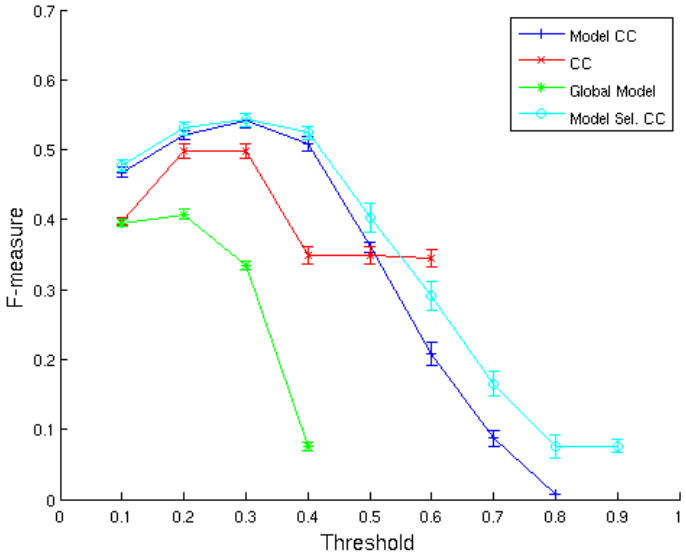
until no change in k and l

- Gives better local minimum
- Fast convergence

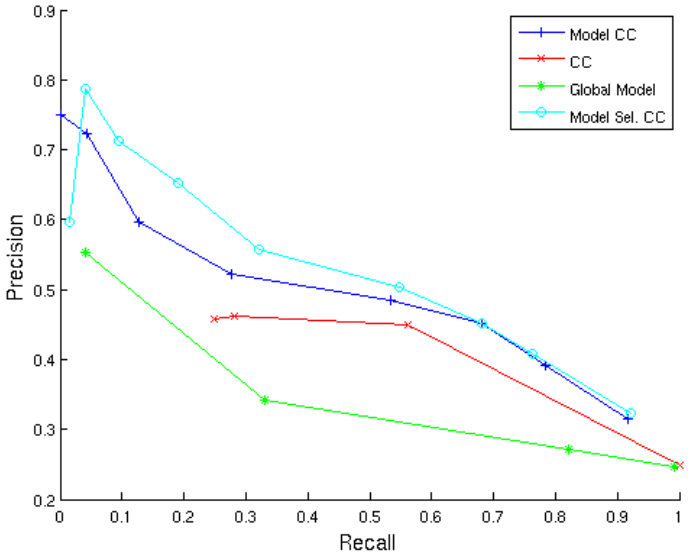
Recommender System Application

- Predict unknown course choices of masters students
  - 32 courses, 326 students
  - Student attributes: career aspiration, undergraduate degree
  - Course attributes: department, evaluation score

F – measure plot



Precision-Recall curve



ERIM Marketing Dataset

- Household panel data collected by A.C. Nielsen
- 1714 customers, 121 products from 6 product categories (ketchup, sugar, etc.)
- Customer-product matrix cell values = # units purchased
  - Household Attributes – income, # residents, male head employed, female head employed, total visits, total expense
  - Product Attributes – market share, price, # times product was advertised
- Predict # units purchased

Data Sample

Income	# members	Male head emp	Female head emp	# visits	Total spent
12500	3	0	1	385	10432
37500	4	1	1	265	8047
7000	2	1	1	106	1703
37500	6	1	1	492	9473
7000	1	0	0	213	2569
12500	2	0	0	476	5582
17500	5	1	0	442	6473
12500	2	0	1	371	5696
7000	2	1	1	438	6053

Category	sugar	ketchup	sugar	tissue	tuna
Price	1.4	1.1	0.9	1.4	1.3
Mkt. Share	1.1	24	2.128	10.3	1.7
# Times Adv.	6	40	16	0	3

7	17	0	0	0
2	1	0	0	0
0	1	0	0	0
2	2	5	0	4
1	1	1	0	0
0	1	1	0	0
1	0	6	3	1
0	1	0	0	0
3	0	2	0	0

Dataset Details

- Properties
  - Sparse – 74.86% values are 0
  - Very skewed – 99.12% values < 20, rest very high (outliers)
- Standardization of product attributes and # units purchased
- Linear least squares - very sensitive to outliers
  - Separate models for high and low valued matrix entries
  - Threshold of 20 units purchased

Results

- Model for low valued matrix entries
  - Bulk of the data (99.12%)

Algorithm	**Test Error**
Global Model (k=1,*l*=1)	4.24 (0.06)
CC (k=4,*l*=4)	4.002 (0.056)
Co-Cluster Models (k=4, l=4)	3.967 (0.034)
Reduced Parameter (k=4,*l*=4)	3.893 (0.052)
SCOAL(k=4,*l*=4)	3.965 (0.044)
M-SCOAL	**3.832 (0.035)**

