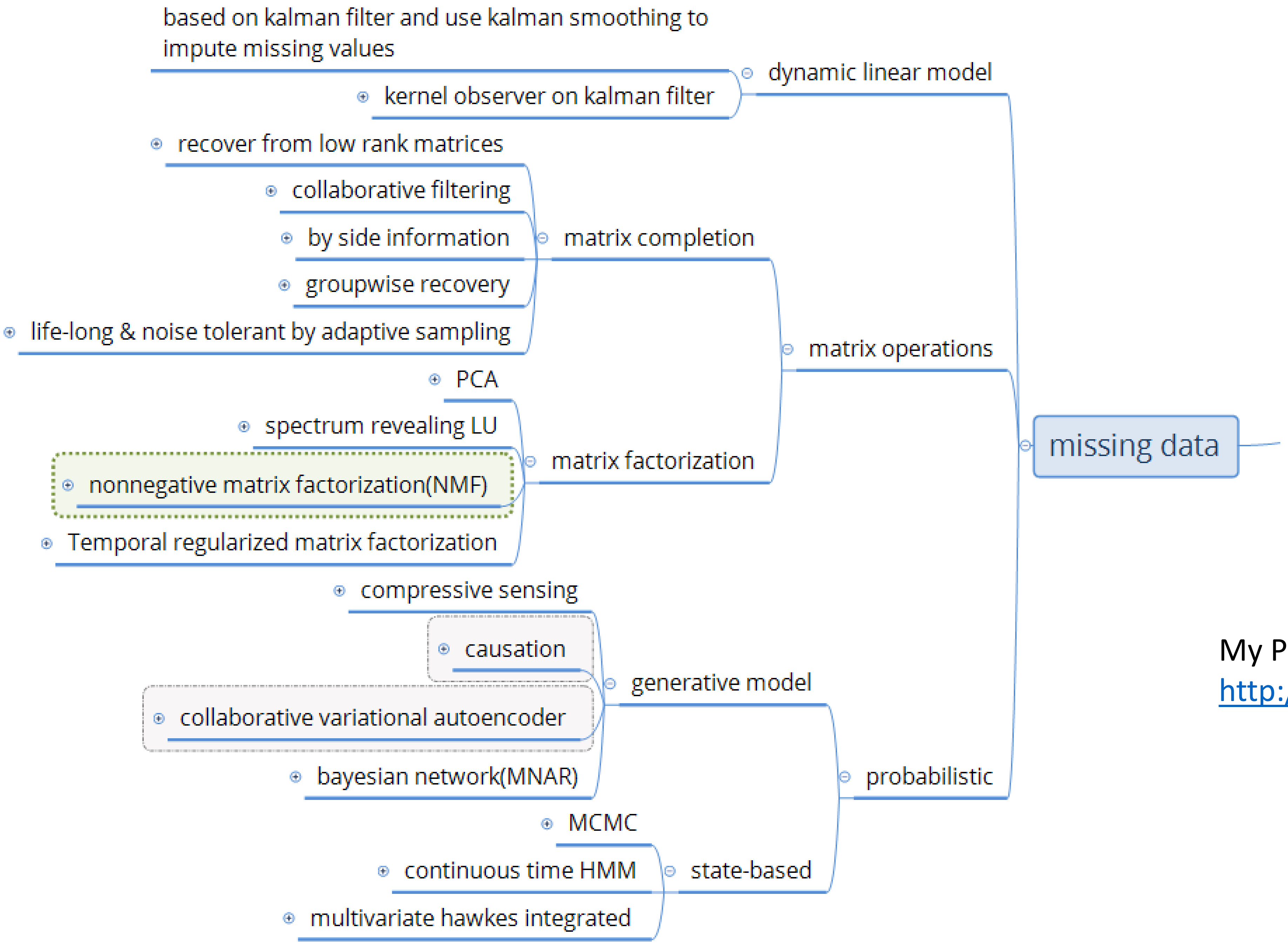


My Previous Review on
http://www.xmind.net/m/fPfe

Highlights

- Non-Negative Matrix Factorization
- Group-wise Recovery
- Collaborative filtering by Autoencoders
  - collaborative recurrent autoencoder
  - collaborative variational autoencoder
- Collaborative filtering by side information
  - by interactive
  - by co-embedding
- Learning Bayesian Network by Approximate Algorithm

