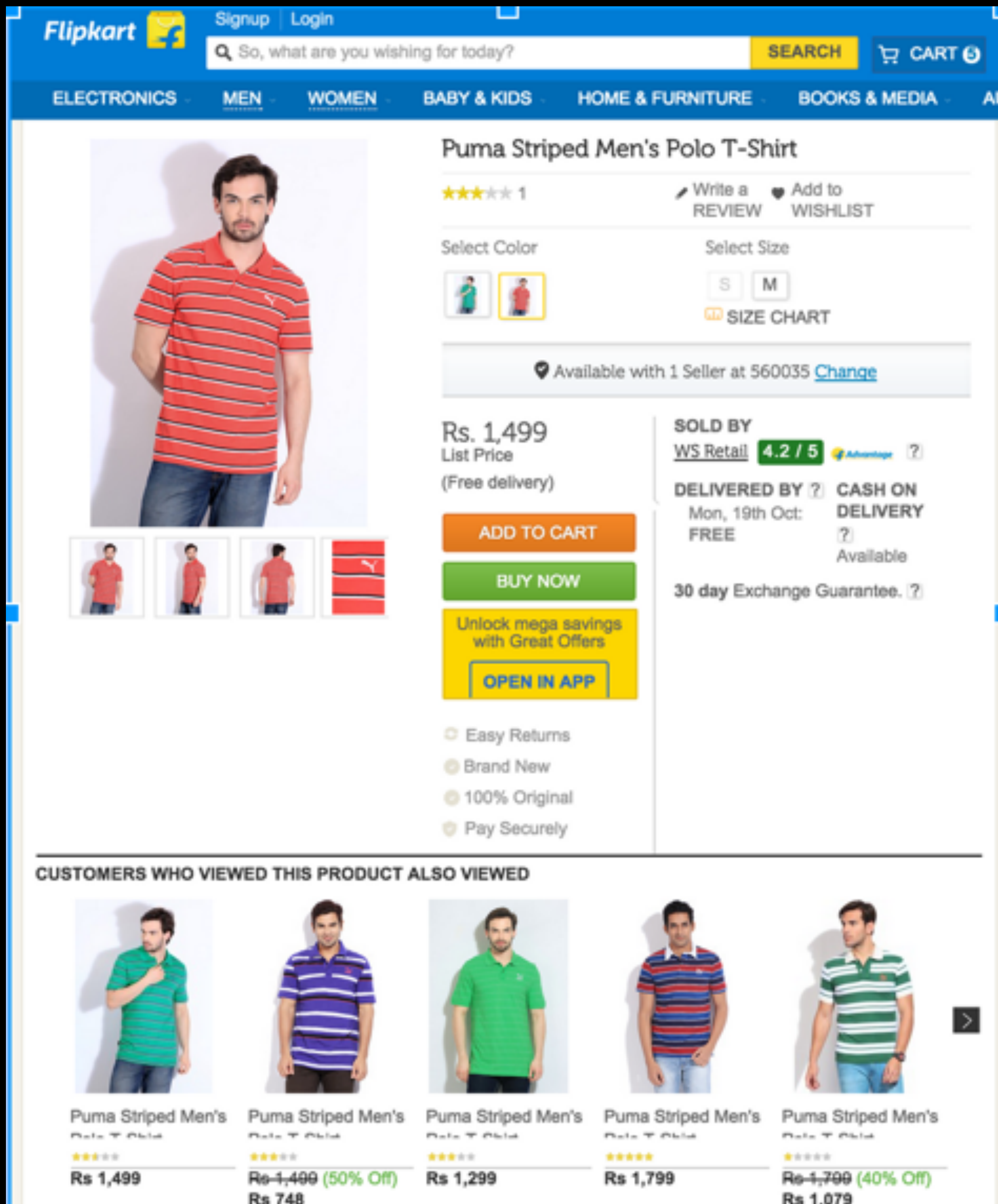
