Matrix and Tensor Factorization from a Machine Learning Perspective

Christoph Freudenthaler

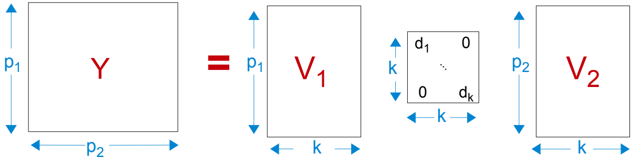
Information Systems and Machine Learning Lab, University of Hildesheim

Research Seminar, Vienna University of Economics and Business, January 13, 2012

Deutsche Forschungsgemeinschaft DFG

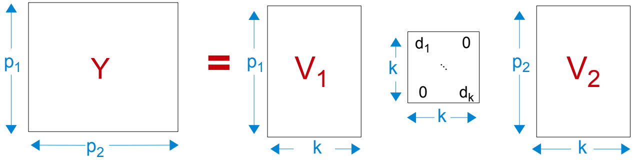
Matrix Factorization - SVD

- **SVD:** Decompose $p_1 \times p_2$ matrix $Y := V_1 \, D \, V_2^T$
  - V_1 are k eigenvectors of YY^T
  - V_2 are k eigenvectors of $Y^T Y$
  - $D := \sqrt{eig(diag(YY^T))}$



Matrix Factorization - SVD

- **SVD:** Decompose $p_1 \times p_2$ matrix $Y := V_1\, D\, V_2^T$
  - V_1 are k eigenvectors of YY^T
  - V_2 are k eigenvectors of Y^TY
  - $D := \sqrt{eig(diag(YY^T))}$

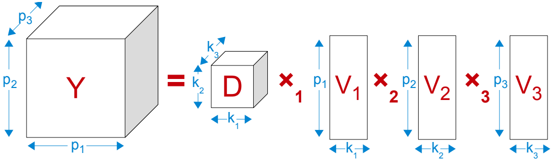


Properties:

- Method from linear algebra for arbitrary Y
- Tool for descriptive and exploratory data analysis
- Decomposition optimizes the Frobenius norm with orthogonality constraints

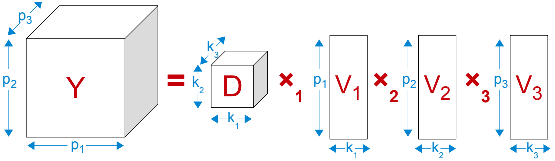
Tensor Factorization - Tucker Decomposition

- **Tucker Decomposition:**
  Decompose $p_1 \times p_2 \times p_3$ tensor $Y := D \times_1 V_1 \times_2 V_2 \times_3 V_3$
  - V_1 are k_1 eigenvectors of mode-1 unfolded Y
  - V_2 are k_2 eigenvectors of mode-2 unfolded Y
  - V_3 are k_3 eigenvectors of mode-3 unfolded Y
  - $D \in \mathbb{R}^{k_1 \times k_2 \times k_3}$ non-diagonal core tensor



Tensor Factorization - Tucker Decomposition

- **Tucker Decomposition:**
  Decompose $p_1 \times p_2 \times p_3$ tensor $Y := D \times_1 V_1 \times_2 V_2 \times_3 V_3$
  - V_1 are k_1 eigenvectors of mode-1 unfolded Y
  - V_2 are k_2 eigenvectors of mode-2 unfolded Y
  - V_3 are k_3 eigenvectors of mode-3 unfolded Y
  - $D \in \mathbb{R}^{k_1 \times k_2 \times k_3}$ non-diagonal core tensor

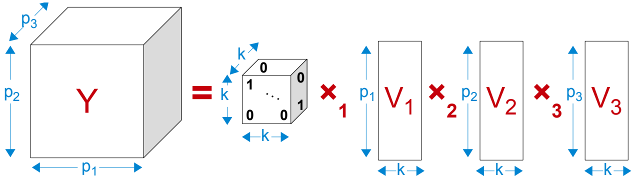


Properties:

- Extension of SVD method for arbitrary tensor $Y \rightarrow$ orthogonal representation
- Tool for descriptive and exploratory data analysis
- Decomposition optimizes the Frobenius norm
- Expensive inference: sequences of SVD + core tensor D

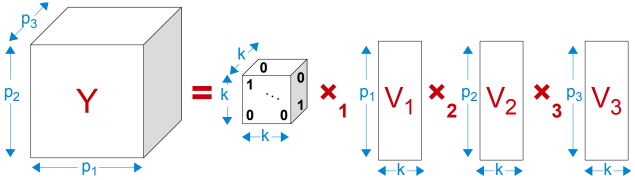
Tensor Factorization - Canonical Decomposition

- **Canonical Decomposition (CD):**
  Decompose $p_1 \times p_2 \times p_3$ tensor $Y := D \times_1 V_1 \times_2 V_2 \times_3 V_3$
  - with diagonal, identity tensor D
  - as a sum of k rank-one tensors $Y = \sum_{f=1}^{k} \mathbf{v}_{1,f} \circ \mathbf{v}_{2,f} \circ \mathbf{v}_{3,f}$



Tensor Factorization - Canonical Decomposition

- **Canonical Decomposition (CD):**
  Decompose $p_1 \times p_2 \times p_3$ tensor $Y := D \times_1 V_1 \times_2 V_2 \times_3 V_3$
  - with diagonal, identity tensor D
  - as a sum of k rank-one tensors $Y = \sum_{f=1}^{k} \mathbf{v}_{1,f} \circ \mathbf{v}_{2,f} \circ \mathbf{v}_{3,f}$



Properties:

- Extension of SVD method for arbitrary tensor Y
- Tool for descriptive and exploratory data analysis
- Decomposition optimizes the Frobenius norm
- Fast inference due to less parameters

Machine Learning Perspective

Machine Learning:

...is the task of

- learning from (noisy) experience **E** with respect to some class of tasks **T** and performance measure **P**
  - Experience **E**: data
  - Tasks **T**: predictions
  - Performance **P**: root mean square error (RMSE), misclassification,...