Non-negative Matrix Factorization(NMF)

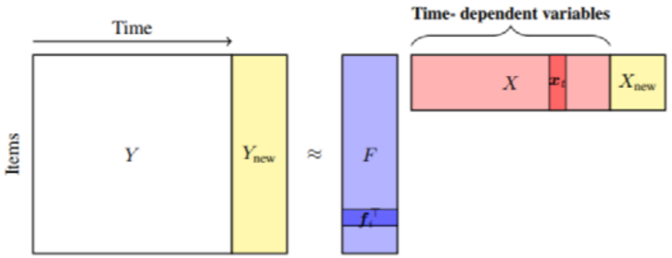
Non-negative Matrix Factorization(NMF)

- Nonnegative Matrix Factorization: Decompose to Y ≈ WH
  - each data can be written as a linear combination of columns of W with weights in H
  - constraint: Y, W, H are all positive
  - i.e. the loss is

$$\min_{\mathbf{V},\mathbf{W},\mathbf{H}} \quad \ell(\mathbf{V},\mathbf{W},\mathbf{H}) = \|\mathbf{V}-\mathbf{WH}\|_F^2$$

$$\text{s.t.} \quad \mathbf{V} \geq \mathbf{0}, \quad \mathbf{W} \geq \mathbf{0}, \quad \mathbf{H} \geq \mathbf{0},$$

  - updating procedure is iterative for missing data



NMF with only temporal aggregates

- http://proceedings.mlr.press/v70/mei17a/mei17a.pdf
- Usage scenario: electricity balancing as supply should be as much electricity as consumers consume
- inputs are time aggregates from each users but data is sometimes missing
- Its focus: projecting time aggregates and constraining temporal autocorrelations by penalizing imposed <u>temporal autocorrelation</u>

$$\min_{\mathbf{V},\mathbf{W},\mathbf{H}} \quad \|\mathbf{V}-\mathbf{W}\mathbf{H}\|_F^2 - \lambda \sum_{n=1}^{N} \mathbf{v}_n' \boldsymbol{\Delta}_{\rho_n} \mathbf{v}_n$$

$$\text{s.t.} \quad \mathbf{V} \geq 0, \quad \mathbf{W} \geq 0, \quad \mathbf{H} \geq 0, \quad \mathcal{A}(\mathbf{V}) = \mathbf{a}, \tag{4}$$

By extra term, we impose a threshold of v_n to be at least equal to ρ_n

- Then by updating W and H, they impute missing values by projection rule:

while Stopping criterion is not satisfied **do**

$\mathbf{W}^{i+1} = \mathrm{Update}(\mathbf{W}^{i}, \mathbf{H}^{i}, \mathbf{V}^{i})$

$\mathbf{H}^{i+1} = \mathrm{Update}(\mathbf{W}^{i+1}, \mathbf{H}^{i}, \mathbf{V}^{i})$

for all $1 \leq n \leq N$ **do**

$\mathbf{v}_n^{i+1} = (\mathbf{Q}_n \mathbf{c}_n + (\mathbf{I} - \mathbf{Q}_n \mathbf{A}_n)(\mathbf{I} - \lambda \mathbf{\Delta}_{\rho_n})^{-1} \mathbf{W}^{i+1} \mathbf{h}_n^{i+1})_+$ I think the + should be original v_n, they made a typo in pseudocode.