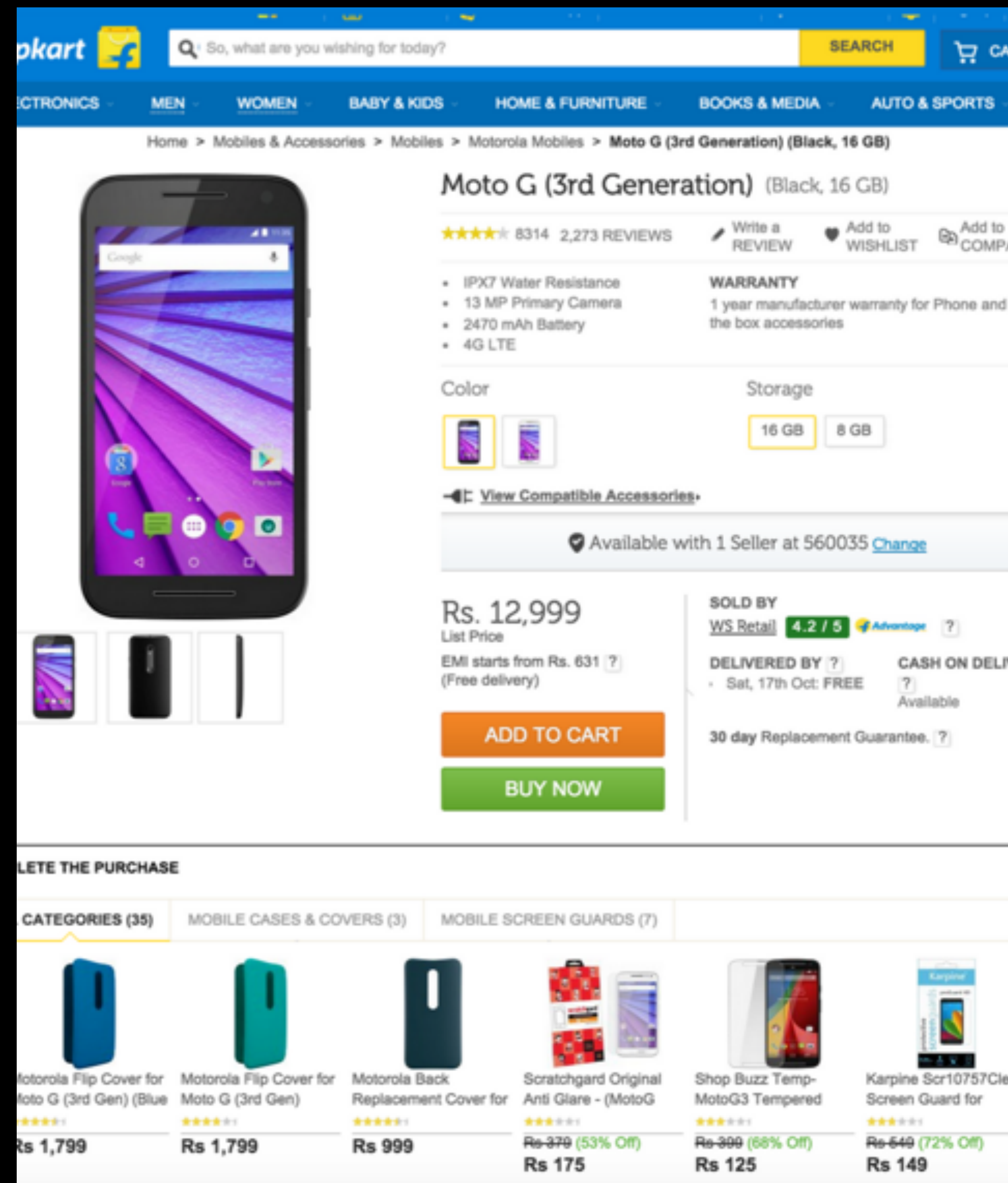
Quick Recap