Market Segmentation and Structure

Coefficients of global model and sample co-cluster models

Attribute	Global	Cust Seg 3, Prod Seg 3	Cust Seg 4, Prod Seg 4	Cust Seg 1, Prod Seg 2
intercept	0.00 (1.00)	-0.42 (0.00)	-0.14 (0.00)	0.15 (0.00)
income	-0.02 (0.00)	-0.09 (0.31)	-0.03 (0.00)	-0.02 (0.42)
# members	0.03 (0.00)	0.03 (0.74)	0.04 (0.00)	-0.04 (0.06)
male head emp.	0.00 (0.42)	-0.06 (0.42)	0.00 (0.87)	0.05 (0.04)
female head emp.	0.00 (0.62)	-0.07 (0.31)	0.00 (0.45)	0.02 (0.46)
# visits	0.02 (0.00)	-0.11 (0.05)	0.01 (0.06)	0.11 (0.00)
total spent	0.10 (0.00)	0.48 (0.00)	0.03 (0.00)	0.09 (0.00)
price	-0.02 (0.00)	-0.75 (0.00)	-0.02 (0.00)	0.42 (0.00)
market share	0.17 (0.00)	0.43 (0.00)	0.09 (0.00)	0.16 (0.00)
# times advertised	0.10 (0.00)	0.48 (0.00)	0.04 (0.00)	0.04 (0.06)

Cheapest, most popular products

Low market share

High income, large # visits

Lessons Learnt

- Interpretable and actionable segmentation and models
- Coefficients of co-cluster models differ significantly from global model
  - Multiple models required to capture heterogeneity
- Co-cluster models differ significantly
  - Different purchase factors important for different customer-product subsets
- Product attributes more indicative of preference
  - Elimination of insignificant predictors to get sparse models

Extensions

- Modeling time-series data (ICDM ‘08)
  - e.g. customer purchase behavior over time
- Active Learning (SCECR’09)
- Mining for the most reliable predictions (KDD’09)
- Scalable, parallel implementation for large scale applications

Simultaneous Co-segmentation and Learning

- Motivating example
  - Understand customer purchase behavior over time
  - Forecast future trends
- Challenges
  - Shifting trends across time
  - Variability across customers
- Simultaneously cluster customers and segment time
  - Learning a predictive model in each co-cluster
- Segment the time axis
  - Different from clustering – violates ordering constraint

Algorithm

- Iterative algorithm similar to SCOAL
  - alternate between model update and cluster/segment assignment

- Assignment of time segments
  - Dynamic programming (quadratic)
  - Greedy local search (linear)
    - Adjust segment boundaries to reduce objective function

Results on Tensor ERIM Dataset

- Dollars spent by households per week at 5 stores
  - 1240 households X 50 weeks X 5 stores

Algorithm	Train R-sq.	Test MSE	Test R-sq.
Global	0.44	166.36 (1.27)	0.44
Cluster Models	0.45	156.06 (1.36)	0.48
SCOAL	0.58	142.08 (0.99)	0.52
CoSeg	0.60	132.94 (0.88)	0.55

Prediction error averaged over 10 random 60-40 % training-test data splits

Active Learning

- Learner selects instances to be labeled such that the generalization accuracy is improved the most

- Example: predictive modeling in large scale recommender systems
  - Require large number of customers to rate many products
  - Obtaining ratings is expensive
- Solution: Select those customers and products to query whose feedback improves the prediction model the most

Active Learning with Multiple Local Models

- Models in different regions have different fits
  - Poorer fit in noisy/sparse regions of the input space
- Idea: acquire labels in regions with poor model fits
- For SCOAL with linear regression models