end for

$i = i + 1$

end while

Experiments and Results

- NeNMF with penalization works best for all the updating rules. By original NeNMF
- Complexity: by original NeNMF paper,

$$\text{NeNMF} \qquad O(mnr + mr^2 + nr^2) + K \times O(mr^2 + nr^2)$$

+ one matrix inversion on each step

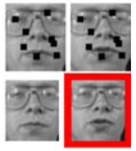
Other Improvements

- Poisson model Bayesian structure
  - a Bayesian treatment of the Poisson model with Gamma conjugate priors on the latent factors
  - makes assumption that, if an entry is not missing, then its value is 1 with a high probability
  - updating rule is similar to other probabilistic models
  - http://proceedings.mlr.press/v48/basbug16.html
- Noise-robust
  - adding a ReLU in updating step to make matrix noise-robust
  - https://papers.nips.cc/paper/6417-recovery-guarantee-of-non-negative-matrix-factorization-via-alternating-updates

Groupwise Recovery

Groupwise Recovery

- https://papers.nips.cc/paper/6357-high-rank-matrix-completion-and-clustering-under-self-expressive-models
- The main idea is to formulate groups based on pairwise similarities and use those similarity to complete the missing data.
- Experiment
  - Image Recovery
  - Motion Segmentation of Videos



- Complexity
  - although word “efficient” appears 13 times in this paper, there is no closed form analysis in complexity.
- Details(next page)

- **Sparse Subspace Clustering** relies on finding vectors from the same subspace(self expressiveness); a similarity graph of vectors can be built on weight of edge $w_{ij} = |c_{ij}| + |c_{ji}|$

$$\min_{\{c_{1j},\dots,c_{Nj}\}} \sum_{i=1}^{N} |c_{ij}| \quad \text{s.t.} \quad \sum_{i=1}^{N} c_{ij}\boldsymbol{y}_i = \boldsymbol{0}, \;\; c_{jj} = -1$$

- To take into account missing data, they add indicator function **I**, along with matrices $\boldsymbol{U}, \boldsymbol{\alpha}$ and get

$$\min_{\substack{\{c_{ij}\},\{\boldsymbol{\alpha}_{ij}\}\\ c_{jj}=-1,\forall j}} \sum_{j=1}^{N}\sum_{i=1}^{N} \mathrm{I}\left(\left\|\begin{bmatrix} c_{ij} \\ \boldsymbol{\alpha}_{ij}\end{bmatrix}\right\|_p\right) \text{ s.t. } \sum_{i=1}^{N} \begin{bmatrix}\bar{\boldsymbol{y}}_i & \boldsymbol{U}_{\Omega_i^c}\end{bmatrix}\begin{bmatrix} c_{ij} \\ \boldsymbol{\alpha}_{ij}\end{bmatrix} = \boldsymbol{0}, \mathrm{rk}\left(\begin{bmatrix} c_{i1} & \cdots & c_{iN} \\ \boldsymbol{\alpha}_{i1} & \cdots & \boldsymbol{\alpha}_{iN}\end{bmatrix}\right) = 1, \forall i, j$$

- To optimize this objective(which is non-convex), they use nuclear-norm relaxation and impute each missing entry by finding the best rank-one factorization.

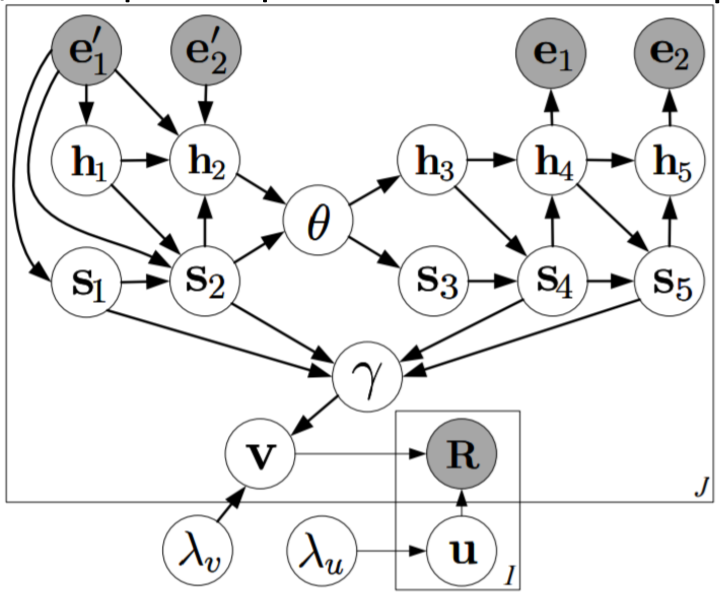
Collaborative Filtering with Generative Model