Flipkart
Signup | Login
So, what are you wishing for today?
SEARCH
CART
ELECTRONICS
MEN
WOMEN
BABY & KIDS
HOME & FURNITURE
BOOKS & MEDIA
Puma Striped Men's Polo T-Shirt
Write a REVIEW
Add to WISHLIST
Select Color
Select Size
S
M
SIZE CHART
Available with 1 Seller at 560035 Change
Rs. 1,499
List Price
(Free delivery)
SOLD BY
WS Retail
4.2 / 5
DELIVERED BY
Mon, 19th Oct: FREE
CASH ON DELIVERY
Available
ADD TO CART
BUY NOW
Unlock mega savings with Great Offers
OPEN IN APP
30 day Exchange Guarantee.
Easy Returns
Brand New
100% Original
Pay Securely
CUSTOMERS WHO VIEWED THIS PRODUCT ALSO VIEWED
Puma Striped Men's
Rs 1,499
Rs 1,400 (50% Off)
Rs 748
Rs 1,299
Rs 1,799
Rs 1,799 (40% Off)
Rs 1,079



So, what are you wishing for today?
SEARCH
MEN
WOMEN
BABY & KIDS
HOME & FURNITURE
BOOKS & MEDIA
AUTO & SPORTS
Home > Mobiles & Accessories > Mobiles > Motorola Mobiles > Moto G (3rd Generation) (Black, 16 GB)
Moto G (3rd Generation) (Black, 16 GB)
8314 2,273 REVIEWS
Write a REVIEW
Add to WISHLIST
IPX7 Water Resistance
13 MP Primary Camera
2470 mAh Battery
4G LTE
WARRANTY
1 year manufacturer warranty for Phone and the box accessories
Color
Storage
16 GB
8 GB
View Compatible Accessories
Available with 1 Seller at 560035 Change
Rs. 12,999
List Price
EMI starts from Rs. 631
(Free delivery)
SOLD BY
WS Retail
4.2 / 5
DELIVERED BY
Sat, 17th Oct: FREE
Available
ADD TO CART
BUY NOW
30 day Replacement Guarantee.
CATEGORIES (35)
MOBILE CASES & COVERS (3)
MOBILE SCREEN GUARDS (7)
Motorola Flip Cover for Moto G (3rd Gen)
Motorola Back Replacement Cover for
Scratchgard Original Anti Glare - (MotoG
Shop Buzz Temp-MotoG3 Tempered
Rs 1,799
Rs 999
Rs 379 (53% Off)
Rs 175
Rs 399 (68% Off)
Rs 125
Rs 549 (72% Off)
Rs 149

Scaling Neighbourhood Methods

Collaborative Filtering

- m = #items
- n = #users
- Complexity : m * m * n

Comparative Scale of Signals

- ~50 M users
- ~25 M items
- Explicit Ratings ~ O(1M) (1 per billion)
- Purchase ~ O(100M) (100 per billion)
- Browse ~ O(10B) (10000 per billion)

Implicit Signals Used

- Bought History
- Browse History
- Compare History

Category-partitioned v/s Category independent


KEEP
CALM
AND
CARRY
ON
KEEP
CALM
AND
CARRY
ON


KEEP
CALM
AND
CARRY
ON


BRASIL


DESTROYER

Similarity Metric for boolean matrix

- Cosine Similarity

  A. Pair Count (p) - P1 and P2

  B. Individual Count (n_i) - P1, P2 individually

Employing Map Reduce

- Calculate 'p' by forming pairs and counting
- Calculate 'n1' by making P1 as the key
- Calculate 'n2' by making P2 as the key
- Took 2 hours on 5 years of data
- Scales Horizontally

Map Reduce 1

B1 -> P1, P2, P3, P4,
B2 -> P2, P3, P4, P5,
Generating pairs:

Mapper: Key(Pair of items) => Value(weight)
Reducer: Accumulates the weights for each Pair.

Key	Value
P1 P2	1
P1 P3	1
P1 P4	1
P2 P3	1
P2 P4	1
P3 P4	1

Key	Value
P2 P3	1
P2 P4	1
P2 P5	1
P3 P4	1
P3 P5	1
P4 P5	1

=>

Reducer O/P
P1 P2 1
P1 P3 1
P1 P4 1
P2 P3 2
P2 P4 2
P2 P5 1

Reducer O/P
P3 P4 2
P3 P5 1
P4 P5 1

flipkart.com

Map Reduce 2

Calculating the value '**n1**':

Input:

P1 P4 1

P2 P3 1

P4 P5 1

P2 P4 1

Mapper: Key (**i1**) => Value(Pairs with weights)