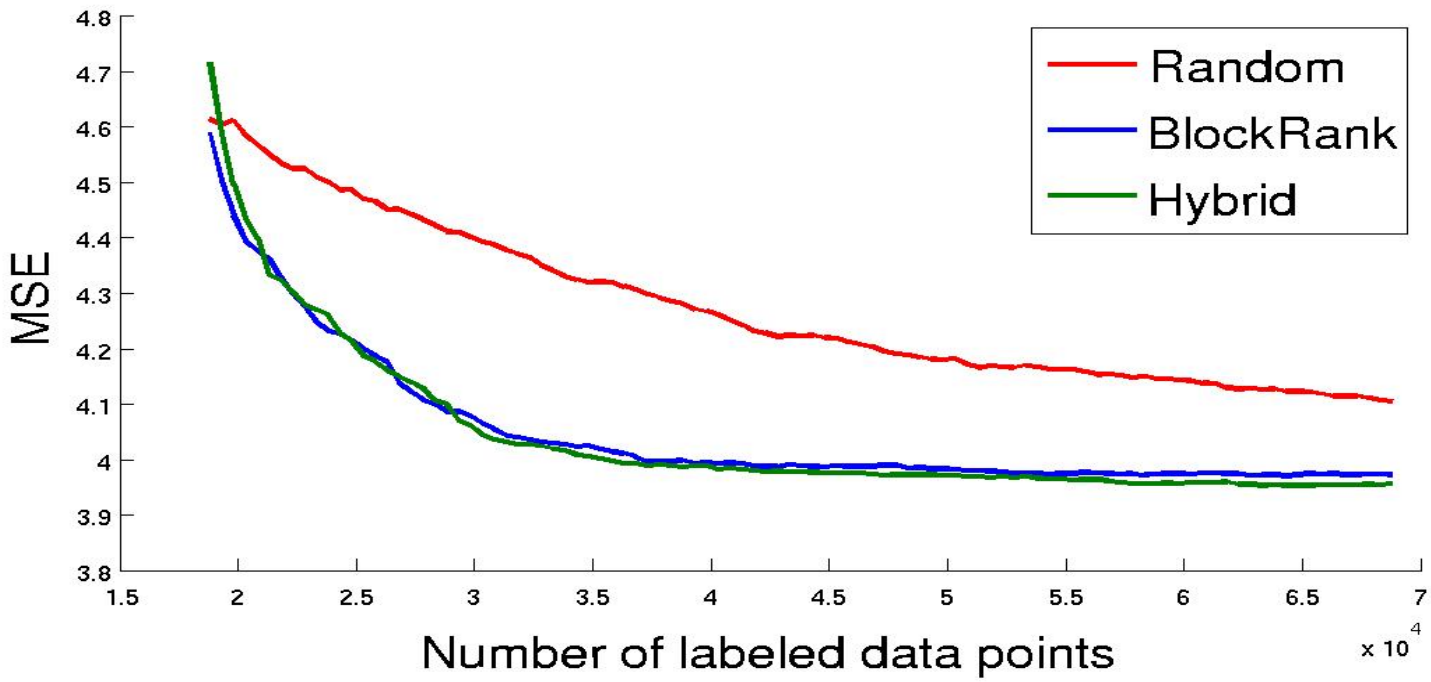
  Local model fit = co-cluster MSE
- Leads to BlockRank policy
  - Fast, actionable
  - Generalized: can be applied to any local modeling technique

Hierarchical BlockRank

- Learning multiple local models involved tuning a lot of parameters
  - May not have enough data
  - Single global model may do better when training data is limited

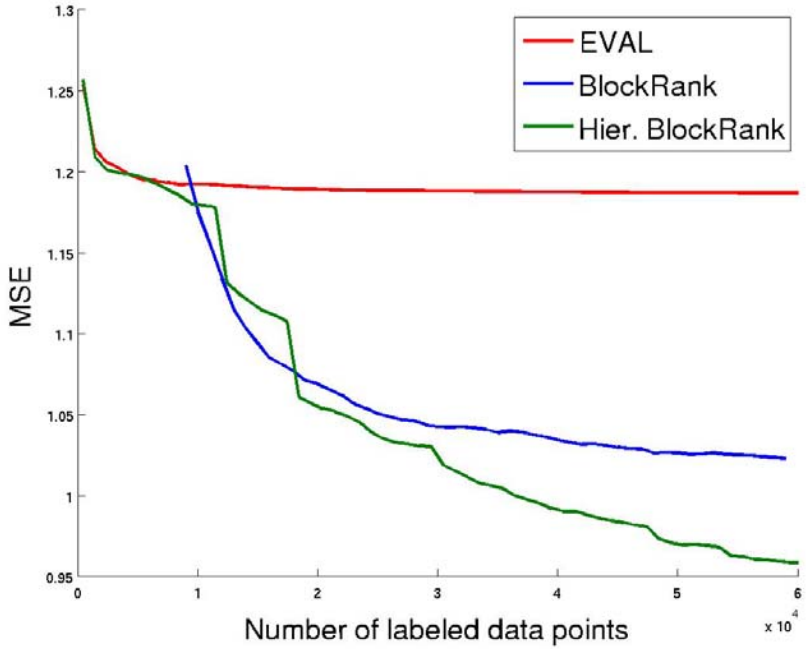
- Solution: increase model complexity (# local models) as more labeled data is acquired
  - Begin with a single co-cluster
  - Perform model selection step every N iterations
    - Increase # co-clusters if **validation set error** reduces
    - Use greedy "bisecting" step to add one row or column cluster