Machine Learning Perspective

Machine Learning:

...is the task of

- learning from (noisy) experience **E** with respect to some class of tasks **T** and performance measure **P**
  - Experience **E**: data
  - Tasks **T**: predictions
  - Performance **P**: root mean square error (RMSE), misclassification,...
- to **generalize** from noisy data to accurately **predict** new cases

Machine Learning Perspective

Machine Learning:

...is the task of

- learning from (noisy) experience **E** with respect to some class of tasks **T** and performance measure **P**
  - Experience **E**: data
  - Tasks **T**: predictions
  - Performance **P**: root mean square error (RMSE), misclassification,...
- to **generalize** from noisy data to accurately **predict** new cases

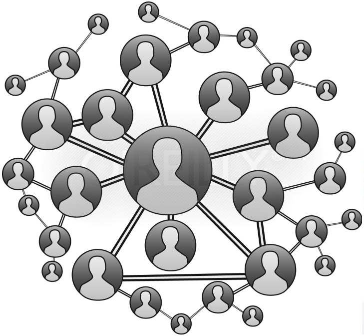
In other words:

- **Prediction accuracy on new cases** = Machine Learning
- **No hypothesis generation/testing** = Data Mining

Applications of the Machine Learning Perspective

Many different applications:

- Social Network Analysis
- Recommender Systems
- Graph Analysis
- Image/Video Analysis
- ...



Applications of the Machine Learning Perspective

Overall Task:

- **Prediction** of missing friendships, interesting items, corrupted pixels, missing edges, etc.
- **Data representation:** matrix/tensor with missing entries

Y =

	i_1	i_2	i_3	...	i_{p_2-1}	i_{p_2}
u_1	3					5
u_2	3	4	4			
...						
u_{p_1}		5			2	2

Applications of the Machine Learning Perspective

Overall Task:

- **Prediction** of missing friendships, interesting items, corrupted pixels, missing edges, etc.
- **Data representation:** matrix/tensor with missing entries

Y =

	i_1	i_2	i_3	...	i_{p_2-1}	i_{p_2}
u_1	3					5
u_2	3	4	4			
...						
u_{p_1}		5			2	2

Further Properties:

- **Unobserved Heterogeneity,** e.g. different consumption preferences of different users and items
- **Large-scale:** millions of interactions
- **Poor Data Quality:** high noise due to indirect data collection

$\rightarrow$ Factorization Models

Factorization Models from a Machine Learning Perspective

Common usage of matrix (and tensor) factorization:

- Identification/Interpretation of unobserved heterogeneity, i.e. latent dependencies between instances of a mode (rows, columns,...)
- Data compression, e.g., instead of $p_1 p_2$ values, store only $(p_1 + p_2)k$ values
- Data preprocessing: uncorrelate p predictor variables of a design matrix X

Factorization Models from a Machine Learning Perspective

Machine Learning perspective on factorization models:

- Factorization models seen as predictive models
  - No probabilistic embedding, e.g. for Bayesian Analysis
  - Gaussian likelihood:

$$Y = V_1 \, D \, V_2^T + E, \qquad \forall e_\ell \in E : e_\ell \sim N(0, \sigma^2 = 1)$$

  - No missing value treatment; often treated as 0

Factorization Models from a Machine Learning Perspective

Machine Learning perspective on factorization models:

- Factorization models seen as predictive models
  - No probabilistic embedding, e.g. for Bayesian Analysis
  - Gaussian likelihood:

$$Y = V_1 \, D \, V_2^T + E, \qquad \forall e_\ell \in E : e_\ell \sim N(0, \sigma^2 = 1)$$

  - No missing value treatment; often treated as 0
  - Non-informative prior:

$$p(V_1, D, V_2) \propto 1$$

  - Does not distinguish between signal and noise

Factorization Models from a Machine Learning Perspective

Machine Learning perspective on factorization models:

- Factorization models seen as predictive models
  - No probabilistic embedding, e.g. for Bayesian Analysis
  - Gaussian likelihood:

$$Y = V_1 \, D \, V_2^T + E, \qquad \forall e_\ell \in E : e_\ell \sim N(0, \sigma^2 = 1)$$

  - No missing value treatment; often treated as 0
  - Non-informative prior:

$$p(V_1, D, V_2) \propto 1$$

  - Does not distinguish between signal and noise
- No interest in latent representation, e.g. interpretation
  - Elimination of orthonormality constraint $Y = V_1, V_2^T + E$
  - $\rightarrow$ for general tensors:

$$Y = \sum_{f=1}^{k} \mathbf{v}_{1,f} \circ \mathbf{v}_{2,f} \circ \ldots \circ \mathbf{v}_{m,f} + E$$

Factorization Models from a Machine Learning Perspective

Machine Learning perspective on factorization models:

- Factorization models seen as predictive models
  - No probabilistic embedding, e.g. for Bayesian Analysis
  - Gaussian likelihood:

$$Y = V_1 \, D \, V_2^T + E, \qquad \forall e_\ell \in E : e_\ell \sim N(0, \sigma^2 = 1)$$

  - No missing value treatment; often treated as 0
  - Non-informative prior:

$$p(V_1, D, V_2) \propto 1$$

  - Does not distinguish between signal and noise
- No interest in latent representation, e.g. interpretation
  - Elimination of orthonormality constraint $Y = V_1, V_2^T + E$
  - $\rightarrow$ for general tensors:

$$Y = \sum_{f=1}^{k} \mathbf{v}_{1,f} \circ \mathbf{v}_{2,f} \circ \ldots \circ \mathbf{v}_{m,f} + E$$

Related Work

Existing Extensions of Factorization Models:

- More general likelihood for multinomial, count data, etc.
  → Exponential Family PCA[1]
- Inference only on observed tensor entries

[1] Mohamed, S., Heller, K. A., Ghahramani, Z.: Exponential Family PCA, NIPS08.

[2] Tipping, M. E., Bishop, C. M.: Probabilistic PCA, Journal of the Royal Statistical Society.

[3] Koren, Y.: Factorization meets the neighborhood: a multifaceted collaborative filtering model, KDD08.

Related Work

Existing Extensions of Factorization Models:

- More general likelihood for multinomial, count data, etc.
  → Exponential Family PCA[1]
- Inference only on observed tensor entries
- Introduction of prior distributions[2]

[1] Mohamed, S., Heller, K. A., Ghahramani, Z.: Exponential Family PCA, NIPS08.

[2] Tipping, M. E., Bishop, C. M.: Probabilistic PCA, Journal of the Royal Statistical Society.

[3] Koren, Y.: Factorization meets the neighborhood: a multifaceted collaborative filtering model, KDD08.