Reducer: Accumulates the **i1**'s to form the Pairs with weights, **n1**

Key	Value
P1	P1 P4 1 **1**
P2	P2 P3 1 **1**
P4	P4 P5 1 **1**
P2	P2 P4 1 **1**

=>

Reducer Output
P1 P4 1 **1**
P2 P3 1 **2**
P4 P5 1 **1**
P2 P4 1 **2**

flipkart.com

Map Reduce 3

Calculating the value '**n2**':

Input:

P1 P4 1

P2 P3 1

P4 P5 1

P2 P4 1

Mapper: Key (**i2**) => Value(Pairs with weights)

Reducer: Accumulates the **i2**'s to form the Pairs with weights, **n2**

Key	Value
P4	P1 P4 1 1 **1**
P3	P2 P3 1 1 **1**
P5	P4 P5 1 1 **1**
P4	P2 P4 1 1 **1**

=>

Reducer Output
P1 P4 1 1 **2**
P2 P3 1 2 **1**
P4 P5 1 1 **1**
P2 P4 1 2 **2**

flipkart.com

Latent Variable Models and Factorization Models

Akash Khandelwal, Avijit Saha, Mohit Kumar, Vivek Mehta

Outline

- Introduction
  - Recommender Systems Recap
  - Latent Variable Models
  - Factorization Models
- Matrix Factorization
  - Singular Value Decomposition
  - BPMF
- Factorization Machine
- Conclusion

Recommender Systems (RSs)

- Collaborative Filtering (CF)
  - Neighborhood Based
    - KNN
  - Model Based
    - Cluster-based CF and Bayesian classifiers.
    - Latent variable models such as, LDA, pLSA, and matrix factorization (MF).
- Content Based
- Knowledge Based
- Hybrid

Latent Variable Models

- Supplementing a set of observed variables with additional latent, or hidden, variables.
- Latent variable models are widely used in several domains such as machine learning, statistics, data mining.
- Reveals hidden structure which explains the data.
- Latent variable models consider a joint distribution over the hidden and observed variables.
- Hidden structure is found by calculating the posterior.
- LDA is an example of latent variable models.

Factorization Models

- One of the widely used latent variable models in the RSs community.
- preferences of a user are determined by a small number of unobserved latent factors.
  - Matrix Factorization : Each user and item are mapped to a latent factor vector:

$$\boldsymbol{u}_i \in \mathbb{R}^K$$

$$\boldsymbol{v}_j \in \mathbb{R}^K$$

  - Tensor Factorization : Mapping of each variable of each category type to a K dimensional latent factor vector.
  - Many problem specific factorization models.

Focus

- Factorization Models
  - Matrix Factorization (MF)
  - Factorization Machine (FM)

Singular Value Decomposition

Singular value decomposition (SVD) is a factorization of a matrix. Formally, the SVD of an $\boldsymbol{R} \in \mathbb{R}^{I \times J}$ is:

$$\boldsymbol{R} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^*, \tag{1}$$

where, $\boldsymbol{U} = I \times I$ unitary matrix
$\boldsymbol{\Sigma} = I \times J$ rectangular diagonal matrix
$\boldsymbol{V}^* = J \times J$ unitary matrix
$\sigma_{i,i}$ of $\boldsymbol{\Sigma}$ are known as the singular values of $\boldsymbol{R}$
I columns of $\boldsymbol{U}$ and the J columns of $\boldsymbol{V}$ are called the left-singular vectors and right-singular vectors of $\boldsymbol{R}$, respectively.

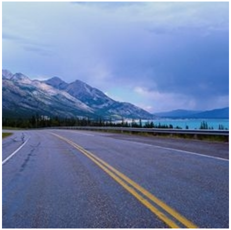
Dimensionality Reduction Using SVD

Let, $\boldsymbol{R} \in \mathbb{R}^{I \times J}$

$$\textit{Apply SVD:} \qquad \boldsymbol{R} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^*, \tag{2}$$

$$\textit{Estimate :} \qquad \hat{\boldsymbol{R}} = \boldsymbol{U} \begin{pmatrix} \sigma_{1,1} & & & & & \\ & \sigma_{2,2} & & & & \\ & & \ddots & & & \\ & & & \sigma_{K,K} & & \\ & & & & 0 & \\ & & & & & \ddots \end{pmatrix} \boldsymbol{V}^*. \tag{3}$$