Evaluation of the BlockRank Policy

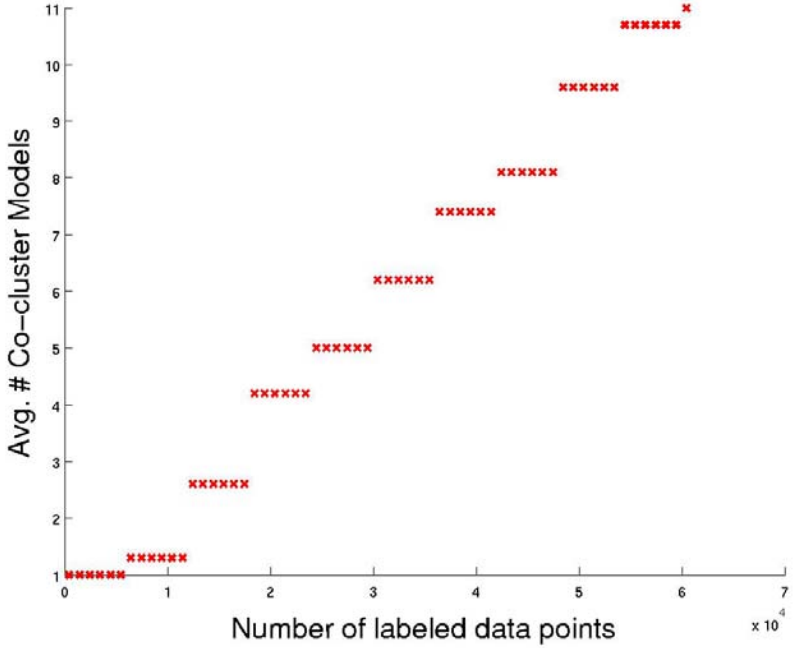


ERIM Marketing Data

MovieLens Data



MSE on held-out test set



local models learnt by Hierarchical BlockRank

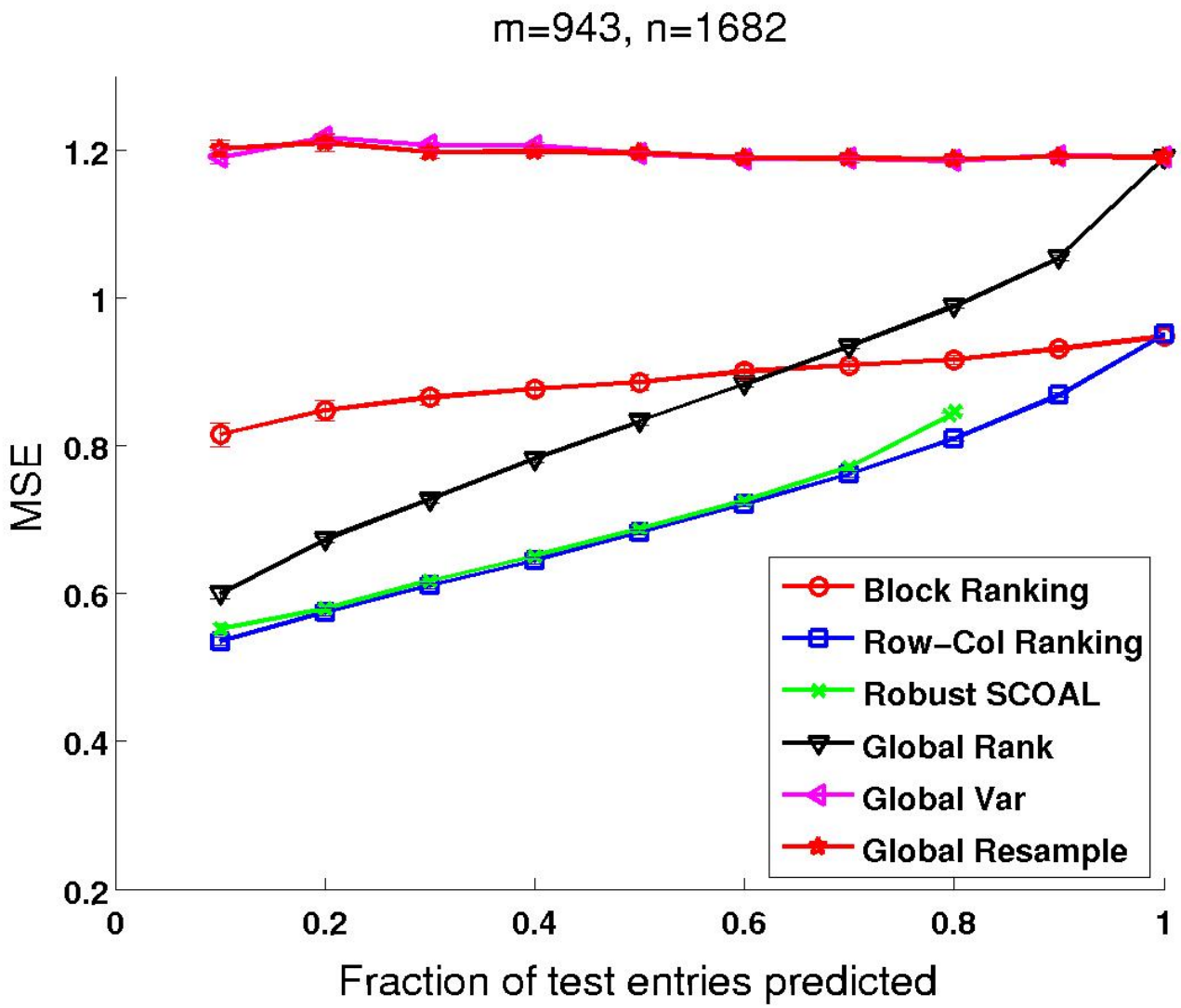
Mining for the Most Reliable Predictions from Dyadic Data