Related Work

Existing Extensions of Factorization Models:

- More general likelihood for multinomial, count data, etc.
  → Exponential Family PCA[1]
- Inference only on observed tensor entries
- Introduction of prior distributions[2]
- Scalable Learning algorithms for large scale data: Gradient Descent Models[3]

[1] Mohamed, S., Heller, K. A., Ghahramani, Z.: Exponential Family PCA, NIPS08.
[2] Tipping, M. E., Bishop, C. M.: Probabilistic PCA, Journal of the Royal Statistical Society.
[3] Koren, Y.: Factorization meets the neighborhood: a multifaceted collaborative filtering model, KDD08.

Related Work

Existing Extensions of Factorization Models:

- More general likelihood for multinomial, count data, etc.
  $\rightarrow$ Exponential Family PCA[1]
- Inference only on observed tensor entries
- Introduction of prior distributions[2]
- Scalable Learning algorithms for large scale data: Gradient Descent Models[3]

Missing Extensions:

- Extensions of the predictive models, i.e. SVD, CD
- Comparison to standard predictive models

[1] Mohamed, S., Heller, K. A., Ghahramani, Z.: Exponential Family PCA, NIPS08.

[2] Tipping, M. E., Bishop, C. M.: Probabilistic PCA, Journal of the Royal Statistical Society.

[3] Koren, Y.: Factorization meets the neighborhood: a multifaceted collaborative filtering model, KDD08.

Outline

Generalized Factorization Model

Relations to standard Models

Empirical Evaluation

Conclusion and Discussion

Outline

Generalized Factorization Model

Relations to standard Models

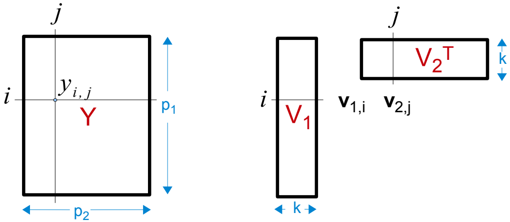
Empirical Evaluation

Conclusion and Discussion

Probabilistic SVD

Frobenius norm optimal:

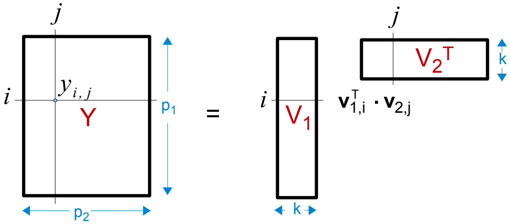
$$\underset{\theta=\{V_1,V_2\}}{\operatorname{argmin}} ||Y - V_1\, V_2^T||_F = \sum_{i=1}^{p_1}\sum_{j=1}^{p_2}(y_{i,j} - \mathbf{v}_{1,i}^T\mathbf{v}_{2,j})^2$$



Probabilistic SVD

Frobenius norm optimal:

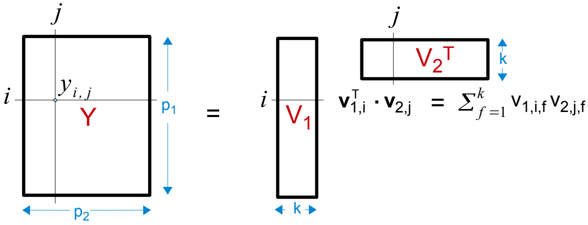
$$\underset{\theta=\{V_1,V_2\}}{\operatorname{argmin}} ||Y - V_1\, V_2^T||_F = \sum_{i=1}^{p_1} \sum_{j=1}^{p_2} (y_{i,j} - \mathbf{v}_{1,i}^T \mathbf{v}_{2,j})^2$$



Probabilistic SVD

Frobenius norm optimal:

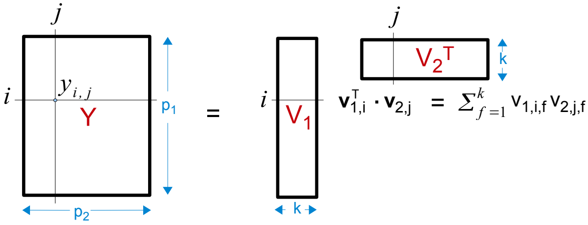
$$\underset{\theta=\{V_1,V_2\}}{\text{argmin}} ||Y - V_1 V_2^T||_F = \sum_{i=1}^{p_1}\sum_{j=1}^{p_2}(y_{i,j} - \mathbf{v}_{1,i}^T \mathbf{v}_{2,j})^2$$



Probabilistic SVD

Frobenius norm optimal:

$$\underset{\theta=\{V_1,V_2\}}{\operatorname{argmin}} ||Y - V_1\, V_2^T||_F = \sum_{i=1}^{p_1}\sum_{j=1}^{p_2}(y_{i,j} - \mathbf{v}_{1,i}^T\mathbf{v}_{2,j})^2$$



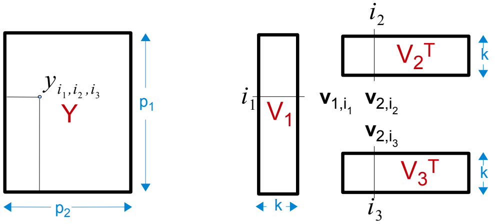
$\propto$ **Gaussian maximum likelihood:**

$$\underset{\theta=\{V_1,V_2\}}{\operatorname{argmax}} \prod_{i=1}^{p_1}\prod_{j=1}^{p_2} \exp\left(-\frac{1}{2}(y_{i,j} - \mathbf{v}_{1,i}^T\mathbf{v}_{2,j})^2\right)$$

Probabilistic CD - Extend SVD to arbitrary *m*

Frobenius norm optimal:

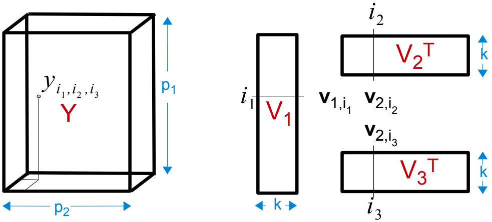
$$\underset{\theta=\{V_1,V_2,\ldots,V_m\}}{\operatorname{argmin}} \sum_{i_1=1}^{p_1}\sum_{i_2=1}^{p_2}\cdots\sum_{i_m=1}^{p_m}\Big(y_{i_1,i_2,\ldots,i_m} - \underbrace{\sum_{f=1}^{k} v_{1,i_1,f}\, v_{2,i_2,f} \ldots v_{m,i_m,f}}_{y^{CD}_{i_1,i_2,\ldots,i_m}}\Big)^2$$