Collaborative Recurrent Autoencoder

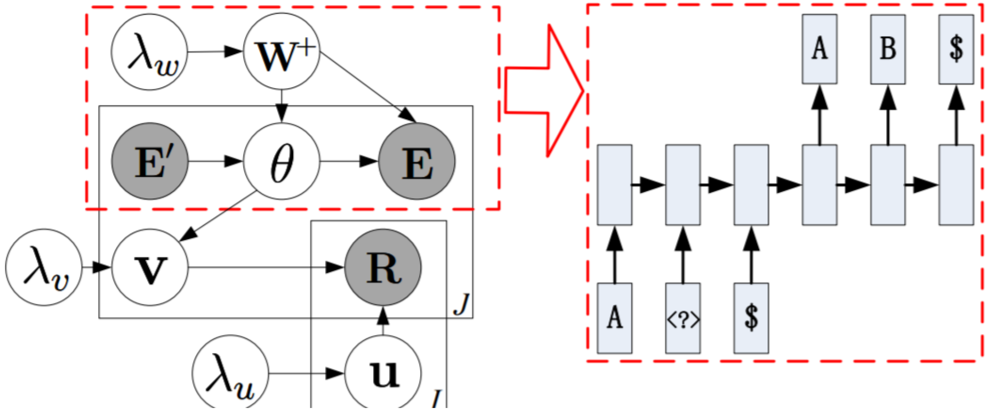
- online prediction model based on RNN
  - which can capture
    - sequential order
    - implicit correlation between item & users(rows and columns)
- Experiments:
  - Netflix recommendation system dataset
  - claims to have better recall than state of art
- https://papers.nips.cc/paper/6163-collaborative-recurrent-autoencoder-recommend-while-learning-to-fill-in-the-blanks.pdf

- Basic structure

noise/corrupted input decoded output



latent vector and hyperparameters



<?>wildcard, this wildcard represents missing values, which is learned without any consequences; it can also be used to denoise and prevent overfitting

Features beyond RNN

- Hierarchical Bayesian nature of the model
  - framework: encode -> compress -> depress -> decode => beta pooling and recommend
- <Wildcard> to Denoise & represent missing value
- Beta Pooling:
  - a weighted average of many vectors and get a beta distribution
  - establish a Beta distribution and pool from it
  - to prevent overfitting and more efficiently factorizing matrix

Collaborative Variational Autoencoder

- It is a Bayesian generative model that learns from both rating and content(both data matrix and side information)
  - thus it jointly learn latent representation on content and implicit relationships
- Architecture(next page)
- Optimization
  - measuring KL divergence and, by Stochastic Gradient Variational Bayes, we can learn it by backpropagation (thus more efficient than learning it by MCMC)
- https://dl.acm.org/citation.cfm?id=3098077

Architecture

- What it has is a model with inference network and generation network

(1) For each layer l of the inference network
  (a) For each column n of the weight matrix W_l, draw

$$W_{l,*n} \sim \mathcal{N}(0, \lambda_w^{-1} I_{K_l}).$$

  (b) Draw the bias vector $b_l \sim \mathcal{N}(0, \lambda_w^{-1} I_{K_l})$.
  (c) For each row j of h_l, draw

$$h_{l,j*} \sim \mathcal{N}(\sigma(h_{l-1,j*} W_l + b_l), \lambda_s^{-1} I_{K_l}).$$