- Importance of accessing accuracy of predictions
  - Limited resources
  - Hence, take action only on the most accurate predictions
  - E.g. stock market player strategy (regression)
- Problem: rank predictions by estimate of their accuracy
- Single linear model
  - Classical approach: rank by prediction error variance
  - Dyadic data: rank by estimated mean row error + col error
- SCOAL based ranking
  - Row-Col ranking: rank by estimated mean row error + col error
  - Block ranking: rank by co-cluster error

Modeling a Selected Data Subset: Robust SCOAL

- Motivation
  - Need to make predictions for only a subset of the data
  - Can learn better models by detecting and discarding outliers
- Aim: simultaneously cluster s_r of m rows, s_c of n cols into k x l co-clusters
- Objective function: MSE over selected s_r x s_c matrix entries
- Algorithm: dynamic prog. to select s_r rows, s_c cols in each iteration

Comparing Ranking Techniques: MovieLens



Robust SCOAL Evaluation

s_r	s_c	% outliers discarded	% data discarded
900	90	98.4	61.6
1068	96	96.1	51.3
1236	102	91.4	40.2
1405	109	83.9	27.3
1573	115	70.7	14.1
1741	121	0	0

ERIM Marketing data: Outlier pruning for varying s_r and s_c values

Dataflow Solution to Co-clustering [KDD '09]

- Exploits parallelism in Bregman co-clustering
  - Parallelizes distance computation, learning co-cluster statistics
- Application to Netflix recommender problem
  - 100+ million ratings, 480,000+ users, 17,770 movies
  - Netflix production runtime: days
  - Dataflow runtime: 16.31 min

Summary

- SCOAL: actionable and interpretable predictive modeling technique for dyadic data

- Extensions
  - Modeling time series data
  - Active learning
  - Modeling noisy datasets
  - Parallelization

- Future work
  - Modeling in non-stationary domains, e.g. time drifts
  - Shrinkage
  - Robust error functions
  - Apply to a variety of very large datasets

Predictive Discrete Latent Factor Models

[D. Agarwal and S. Merugu ’07]

- Similar motivation and problem setting
  - Prediction of missing matrix entries (dyadic response variables), given attributes (covariate information)
- Uses co-clustering to solve a prediction problem
- Response variable modeled as a sum of
  - Function of covariates (global structure)
  - Co-cluster specific constant (local structure)
- Exploits local structure
  - Co-cluster specific constant assumed as part of noise model
  - *Teased out* of global model residues

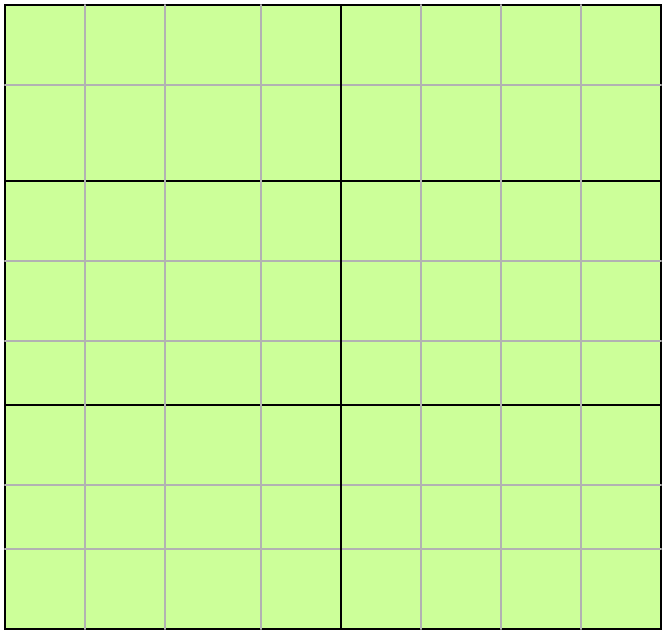
Reduced Parameter Approach
Model Update Step