Probabilistic CD - Extend SVD to arbitrary *m*

Frobenius norm optimal:

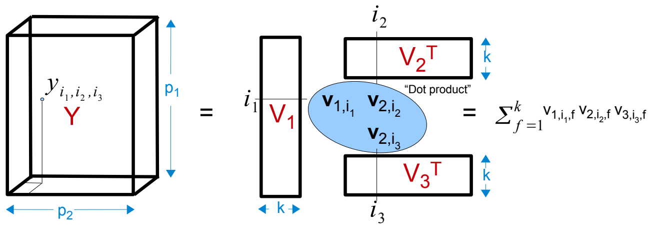
$$\underset{\theta=\{V_1,V_2,\ldots,V_m\}}{\operatorname{argmin}} \sum_{i_1=1}^{p_1}\sum_{i_2=1}^{p_2}\cdots\sum_{i_m=1}^{p_m}\Big(y_{i_1,i_2,\ldots,i_m} - \underbrace{\sum_{f=1}^{k} v_{1,i_1,f}\, v_{2,i_2,f}\cdots v_{m,i_m,f}}_{y^{CD}_{i_1,i_2,\ldots,i_m}}\Big)^2$$



Probabilistic CD - Extend SVD to arbitrary *m*

Frobenius norm optimal:

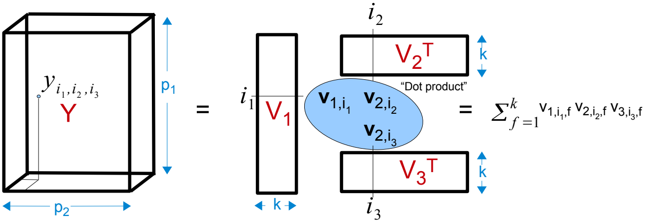
$$\underset{\theta=\{V_1,V_2,\ldots,V_m\}}{\operatorname{argmin}} \sum_{i_1=1}^{p_1}\sum_{i_2=1}^{p_2}\cdots\sum_{i_m=1}^{p_m}\Big(y_{i_1,i_2,\ldots,i_m} - \underbrace{\sum_{f=1}^{k} v_{1,i_1,f}\, v_{2,i_2,f} \ldots v_{m,i_m,f}}_{y^{CD}_{i_1,i_2,\ldots,i_m}}\Big)^2$$



Probabilistic CD - Extend SVD to arbitrary *m*

Frobenius norm optimal:

$$\underset{\theta=\{V_1,V_2,\ldots,V_m\}}{\operatorname{argmin}} \sum_{i_1=1}^{p_1}\sum_{i_2=1}^{p_2}\cdots\sum_{i_m=1}^{p_m}\Big(y_{i_1,i_2,\ldots,i_m} - \underbrace{\sum_{f=1}^{k} v_{1,i_1,f}\, v_{2,i_2,f}\cdots v_{m,i_m,f}}_{y^{CD}_{i_1,i_2,\ldots,i_m}}\Big)^2$$



$\propto$ Gaussian maximum likelihood:

$$\underset{\theta=\{V_1,V_2,\ldots\}}{\operatorname{argmax}} \prod_{i_1=1}^{p_1}\prod_{i_2=1}^{p_2}\cdots\prod_{i_m=1}^{p_m} \exp\left(-\frac{1}{2}\left(y_{i_1,i_2,\ldots,i_m} - y^{CD}_{i_1,i_2,\ldots,i_m}\right)^2\right)$$

GFM I: Predictive Model Interpretation

Adapt Notation:

- Introduce for each tensor element $\ell = 1, \ldots, n$ vector-valued indicator vectors $\mathbf{x}_{\ell,j}$, $j = 1, \ldots, m$ of length p_j

GFM I: Predictive Model Interpretation

Adapt Notation:

- Introduce for each tensor element $\ell = 1, \ldots, n$ vector-valued indicator vectors $\mathbf{x}_{\ell,j}$, $j = 1, \ldots, m$ of length p_j
- Form for each tensor element $y_{i_1,\ldots,i_m}$ a predictor vector $\mathbf{x}_\ell \in \mathbb{R}^{p=p_1+p_2+\ldots+p_m}$ by concatenating $\mathbf{x}_j$:

$$\mathbf{x}_\ell^{CD} = \left(\underbrace{0, \ldots, 1, \ldots, 0}_{\mathbf{x}_{\ell,1}}, \underbrace{0, \ldots, 1, \ldots, 0}_{\mathbf{x}_{\ell,2}}, \ldots \right)$$

GFM I: Predictive Model Interpretation

Adapt Notation:

- Introduce for each tensor element $\ell = 1, \ldots, n$ vector-valued indicator vectors $\mathbf{x}_{\ell,j}$, $j = 1, \ldots, m$ of length p_j
- Form for each tensor element $y_{i_1,\ldots,i_m}$ a predictor vector $\mathbf{x}_\ell \in \mathbb{R}^{p=p_1+p_2+\ldots+p_m}$ by concatenating $\mathbf{x}_j$:

$$\mathbf{x}_\ell^{CD} = \left(\underbrace{0, \ldots, 1, \ldots, 0}_{\mathbf{x}_{\ell,1}}, \underbrace{0, \ldots, 1, \ldots, 0}_{\mathbf{x}_{\ell,2}}, \ldots \right)$$

- Denote with $V = \{V_1, \ldots, V_m\}$ the set of all predictive model parameters

GFM I: Predictive Model Interpretation

Adapt Notation:

- Introduce for each tensor element $\ell = 1, \dots, n$ vector-valued indicator vectors $\mathbf{x}_{\ell,j}$, $j = 1, \dots, m$ of length p_j
- Form for each tensor element $y_{i_1,\dots,i_m}$ a predictor vector $\mathbf{x}_\ell \in \mathbb{R}^{p=p_1+p_2+\dots+p_m}$ by concatenating $\mathbf{x}_j$:

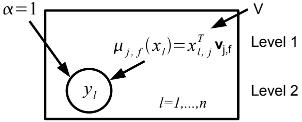
$$\mathbf{x}_\ell^{CD} = \left(\underbrace{0, \dots, 1, \dots, 0}_{\mathbf{x}_{\ell,1}}, \underbrace{0, \dots, 1, \dots, 0}_{\mathbf{x}_{\ell,2}}, \dots \right)$$