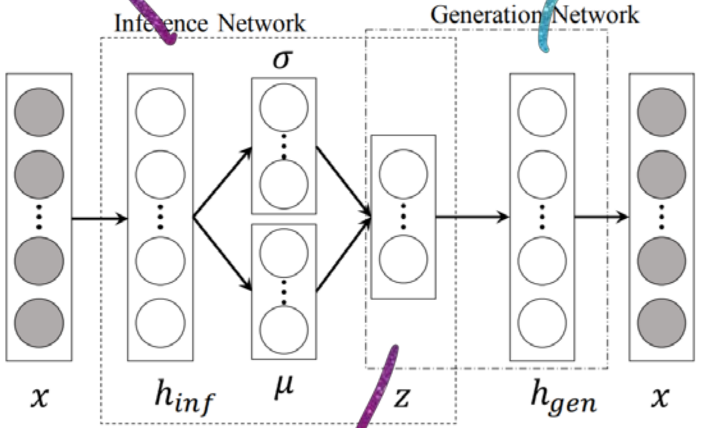
(2) For each item j
  (a) Draw latent mean and covariance vector

$$\mu_j \sim \mathcal{N}(h_L W_\mu + b_\mu, \lambda_s^{-1} I_K)$$

$$\log \sigma_j^2 \sim \mathcal{N}(h_L W_\sigma + b_\sigma, \lambda_s^{-1} I_K).$$

  (b) Draw latent content vector

$$z_j \sim \mathcal{N}(\mu_j, \mathrm{diag}(\sigma_j)).$$



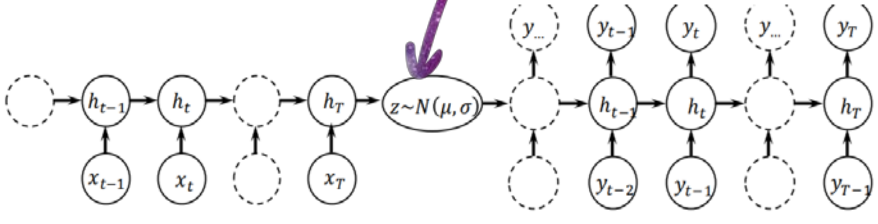
(1) For each layer l of the generation network
  (a) For each column n of the weight matrix W_l, draw

$$W_{l,*n} \sim \mathcal{N}(0, \lambda_w^{-1} I_{K_l}).$$

  (b) Draw the bias vector $b_l \sim \mathcal{N}(0, \lambda_w^{-1} I_{K_l})$.
  (c) For each row j of h_l, draw

$$h_{l,j*} \sim \mathcal{N}(\sigma(h_{l-1,j*} W_l + b_l), \lambda_s^{-1} I_K).$$

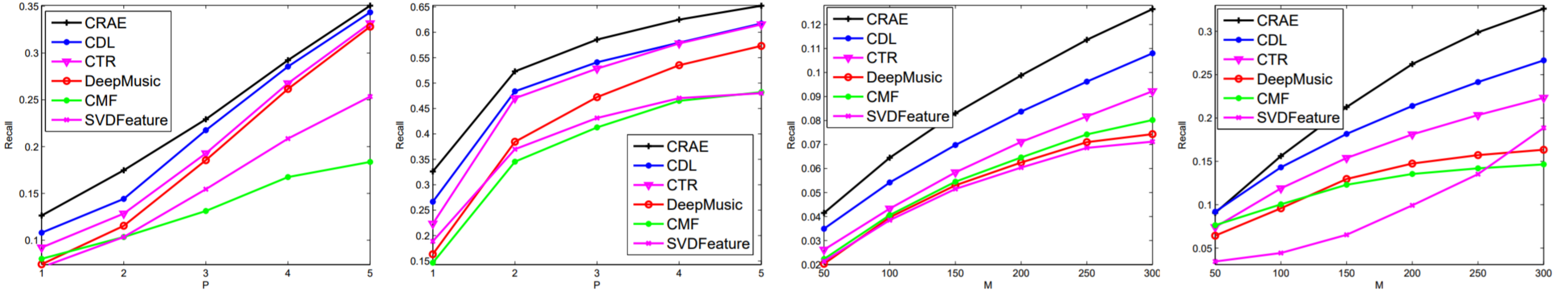
latent variable =
latent collaborative variable
+ latent content variable(side info)