Update customer coefficients

Response variable: $y_{ij} = z_{ij} - \boldsymbol{\beta}_p^T \mathbf{P}_j$

Solve: $\mathbf{y} = \boldsymbol{\beta}_c^T C$

k least squares updates



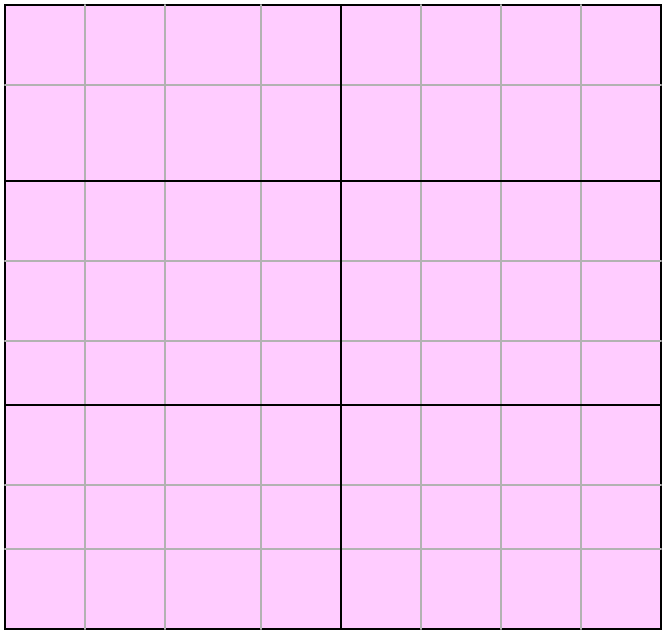
Reduced Parameter Approach
Model Update Step

Update product coefficients

Response variable: $y_{ij} = z_{ij} - \boldsymbol{\beta}_c^T \mathbf{C}_i$

Solve: $\mathbf{y} = \boldsymbol{\beta}_p^T \mathbf{P}$

l least squares updates



Short-Term Load Forecasting

[M. Djukanovic *et al.* '93]

- Problem: Forecast the hourly electric load pattern for a day
- First level of division
  - Model working days, weekends and holidays separately
- For each day type, cluster input data into coherent groups and train model in each group
  - Relation between input features and load profile stronger in each group vs. entire population
- Classify test point into a cluster and use corresponding model to forecast load

Co-clustering

- Simultaneously clusters along multiple axes
- Exploits the duality between the axes
  - Improves upon single sided clustering

- Applications
  - Microarray data analysis (genes and experiments)
  - Text data clustering (documents and words)

- Bregman Co-clustering [Banerjee *et al.* '06]
  - Partitional: divides matrix into a grid of rectangular blocks
  - Can deal with missing data

PDLF Model

$$p(z_{ij} \mid \mathbf{x}_{ij}) = \sum_{I=1}^{k} \sum_{J=1}^{l} \pi_{IJ} f_{\phi}(z_{ij}; g(\mu_{ij}) = \mathbf{x}_{ij}{}^{T}\boldsymbol{\beta} + \delta_{IJ})$$

- Constrained mixture model
  - $k * l$ components, π_{IJ}: mixture prior of IJ^{th} component
- Each component is a generalized linear model
  - f_{ϕ}: exponential family, g: link function
- Global trends $\mathbf{x}_{ij}{}^{T}\boldsymbol{\beta}$ shared across the components
- Each co-cluster/latent factor has an additional offset: δ_{IJ}

Model Estimation

- Generalized EM algorithm
  - Soft vs. hard assignment

- Main steps
  - Random initialization of row/column clusters and parameters
  - Repeat till convergence
    - Estimate global model coefficients β (Newton-Raphson's method)
    - Estimate co-cluster offsets $\delta_{IJ,}$
    - Find the optimal row and column clustering

- Scalable: each iteration linear in # observations

PDLF vs. Model CC

- PDLF
  - Single global model + co-cluster constants
  - Robust even when data is limited