- Denote with $V = \{V_1, \dots, V_m\}$ the set of all predictive model parameters
- Rewrite $y^{CD}_{i_1,\dots,i_m}$ as $y^{CD}_{i_1,\dots,i_m} = y^{CD}_\ell = f(x^{CD}_\ell | V)$
- with general $f(x_\ell|V) = \sum_{f=1}^k \prod_{j=1}^m \mathbf{x}_{\ell,j}^T \mathbf{v}_{j,f} = \sum_{f=1}^k v_{1,i_1,f} v_{2,i_2,f} \dots v_{m,i_m,f}$

GFM I: Predictive Model Interpretation

More intuitive interpretation: 2-level hierarchical representation

$$y_\ell \sim N(\mu_\ell, \alpha = 1)$$

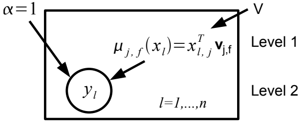


$$f(\mathbf{x}_\ell|V) = \mu_\ell = \sum_{f=1}^{k} \prod_{j=1}^{m} \underbrace{\mathbf{x}_{\ell,j}^T \mathbf{v}_{j,f}}_{\mu_{j,f}(\mathbf{x}_\ell)}$$

GFM I: Predictive Model Interpretation

More intuitive interpretation: 2-level hierarchical representation

$$y_\ell \sim N(\mu_\ell, \alpha = 1)$$



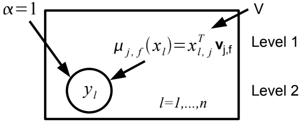
$$f(\mathbf{x}_\ell | V) = \mu_\ell = \sum_{f=1}^{k} \prod_{j=1}^{m} \underbrace{\mathbf{x}_{\ell,j}^T \mathbf{v}_{j,f}}_{\mu_{j,f}(\mathbf{x}_\ell)}$$

- each pair of mode j and latent dimension f has a different linear model $\mu_{j,f}(\mathbf{x}_\ell) = \mathbf{x}_{\ell,j}^T \mathbf{v}_{j,f}$
- with predictor vector $\mathbf{x}_{\ell,j}$ describing each mode j and p_j different model parameters $\mathbf{v}_{j,f}$ per mode j and latent dimension f

GFM I: Predictive Model Interpretation

More intuitive interpretation: 2-level hierarchical representation

$$y_\ell \sim N(\mu_\ell, \alpha = 1)$$



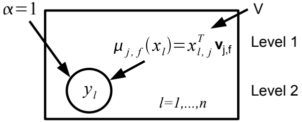
$$f(\mathbf{x}_\ell | V) = \mu_\ell = \sum_{f=1}^{k} \prod_{j=1}^{m} \underbrace{\mathbf{x}_{\ell,j}^T \mathbf{v}_{j,f}}_{\mu_{j,f}(\mathbf{x}_\ell)}$$

- each pair of mode j and latent dimension f has a different linear model $\mu_{j,f}(\mathbf{x}_\ell) = \mathbf{x}_{\ell,j}^T \mathbf{v}_{j,f}$
- with predictor vector $\mathbf{x}_{\ell,j}$ describing each mode j and p_j different model parameters $\mathbf{v}_{j,f}$ per mode j and latent dimension f
- $\rightarrow$ factorization models are 2-level hierarchical (multi-)linear models with
  - $\mu_{j,f}(\mathbf{x}_\ell)$ per latent dimension f and mode j in the upper level

GFM I: Predictive Model Interpretation

More intuitive interpretation: 2-level hierarchical representation

$$y_\ell \sim N(\mu_\ell, \alpha = 1)$$



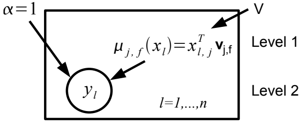
$$f(\mathbf{x}_\ell|V) = \mu_\ell = \sum_{f=1}^{k}\prod_{j=1}^{m}\underbrace{\mathbf{x}_{\ell,j}^T\mathbf{v}_{j,f}}_{\mu_{j,f}(\mathbf{x}_\ell)}$$

- each pair of mode j and latent dimension f has a different linear model $\mu_{j,f}(\mathbf{x}_\ell) = \mathbf{x}_{\ell,j}^T\mathbf{v}_{j,f}$
- with predictor vector $\mathbf{x}_{\ell,j}$ describing each mode j and p_j different model parameters $\mathbf{v}_{j,f}$ per mode j and latent dimension f
- $\rightarrow$ factorization models are 2-level hierarchical (multi-)linear models with
  - $\mu_{j,f}(\mathbf{x}_\ell)$ per latent dimension f and mode j in the upper level
  - modeled as linear functions of mode-dependent predictors $\mathbf{x}_{\ell,j}$

GFM I: Predictive Model Interpretation

More intuitive interpretation: 2-level hierarchical representation

$$y_\ell \sim N(\mu_\ell, \alpha = 1)$$



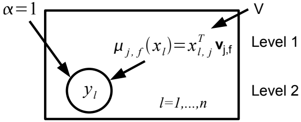
$$f(\mathbf{x}_\ell|V) = \mu_\ell = \sum_{f=1}^{k} \prod_{j=1}^{m} \underbrace{\mathbf{x}_{\ell,j}^T \mathbf{v}_{j,f}}_{\mu_{j,f}(\mathbf{x}_\ell)}$$

- each pair of mode j and latent dimension f has a different linear model $\mu_{j,f}(\mathbf{x}_\ell) = \mathbf{x}_{\ell,j}^T \mathbf{v}_{j,f}$
- with predictor vector $\mathbf{x}_{\ell,j}$ describing each mode j and p_j different model parameters $\mathbf{v}_{j,f}$ per mode j and latent dimension f
- $\rightarrow$ factorization models are 2-level hierarchical (multi-)linear models with
  - $\mu_{j,f}(\mathbf{x}_\ell)$ per latent dimension f and mode j in the upper level
  - modeled as linear functions of mode-dependent predictors $\mathbf{x}_{\ell,j}$
  - merged by the *dot product* $\sum_{f=1}^{k} \prod_{j=1}^{m} \mu_{j,f}$ to get $f(x_\ell|V)$