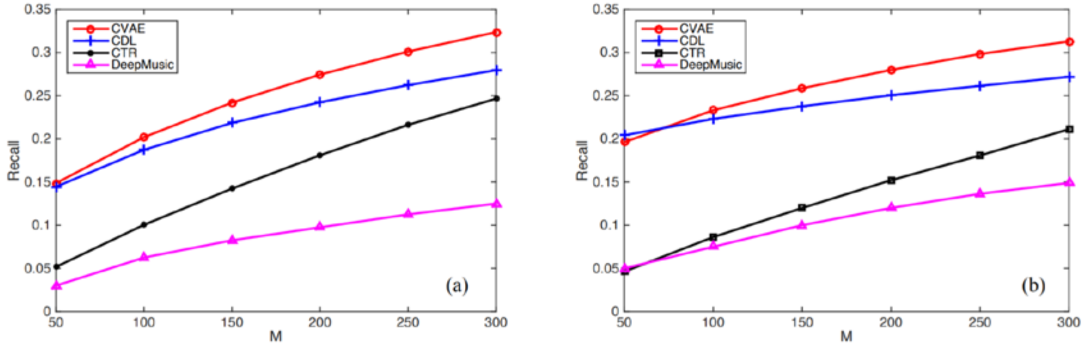


structure of sequential autoencoder
(our case)

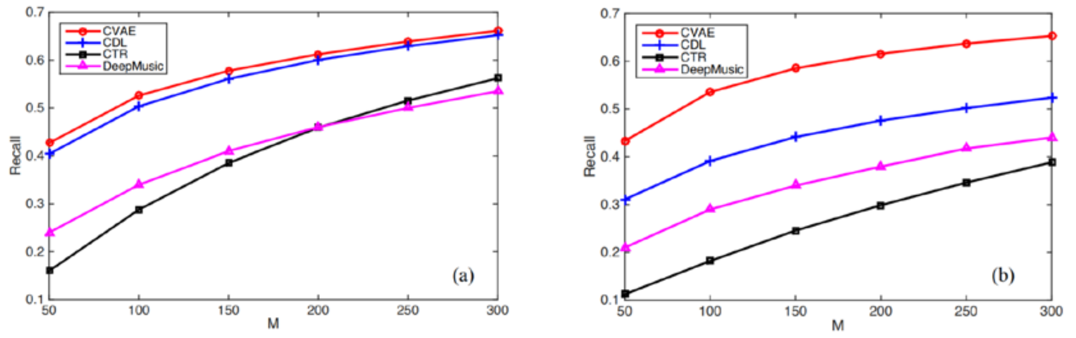
- Experiment
  - still use CiteULike dataset
  - Similar Baseline models and still better than state of art

First one: RNN





gure 4: Performance comparison of CVAE, CDL, CTR and DeepMusic based on recall in the sparse settin *teulike-a* **and (b)** *citeulike-t***.**



Performance comparison of CVAE, CDL, CTR and DeepMusic based on recall in the dense setting for dataset (a) *a* **and (b)** *citeulike-t***.**

Second one: Variational

Collaborative Filtering With Side Information

Usage of Side Information

- Usually, we approximate **F** by partially observed data(R is partially seen function)

$$\min_{\mathbf{E}} \|\mathbf{E}\|_*, \quad \text{subject to} \quad R_\Omega(\mathbf{E}) = R_\Omega(\mathbf{F}),$$

by minimizing its nuclear norm

- With side information, what we have is **X Y** as side feature matrix, we can approximate $\boldsymbol{F} = \boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{G}\boldsymbol{Y}$ given $\mathrm{R}(\boldsymbol{F}) = R\left(\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{G}\boldsymbol{Y}\right)$.