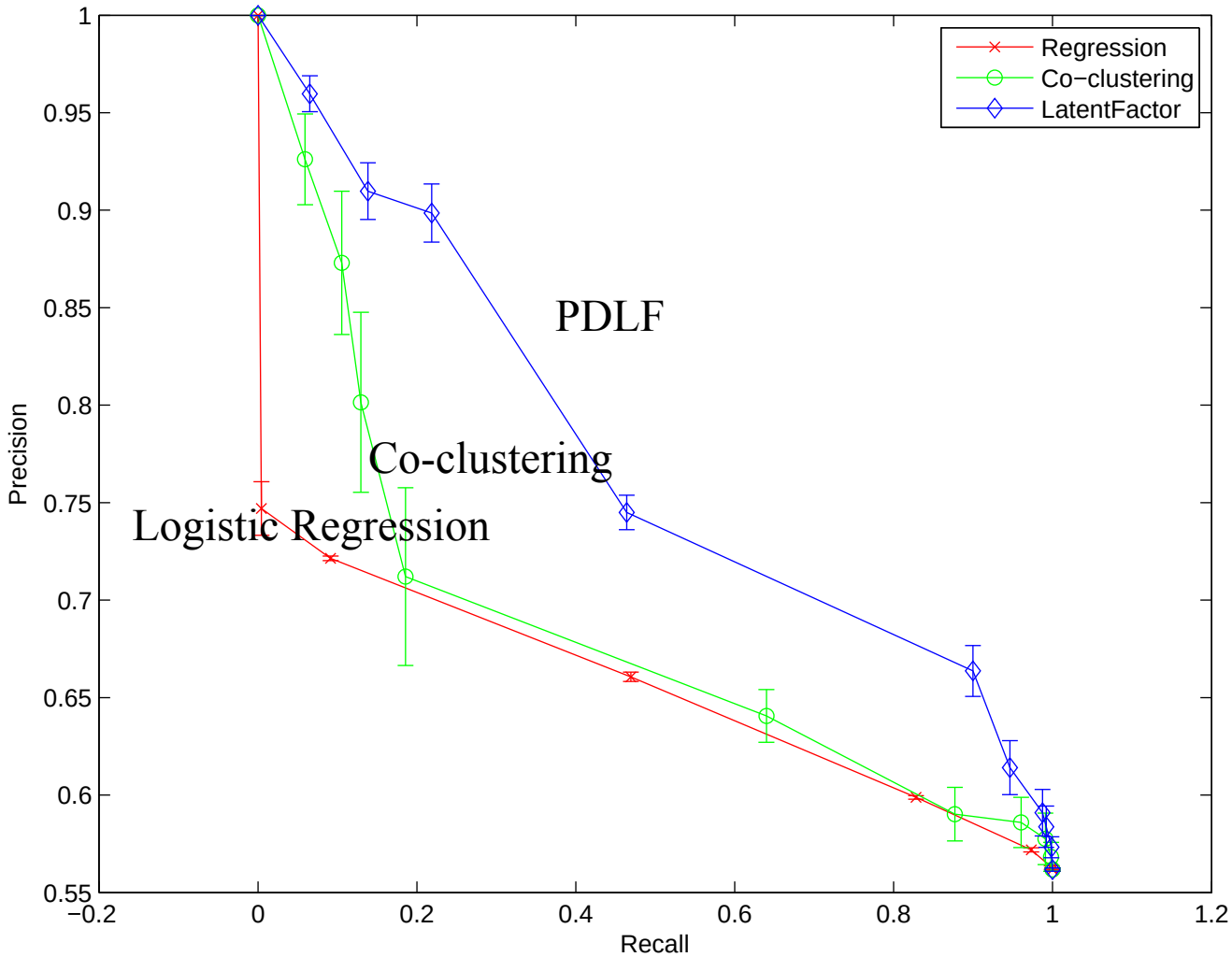
$$p(z_{ij} \mid x_{ij}, \rho, \gamma) = f_{\phi}(z_{ij}; \boldsymbol{\beta}^T \mathbf{x}_{ij} + \delta_{\rho(i),\gamma(j)}), [i]_1^m, [j]_1^n$$

- Model CC
  - k x l co-cluster models
  - Works well when large amount of data is available

$$p(z_{ij} \mid x_{ij}, \rho, \gamma) = f_{\phi}(z_{ij}; \boldsymbol{\beta}^T_{\rho(i),\gamma(j)} \mathbf{x}_{ij}), [i]_1^m, [j]_1^n$$

- Complementary approaches

Logistic Regression on Movie Lens



Rating > 3: +ve
23 covariates

Objective Function Details

- Indicates how well the co-cluster models fit given data
- Based on the prediction model, not cluster homogeneity!
- Elementwise squared error summed over all matrix entries

Sum over
all matrix entries

$$\sum_{u,v} w_{uv}(z_{uv} - \hat{z}_{uv})^2$$

$$\hat{z}_{uv} = \boldsymbol{\beta}_{\rho(u)\gamma(v)}{}^T \mathbf{x}_{uv}$$

Predicted value: using
co-cluster specific (linear) model

Reduced Parameter Approach

- Simultaneous co-clustering and prediction
  - k * *l* independent models
  - (1 + |C| + |P|) * k * *l* parameters
  - May overfit when training data is limited
- Single model
  - (1 + |C| + |P|) parameters
  - May not be adequate
- Reduced Parameter Approach
  - k * *l* models, but smoothing achieved by sharing parameters
  - Customer (product) coefficients for all models in the same row (column) cluster are constrained to be identical
  - (1 + |C|) * k + (1 + |P|) * *l* parameters
- Alternative: Shrinkage between global model and local models