GFM I: Predictive Model Interpretation

More intuitive interpretation: 2-level hierarchical representation

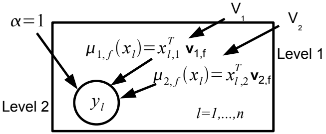
$$y_\ell \sim N(\mu_\ell, \alpha = 1)$$



$$f(\mathbf{x}_\ell|V) = \mu_\ell = \sum_{f=1}^{k} \prod_{j=1}^{m} \underbrace{\mathbf{x}_{\ell,j}^T \mathbf{v}_{j,f}}_{\mu_{j,f}(\mathbf{x}_\ell)}$$

- each pair of mode j and latent dimension f has a different linear model $\mu_{j,f}(\mathbf{x}_\ell) = \mathbf{x}_{\ell,j}^T \mathbf{v}_{j,f}$
- with predictor vector $\mathbf{x}_{\ell,j}$ describing each mode j and p_j different model parameters $\mathbf{v}_{j,f}$ per mode j and latent dimension f
- $\rightarrow$ factorization models are 2-level hierarchical (multi-)linear models with
  - $\mu_{j,f}(\mathbf{x}_\ell)$ per latent dimension f and mode j in the upper level
  - modeled as linear functions of mode-dependent predictors $\mathbf{x}_{\ell,j}$
  - merged by the *dot product* $\sum_{f=1}^{k} \prod_{j=1}^{m} \mu_{j,f}$ to get $f(x_\ell|V)$
  - and $y_\ell^{CD} = f(x_\ell^{CD}|V)$ using $\mathbf{x}_\ell^{CD}$

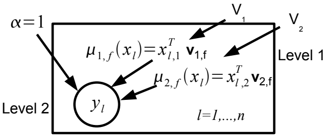
Important example: Matrix Factorization



Level 1:

- $\mu_{1,f}(\mathbf{x}_\ell) = \mathbf{x}_{\ell,1}^T \mathbf{v}_{1,f}$ (k row means)
- $\mu_{2,f}(\mathbf{x}_\ell) = \mathbf{x}_{\ell,2}^T \mathbf{v}_{2,f}$ (k column means)

Important example: Matrix Factorization



Level 1:

- $\mu_{1,f}(\mathbf{x}_\ell) = \mathbf{x}_{\ell,1}^T \mathbf{v}_{1,f}$ (k row means)
- $\mu_{2,f}(\mathbf{x}_\ell) = \mathbf{x}_{\ell,2}^T \mathbf{v}_{2,f}$ (k column means)

Level 2:

- $f(\mathbf{x}_\ell^{CD}|V) = \sum_{f=1}^{k} \mu_{1,f}(\mathbf{x}_\ell)\mu_{2,f}(\mathbf{x}_\ell) = \mathbf{v}_{1,f}^T \mathbf{v}_{2,f}$

GFM II: Predictive Model Extension

From $\mathbf{x}_\ell^{CD}$ to arbitrary $\mathbf{x}_\ell$:

- $\mathbf{x}_\ell = (\mathbf{x}_{\ell,1}, \ldots, \mathbf{x}_{\ell,m})$ still consists of m subvectors for each mode

GFM II: Predictive Model Extension

From $\mathbf{x}_\ell^{CD}$ to arbitrary $\mathbf{x}_\ell$:

- $\mathbf{x}_\ell = (\mathbf{x}_{\ell,1}, \ldots, \mathbf{x}_{\ell,m})$ still consists of m subvectors for each mode
- While $\mathbf{x}_{\ell,j} \in \mathbf{x}_\ell^{CD}$ has exactly one active position, thus $p_j - 1$ zero values, e.g.,

$$\mathbf{x}_{\ell,1} = (\underbrace{1}_{1^{st}\,row}, 0, \ldots, 0) \qquad \mathbf{x}_{\ell,2} = (0, \underbrace{1}_{2^{nd}\,column}, 0, \ldots, 0)$$

GFM II: Predictive Model Extension

From $\mathbf{x}_\ell^{CD}$ to arbitrary $\mathbf{x}_\ell$:

- $\mathbf{x}_\ell = (\mathbf{x}_{\ell,1}, \ldots, \mathbf{x}_{\ell,m})$ still consists of m subvectors for each mode
- While $\mathbf{x}_{\ell,j} \in \mathbf{x}_\ell^{CD}$ has exactly one active position, thus $p_j - 1$ zero values, e.g.,

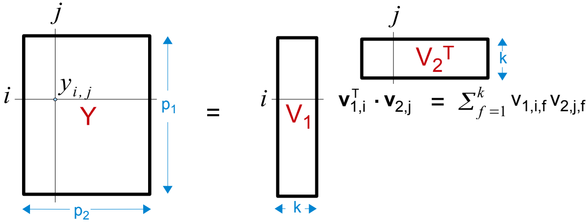
$$\mathbf{x}_{\ell,1} = (\underbrace{1}_{1^{st}\,row}, 0, \ldots, 0) \qquad \mathbf{x}_{\ell,2} = (0, \underbrace{1}_{2^{nd}\,column}, 0, \ldots, 0)$$

- Mode-vectors $\mathbf{x}_{\ell,j}$ may be defined arbitrarily, e.g., to take into account more complex conditional dependencies

Understanding induced dependencies:

Overall independence assumption: $y_\ell | \mathbf{x}_\ell \sim N(\mu_\ell = f(\mathbf{x}_\ell | V), 1)$

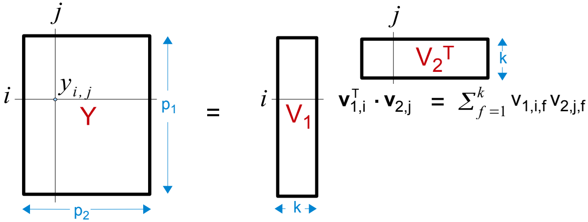
- **Simplest case SVD:** Correlation only on identical mode-indices, e.g. row or column index ↔ rows/columns are independent from other rows/columns



Understanding induced dependencies:

Overall independence assumption: $y_\ell | \mathbf{x}_\ell \sim N(\mu_\ell = f(\mathbf{x}_\ell | V), 1)$

- **Simplest case SVD:** Correlation only on identical mode-indices, e.g. row or column index $\leftrightarrow$ rows/columns are independent from other rows/columns