Interactive Model with side information

- (notations follows from last slide)
- Their prediction function is
  - $f = \boldsymbol{x^T H y} + x^T u + y^T v + g$, where $x^T \boldsymbol{H} y$ is their interactive term between **X** and **Y.** The rest of terms are linear model. Then they rewrite the whole term as G. Then their objective becomes

$$\min_{\mathbf{G},\mathbf{E}} \quad \frac{1}{2}\|\mathbf{X}^T\mathbf{G}\mathbf{Y} - \mathbf{E}\|_F^2 + \lambda_G g(\mathbf{G}) + \lambda_E \|\mathbf{E}\|_*, \qquad \text{subject to} \quad R_\Omega(\mathbf{E}) = R_\Omega(\mathbf{F}).$$

  - They take into account noise, nuclear norm of completed matrix and above linear model at same time

Results

- Sample Complexity for is log(N), N is whole matrix
- They developed an adaptive sampling to optimize above objective
- They proposed a Linear ADMM takes O(1/t) iterations to converge than vanilla ADMM. The worst case sampling complexity is $O(n^{\frac{3}{2}})$
- All their datasets used(MovieLens and NCI-DREAM challenge) are for recommendation system
- https://papers.nips.cc/paper/6265-a-sparse-interactive-model-for-matrix-completion-with-side-information.pdf

Convex Co-embedding For Matrix Completion

- https://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14286/14360
- Mainly solves incomplete multi-label problem
  - can complete matrix by linear prediction model
- They have a similar objective function

$$\min_{Z,B,W,\mathbf{b}} \ \ell(f(X), ZB^\top) + \frac{\alpha}{2}\|W\|_F^2 + \frac{\gamma}{2}(\|Z\|_F^2 + \|B\|_F^2)$$

  - where the only difference from last objective is (last term) they decompose nuclear norm into a sum of two latent matrices

(continued next page)

- They propose a simplistic primal gradient descent

Repeat

1. Set $t = t + 1$

2. Update: $M^{(t)} = \mathcal{P}_{\eta^*}(Q^{(t)}), \quad \beta_{t+1} = \frac{1+\sqrt{1+4\beta_t^2}}{2}$,

$Q^{(t+1)} = M^{(t)} + \left(\frac{\beta_t - 1}{\beta_{t+1}}\right)(M^{(t)} - M^{(t-1)})$

Until Converge

to solve the previous problem, and have a convergence rate of $O\left(\frac{1}{t^2}\right)$, which is faster than the previous method.

- Experiment
  - Recommendation system
  - Incomplete multilabel learning
    - Yahoo Web Page Classification
      - treats labels of webpages as target completion matrix

Learning Bayesian Network By EM

Structural EM(Classical Method)

- begins with an initial graph structure
- **Maximize**: the probability distribution of variables with missing values is estimated by EM
- **Expect**: score of each neighboring graph is computed. After convergence, the graph maximizing the score is chosen.
- Missing data is imputed in a separate step from EMing Bayesian structure