Example of SVD



(a)

(b)

(c)

(d)

MF for RSs

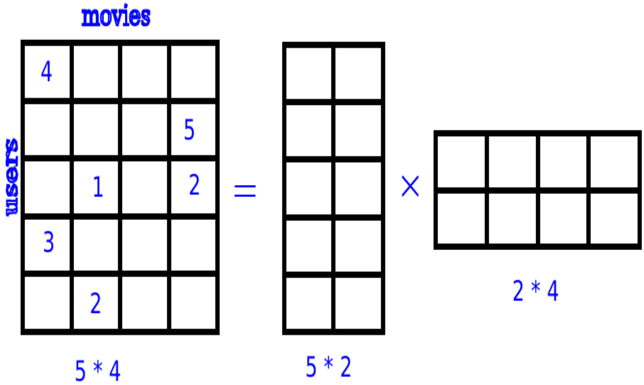


Figure 1: Matrix Factorization

MF for RSs

- Consider a user-movie matrix $\boldsymbol{R} \in \mathbb{R}^{I \times J}$ where the r_{ij} cell represents the rating provided to the j^{th} movie by the i^{th} user. MF decomposes the matrix $\boldsymbol{R}$ into two low-rank matrices $\boldsymbol{U} = [\boldsymbol{u}_1, \boldsymbol{u}_2, ..., \boldsymbol{u}_I]^T \in \mathbb{R}^{I \times K}$ and $\boldsymbol{V} = [\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_J]^T \in \mathbb{R}^{J \times K}$:

$$\boldsymbol{R} \sim \boldsymbol{U}\boldsymbol{V}^T. \tag{4}$$

$$\sum_{(i,j)\in\Omega} \left(r_{ij} - \boldsymbol{u}_i^T \boldsymbol{v}_j \right)^2 \tag{5}$$

- SVD with K singular value would been the solution if $\boldsymbol{R}$ is fully observed.
- However, $\boldsymbol{R}$ is partially observed.
- Solution: Stochastic gradient descent to rank-1 update:

$$e_{ij} = r_{ij} - \boldsymbol{u}_i^T \boldsymbol{v}_j$$

$$u_{ik} = u_{ik} + \nu(e_{ij} v_{jk} - \lambda u_{ik}) \tag{6}$$

$$v_{jk} = v_{jk} + \nu(e_{ij} u_{ik} - \lambda v_{jk}) \tag{7}$$

Then iterate this for each rank K.

Problem with SVD approach

- Learning rate and regularization parameters needs to be tuned manually.
- Overfitting.
- Solution: Bayesian Probabilistic Matrix Factorization (BPMF) [1].

BPMF Model

The likelihood term of BPMF is as follows:

$$p(\boldsymbol{R}|\boldsymbol{\Theta}) = \prod_{(i,j)\in\Omega} \mathcal{N}(r_{ij}|\boldsymbol{u}_i^T \boldsymbol{v}_j, \tau^{-1}), \tag{8}$$

where $\boldsymbol{u}_i$ is the latent factor vector for the i^{th} user,
$\boldsymbol{v}_j$ is the latent factor vector for the j^{th} item,
τ is the model precision,
Ω is the set of all observations, and
$\boldsymbol{\Theta}$ is the set of all the model parameters.

Priors

Independent priors are placed on all the model parameters in $\boldsymbol{\Theta}$ as:

$$p(\boldsymbol{U}) = \prod_{i=1}^{I} \mathcal{N}(\boldsymbol{u}_i | \boldsymbol{\mu_u}, \boldsymbol{\Lambda_u}^{-1}), \tag{9}$$

$$p(\boldsymbol{V}) = \prod_{j=1}^{J} \mathcal{N}(\boldsymbol{v}_j | \boldsymbol{\mu_v}, \boldsymbol{\Lambda_v}^{-1}). \tag{10}$$