- e.g. $p_1 = 3$ rows, $p_2 = 8$ columns:

$$\mathbf{x}_{\ell,1} = (1,0,0), \ \mathbf{x}_{\ell,2} = (0,1,0,0,0,0,0,0)$$

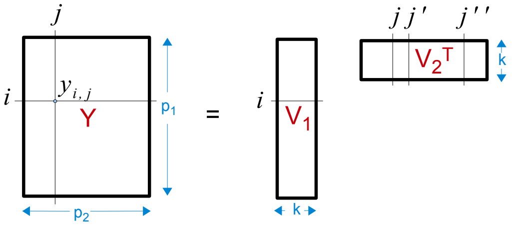
$$y_\ell = \sum_{f=1}^{k} \mu_{1,f}(\mathbf{x}_\ell)\mu_{2,f}(\mathbf{x}_\ell) + \epsilon_\ell = \sum_{f=1}^{k} (\mathbf{x}_{\ell,1}^T \mathbf{v}_{1,f})(\mathbf{x}_{\ell,2}^T \mathbf{v}_{2,f}) + \epsilon_\ell = \sum_{f=1}^{k} v_{1,1,f} v_{2,2,f} + \epsilon_\ell$$

Creating new dependencies:

Overall independence assumption: $y_\ell | \mathbf{x}_\ell \sim N(\mu_\ell = f(\mathbf{x}_\ell | V), 1)$

- **Dependencies between different rows/columns for SVD:**
  More than one active indicator per mode, with $p_1 = 3$ rows, $p_2 = 8$ columns:

$$\mathbf{x}_{\ell,1} = (1, 0, 0),\ \mathbf{x}_{\ell,2} = (0, 1, 1, 0, 1, 0, 0, 0)$$



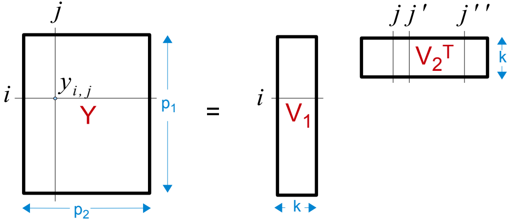
[4] Koren, Y.: Factorization meets the neighborhood: a multifaceted collaborative filtering model, KDD08.

Creating new dependencies:

Overall independence assumption: $y_\ell | \mathbf{x}_\ell \sim N(\mu_\ell = f(\mathbf{x}_\ell | V), 1)$

- **Dependencies between different rows/columns for SVD:**
  More than one active indicator per mode, with $p_1 = 3$ rows, $p_2 = 8$ columns:

$$\mathbf{x}_{\ell,1} = (1,0,0),\ \mathbf{x}_{\ell,2} = (0,1,1,0,1,0,0,0)$$



$$f(\mathbf{x}_\ell | V) = \sum_{f=1}^{k} (\mathbf{x}_{\ell,1}^T \mathbf{v}_{1,f})(\mathbf{x}_{\ell,2}^T \mathbf{v}_{2,f}) = \sum_{f=1}^{k} v_{1,1,f}(v_{2,2,f} + v_{2,3,f} + v_{2,5,f})$$

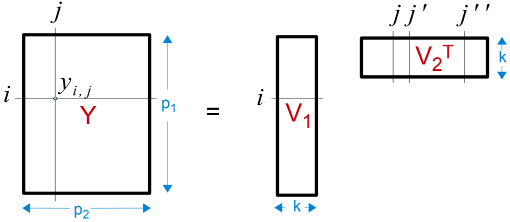
[4] Koren, Y.: Factorization meets the neighborhood: a multifaceted collaborative filtering model, KDD08.

Creating new dependencies:

Overall independence assumption: $y_\ell | \mathbf{x}_\ell \sim N(\mu_\ell = f(\mathbf{x}_\ell | V), 1)$

- **Dependencies between different rows/columns for SVD:** More than one active indicator per mode, with $p_1 = 3$ rows, $p_2 = 8$ columns:

$$\mathbf{x}_{\ell,1} = (1,0,0),\ \mathbf{x}_{\ell,2} = (0,1,1,0,1,0,0,0)$$



$$f(\mathbf{x}_\ell | V) = \sum_{f=1}^{k} (\mathbf{x}_{\ell,1}^T \mathbf{v}_{1,f})(\mathbf{x}_{\ell,2}^T \mathbf{v}_{2,f}) = \sum_{f=1}^{k} v_{1,1,f}(v_{2,2,f} + v_{2,3,f} + v_{2,5,f})$$

- Example from recommender systems: SVD++[4]

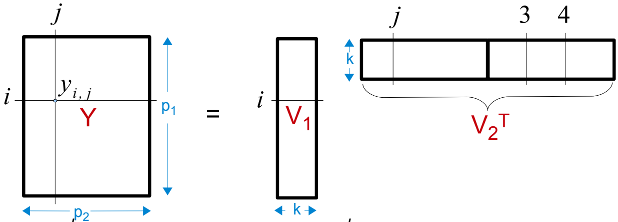
[4] Koren, Y.: Factorization meets the neighborhood: a multifaceted collaborative filtering model, KDD08.

Creating new dependencies:

Overall independence assumption: $y_\ell | \mathbf{x}_\ell \sim N(\mu_\ell = f(\mathbf{x}_\ell | V), 1)$

- **Dependencies beyond rows and/or columns for SVD:**
  $p_1 = 3$ rows and $p_2 = 8$ columns + additional rating information:

$$\mathbf{x}_{\ell,1} = \underbrace{(1,0,0)}_{users},\ \mathbf{x}_{\ell,2} = (\underbrace{0,1,0,0,0,0,0,0}_{items},\ \underbrace{0,0,3,0,4,0,0,0}_{\text{ratings for co-rated items}})$$



$$f(\mathbf{x}_\ell | V) = \sum_{f=1}^{k} (\mathbf{x}_{\ell,1}^T \mathbf{v}_{1,f})(\mathbf{x}_{\ell,2}^T \mathbf{v}_{2,f}) = \sum_{f=1}^{k} v_{1,1,f}(3 \cdot v_{2,3,f} + 4 \cdot v_{2,5,f})$$