Predicting Missing Values

- If missing matrix entry z_{uv} is assigned to row cluster g and col cluster h with model parameters $\boldsymbol{\beta}_{gh}$, predict z_{uv} as

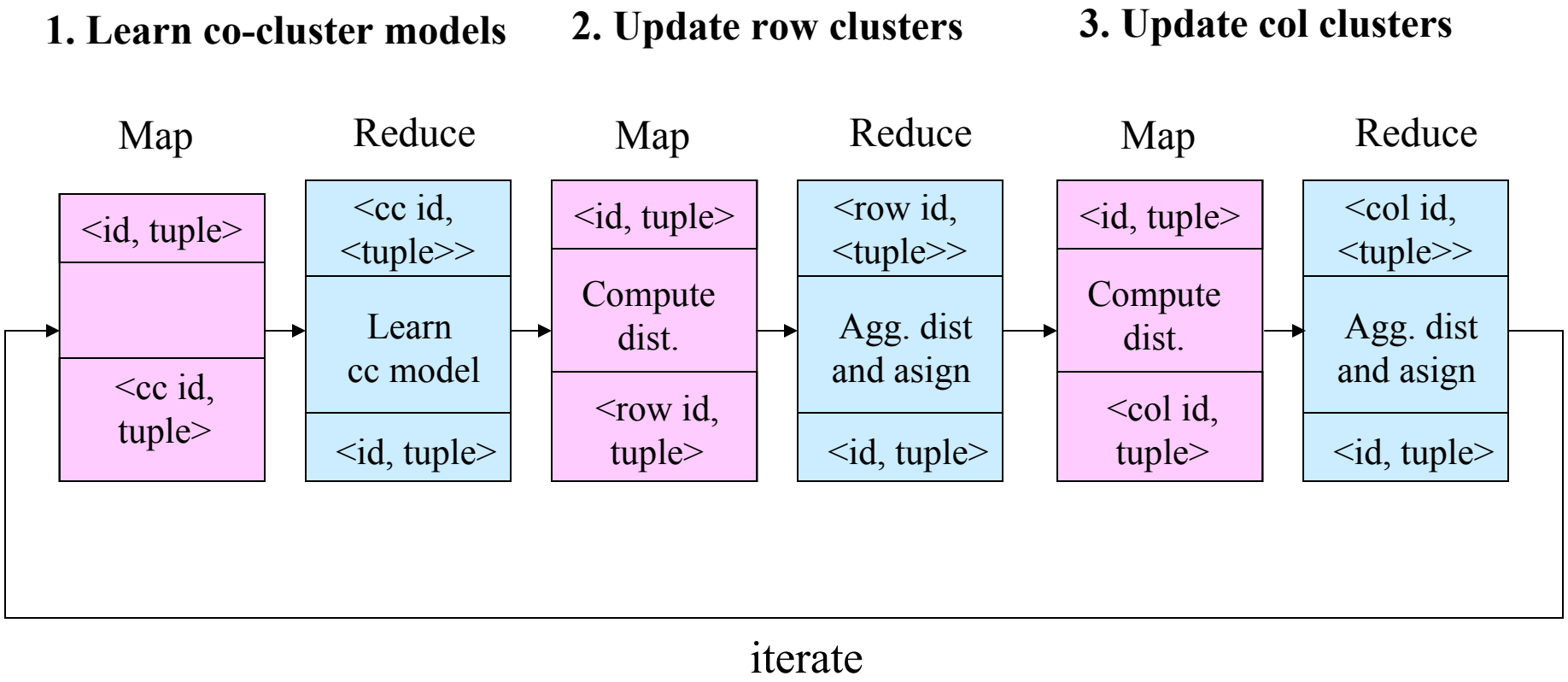
1. Classification:

$$P(z_{uv} = 1) = \frac{1}{1 + \exp(-\boldsymbol{\beta}_{gh}^T \mathbf{x}_{uv})}$$

2. Regression:

$$\hat{z}_{uv} = \boldsymbol{\beta}_{gh}^T \mathbf{x}_{uv}$$

Large Scale SCOAL

- Expected to show maximum value on large datasets
  - Large heterogeneity
  - Sufficient data available for learning parameters
- Distributed, scalable implementation using Map-Reduce framework
  - Run on Hadoop cluster
- Can then analyze impact of data size, sparsity on accuracy, computation time

SCOAL Map-Reduce Pseudo Code