Hyperpriors

Further place Normal-Wishart priors are placed on all the hyperparameters $\mathbf{\Theta}_H = \{\{\boldsymbol{\mu}_{\boldsymbol{u}}, \mathbf{\Lambda}_{\boldsymbol{u}}\}, \{\boldsymbol{\mu}_{\boldsymbol{v}}, \mathbf{\Lambda}_{\boldsymbol{v}}\}\}$ as:

$$\begin{aligned} p(\boldsymbol{\mu}_{\boldsymbol{u}}, \mathbf{\Lambda}_{\boldsymbol{u}}) &= \mathcal{N}\mathcal{W}\left(\boldsymbol{\mu}_{\boldsymbol{u}}, \mathbf{\Lambda}_{\boldsymbol{u}} | \mu_0, \beta_{\mathbf{0}}, \boldsymbol{W}_0, \boldsymbol{\nu}_0\right), \quad (11) \\ &= \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{u}} | \mu_0, (\beta_0 \mathbf{\Lambda}_u)^{-1}) \boldsymbol{\mathcal{W}}(\mathbf{\Lambda}_u | \boldsymbol{W}_0, \boldsymbol{\nu}_0) \\ p(\boldsymbol{\mu}_v, \mathbf{\Lambda}_v) &= \mathcal{N}\mathcal{W}\left(\boldsymbol{\mu}_v, \mathbf{\Lambda}_v | \mu_0, \beta_0, \boldsymbol{W}_0, \boldsymbol{\nu}_0\right). \quad (12) \end{aligned}$$

where

$$\boldsymbol{\mathcal{W}}(\mathbf{\Lambda} | \boldsymbol{W}_0, \boldsymbol{\nu}_0) = \frac{1}{C} |\mathbf{\Lambda}|^{\nu_0 - D - 1} \exp(-\frac{1}{2} Tr(\boldsymbol{W}_0^{-1} \mathbf{\Lambda}))$$

Joint Distributions

The joint distribution of the observations and the hidden variables can be written as:

$$p(\boldsymbol{R}, \boldsymbol{\Theta}, \boldsymbol{\Theta}_H | \boldsymbol{\Theta}_0) = p(\boldsymbol{R}|\boldsymbol{\Theta}) p(\boldsymbol{U}) p(\boldsymbol{V}) p(\boldsymbol{\mu}_u, \boldsymbol{\Lambda}_u | \boldsymbol{\Theta}_0) p(\boldsymbol{\mu}_v, \boldsymbol{\Lambda}_v | \boldsymbol{\Theta}_0), \quad (13)$$

where $\boldsymbol{\Theta}_0 = \{\boldsymbol{\mu}_0, \boldsymbol{\beta}_0, \boldsymbol{W}_0, \boldsymbol{\nu}_0\}$

Inference

- Evaluation of the joint distribution in Eq. (13) is intractable.
- However, all the model parameters are conditionally conjugate.
- So we Gibbs sampler has closed form updates.
- Replacing Eq. (8)-(12) in Eq. (13), the sampling distribution of $\boldsymbol{u}_i$ can be written as follows:

$$p(\boldsymbol{u}_i|-) \sim \mathcal{N}\left(\boldsymbol{u}_i|\boldsymbol{\mu}^*, (\boldsymbol{\Lambda}^*)^{-1}\right), \tag{14}$$

$$\boldsymbol{\Lambda}^* = \left(\boldsymbol{\Lambda}_u + \tau \sum_{j \in \Omega_i} \boldsymbol{v}_j \boldsymbol{v}_j^T\right) \tag{15}$$

$$\boldsymbol{\mu}^* = (\boldsymbol{\Lambda}^*)^{-1} \left(\boldsymbol{\Lambda}_u \boldsymbol{\mu}_u + \tau \sum_{j \in \Omega_i} \boldsymbol{v}_j r_{ij}\right) \tag{16}$$