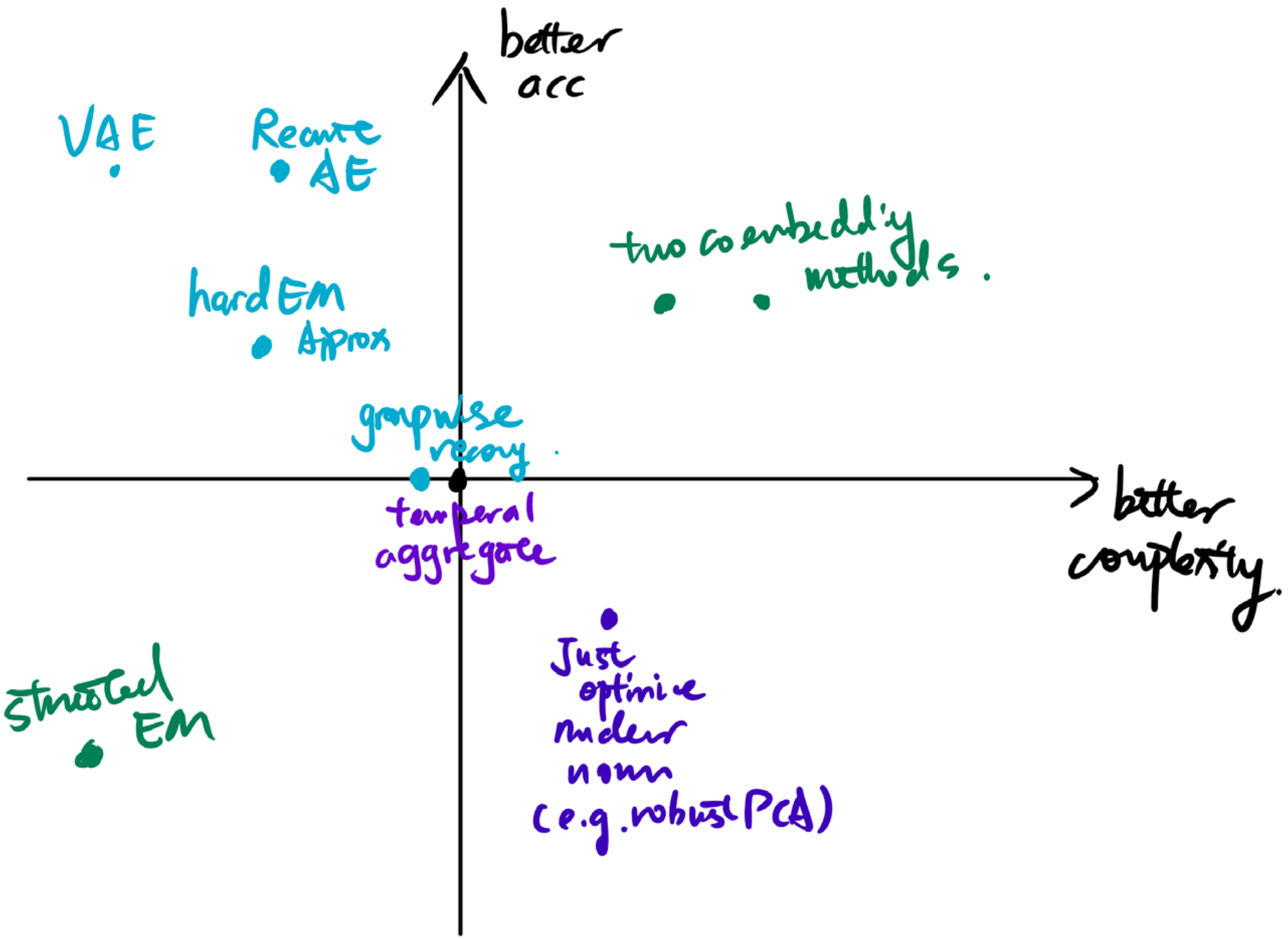
Approximate EM

- https://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14711
- By experiments of this paper, previous structural EM has **very low accuracy** when data do not miss at random(**NMAR**)

# missing values	Algorithm	Avg. Accuracy
25	Approx. Learning	**90%**
	Structural EM	15%
	Laplacian SVM	76%
50	Approx. Learning	**88%**
	Structural EM	10%
	Laplacian SVM	73.5%
75	Approx. Learning	**88%**
	Structural EM	8%
	Laplacian SVM	76%

Approximate Algorithm

- Exact algorithm is done by considering all possible completions **Z** and compute scores for each one of them.
  - Because **parent set identification** is NP-complete, it has an assumed bound for parents sets
  - then creates gadgets that is related to missing values
  - maximize score by considering all possible values
- Approximate Algorithm has a limitation on performing at most t completions at a time(then as proved, it will converge to a t-locally optimal).
  - Complexity is $O(RmC)$ for R is possible realization of parent set, m nodes, C missing values; it also needs $O(nk(Rm)^t)$ per parent set evaluation.



(Potentially Misleading Comparison of complexity)