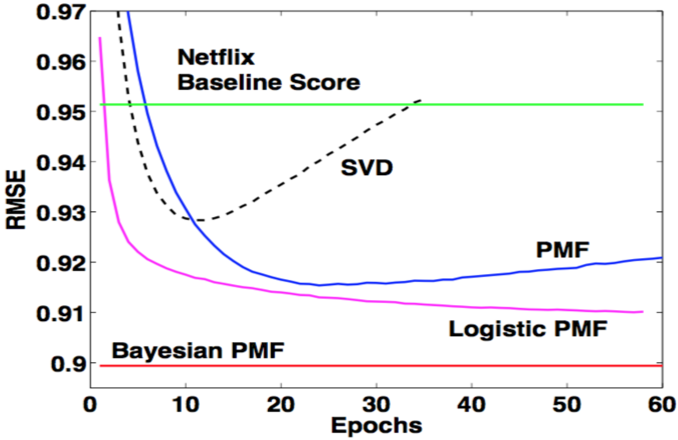


Figure 2: RMSE vs Iterations

Factorization Models

- Matrix factorization [2].
- Tensor factorization [3].
- Specific models like SVD++ [4], TimeSVD++ [5], FPMC [6], and BPTF [3], etc. have been developed.
- Several Learning technique like SGD, ALS, variational Bayes, MCMC Gibbs sampling have been developed.

Factorization Models Vs Feature Based Techniques

Advantages of Factorization Model:

- Scalability and Performance.

Problem:

- deriving inference techniques for each individual model is a time consuming task and requires considerable expertise.

Advantages of Feature Based Techniques:

- Generic approach.
- Can be solved using standard tools like LIBSVM or SVMLight.

Problem:

- Can not handle very sparse data.

Factorization Machine

- A Generic framework [7] proposed by Stephen Rendle.
- Combines advantages of both factorization models and feature based model.
- Can subsume many state of the art factorization model like SVD++, TimeSVD++, FPMC, PITF, etc.
- Performs well for sparse data where SVMs fails.

Feature Representation of Factorization Machine

Example: (U1,M1,G1,2),(U1,M3,G2,5),(U2,M2,G3,4) and (U3,M1,G1,5)

Feature x

	User				Movie				Genre				**Target y**	
x1	1	0	0	....	1	0	0	....	1	0	0	....	2	y1
x2	1	0	0	....	0	0	1	....	0	1	0	....	5	y2
x3	0	1	0	....	0	1	0	....	0	0	1	....	4	y3
x4	0	0	1	....	1	0	0	....	1	0	0	....	5	y4
	U1	U2	U3	....	M1	M2	M3	....	G1	G2	G3	....		

Factorization Machine

Following is the equation for FM.

$$y_n = w_0 + \sum_{i=1}^{D} w_i x_{ni} + \sum_{i=1}^{D} \sum_{j=i+1}^{D} x_{ni} x_{nj} \sum_{k=1}^{K} v_{ik} v_{jk} \tag{17}$$

Assumptions of FM are following:

$$y_n | x_n, \theta \sim \mathcal{N}(\hat{y}(x_n, \theta), \alpha^{-1})$$

$$y_n | x_n, \theta \sim Bernoulli(b(\hat{y}(x_n, \theta)))$$

And L2 regularization on θ

Mimic MF

$$\hat{y} = w_0 + w_u + w_i + \boldsymbol{v}_u^T \boldsymbol{v}_i \quad (18)$$

Feature representation of FM: $D = |U \cup I|$

$$x_j = \delta(j = u \vee j = i)$$

Mimic SVD++

$$\hat{y} = w_0 + w_u + w_i + \mathbf{v}_u^T \mathbf{v}_i + \frac{1}{\sqrt{N}} \sum_{l \in N_u} \mathbf{v}_i^T \mathbf{v}_l \quad (19)$$

Feature representation of FM: $D = |U \cup I \cup L|$

$x_j = 1$ if $j = u \vee j = i$

$= \frac{1}{\sqrt{N}}$ if $j \in N_u$

$= 0$ else

Algorithms of Factorization Machine

- Stochastic gradient descent is the simplest algorithm to solve FM (SGD-FM).
- MCMC based Bayesian Factorization Machine gives state-of-the-art performance (MCMC-FM).
- Alternating least square and adaptive stochastic gradient descent.

Comparison of each Leaning Techniques

- SGD-FM
  - Pros: SGD online algorithm and more scalable.
  - Cons: Costly cross validation of parameters.
- MCMC-FM
  - Pros: Performance.
  - No cross validation required.

SGD Learning

Following is the equation for FM.

$$\hat{y}_n = w_0 + \sum_{i=1}^{D} w_i x_{ni} + \sum_{i=1}^{D} \sum_{j=i+1}^{D} x_{ni} x_{nj} \sum_{k=1}^{K} v_{ik} v_{jk} \tag{20}$$

Cost function is:

$$\sum_{n=1}^{N} (y_n - \hat{y}_n)^2 + L2 \ regularization \tag{21}$$

SGD update equations:

$$v_{ik} = v_{ik} + \nu \left(2 * (y_n - \hat{y}_n) x_{ni} \sum_{l=1 \& l \neq i}^{N} x_{nl} v_{nl} - 2\lambda v_{ik} \right) \tag{22}$$

MCMC-FM

Likelihood:

$$y_n \sim \mathcal{N}(y_n|\hat{y}_n, \tau)$$

Prior:

$$p(w_0) \sim \mathcal{N}(w_0|\mu_0, \sigma_0)$$

$$p(w_i) \sim \mathcal{N}(w_i|\mu_w, \sigma_w)$$

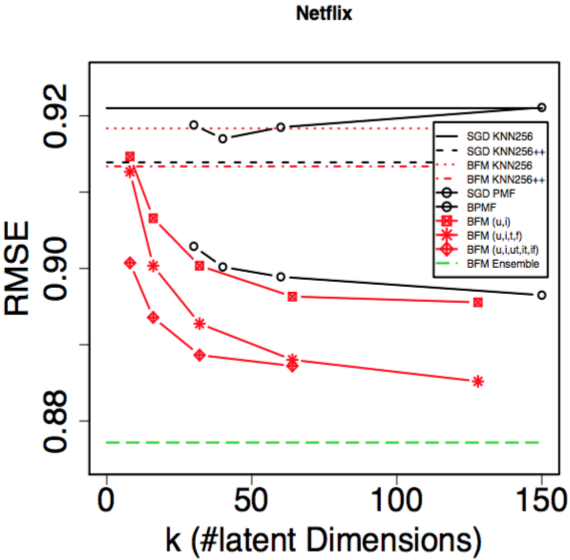
$$p(v_{ik}) \sim \mathcal{N}(v_{ik}|\mu_k, \sigma_k)$$

Hyperprior:

$$p(\mu_w) \sim \mathcal{N}(\mu_w|\mu, \sigma_w \nu_0) \qquad p(\sigma_w) \sim G(\sigma_w|\alpha_0, \beta_0)$$

$$p(\mu_k) \sim \mathcal{N}(\mu_k|\mu, \sigma_k \nu_0) \qquad p(\sigma_k) \sim G(\sigma_k|\alpha_0, \beta_0)$$

Results



Netflix
RMSE
0.92
0.90
0.88
0
50
100
150
k (#latent Dimensions)
SGD KNN256
SGD KNN256++
BFM KNN256
BFM KNN256++
SGD PMF
BPMF
BFM (u,i)
BFM (u,i,t,f)
BFM (u,i,ut,it,if)
BFM Ensemble

Datasets

- Yelp challenge
- Yelp datasets
- Users information
- Social information
- Location
- Time
- Ratings
- Reviews

References

- R. Salakhutdinov and A. Mnih, "Bayesian probabilistic matrix factorization using markov chain monte carlo," in *Proc. of ICML*, pp. 880–887, 2008.
- R. Salakhutdinov and A. Mnih, "Probabilistic matrix factorization," in *Proc. of NIPS*, 2007.
- L. Xiong, X. Chen, T. K. Huang, J. Schneider, and J. G. Carbonell, "Temporal collaborative filtering with bayesian probabilistic tensor factorization," in *Proc. of SDM*, 2010.
- Y. Koren, "Factorization meets the neighborhood: A multifaceted collaborative filtering model," in *Proc. of KDD*, pp. 426–434, 2008.
- Y. Koren, "Collaborative filtering with temporal dynamics," in *Proc. of KDD*, pp. 447–456, 2009.
- S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme, "Factorizing personalized markov chains for next-basket recommendation," in *Proc. of WWW*, pp. 811–820, 2010.
- S. Rendle, "Factorization machines," in *Proc. of ICDM*, 2010.

THANK YOU