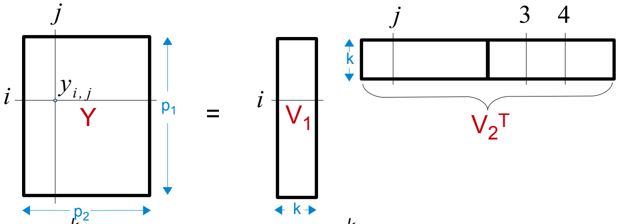
[5] Rendle, S., Freudenthaler, C., Schmidt-Thieme, L.: Factorizing personalized Markov chains for next-basket

Creating new dependencies:

Overall independence assumption: $y_\ell|\mathbf{x}_\ell \sim N(\mu_\ell = f(\mathbf{x}_\ell|V), 1)$

- **Dependencies beyond rows and/or columns for SVD:** $p_1 = 3$ rows and $p_2 = 8$ columns + additional rating information:

$$\mathbf{x}_{\ell,1} = \underbrace{(1,0,0)}_{users},\ \mathbf{x}_{\ell,2} = (\underbrace{0,1,0,0,0,0,0,0}_{items},\ \underbrace{0,0,3,0,4,0,0,0}_{\text{ratings for co-rated items}})$$



$$f(\mathbf{x}_\ell|V) = \sum_{f=1}^{k} (\mathbf{x}_{\ell,1}^T \mathbf{v}_{1,f})(\mathbf{x}_{\ell,2}^T \mathbf{v}_{2,f}) = \sum_{f=1}^{k} v_{1,1,f}(3 \cdot v_{2,3,f} + 4 \cdot v_{2,5,f})$$

- Example from recommender systems: Factorized Transition Tensors[5]

[5] Rendle, S., Freudenthaler, C., Schmidt-Thieme, L.: Factorizing personalized Markov chains for next-basket

GFM II: Predictive Model Extension II

$$f(\mathbf{x}_\ell | V) = \mu_\ell = \sum_{f=1}^{k} \prod_{j=1}^{m} \mathbf{x}_{\ell,j}^T \mathbf{v}_{j,f} = \sum_{f=1}^{k} \prod_{j=1}^{m} \sum_{i_j=1}^{p_j} x_{\ell,j,i_j} v_{j,i_j,f}$$

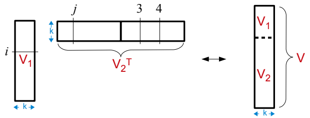
- Only one predictive model per mode m
- Increase predictive accuracy: combine several reasonable predictive models per mode

GFM II: Predictive Model Extension II

$$f(\mathbf{x}_\ell|V) = \mu_\ell = \sum_{f=1}^{k}\prod_{j=1}^{m}\mathbf{x}_{\ell,j}^T\mathbf{v}_{j,f} = \sum_{f=1}^{k}\prod_{j=1}^{m}\sum_{i_j=1}^{p_j} x_{\ell,j,i_j}\cdot 1\cdot v_{j,i_j,f}$$

- Only one predictive model per mode m
- Increase predictive accuracy: combine several reasonable predictive models per mode

- Introduce mode-defining selection vectors $\mathbf{d}_j \in \{0,1\}^p$ on $\mathbf{x}_\ell = (\mathbf{x}_1,\ldots,\mathbf{x}_m) \in \mathbb{R}^p$, $j = 1,\ldots,m$
- One set of selection vectors $\{\mathbf{d}_1,\ldots,\mathbf{d}_m\}$ defines one model



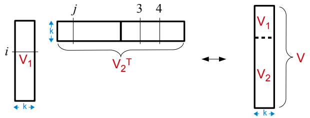
$$f(\mathbf{x}_\ell|V) = \mu_\ell = \sum_{f=1}^{k}\prod_{j=1}^{m}\sum_{i=1}^{p} x_{\ell,i}\, d_{j,i}\, v_{i,f}$$

GFM II: Predictive Model Extension II

$$f(\mathbf{x}_\ell|V) = \mu_\ell = \sum_{f=1}^{k} \prod_{j=1}^{m} \mathbf{x}_{\ell,j}^T \mathbf{v}_{j,f} = \sum_{f=1}^{k} \prod_{j=1}^{m} \sum_{i_j=1}^{p_j} x_{\ell,j,i_j} \cdot 1 \cdot v_{j,i_j,f}$$

- Only one predictive model per mode m
- Increase predictive accuracy: combine several reasonable predictive models per mode

- Introduce mode-defining selection vectors $\mathbf{d}_j \in \{0,1\}^p$ on $\mathbf{x}_\ell = (\mathbf{x}_1, \dots, \mathbf{x}_m) \in \mathbb{R}^p$, $j = 1, \dots, m$
- One set of selection vectors $\{\mathbf{d}_1, \dots, \mathbf{d}_m\}$ defines one model



$$f(\mathbf{x}_\ell|V) = \mu_\ell = \sum_{f=1}^{k} \prod_{j=1}^{m} \sum_{i=1}^{p} x_{\ell,i} d_{j,i} v_{i,f}$$

- Collect several selection sets $\{\mathbf{d}_1, \dots, \mathbf{d}_m\} \in \mathcal{D}$:

$$f^{GFM}(\mathbf{x}_\ell|V) = \mu_\ell = \sum_{f=1}^{k} \sum_{\{\mathbf{d}_1,\dots,\mathbf{d}_m\}\in\mathcal{D}} \prod_{j=1}^{m} \sum_{i=1}^{p} x_{\ell,i} d_{j,i} v_{i,f}$$

GFM II: Predictive Model Learning?

$$f^{GFM}(\mathbf{x}_\ell|V) = \sum_{i_1=1}^{p} \cdots \sum_{i_m=1}^{p} x_{\ell,i_1} \cdots x_{\ell,i_m} \underbrace{\sum_{\mathcal{D}} d_{1,i_1} \cdots d_{m,i_m}}_{\text{Interaction Weight I}} \underbrace{\sum_{f=1}^{k} v_{i_1,f} \cdots v_{i_m,f}}_{\text{Interaction Weight II}}$$

- Infer model selection $\mathcal{D} \rightarrow$ Bayesian model averaging ?
  - $+$ Also negative interaction effects of even order
  - \- Redundant parameterization
  - \- Expensive model prediction $O(|D|kmp)$

GFM II: Predictive Model Selection

- Specify $\mathcal{D}$, i.e. select a specific model

GFM II: Predictive Model Selection

- Specify $\mathcal{D}$, i.e. select a specific model

Extended Matrix Factorization

- $m = 2$ different modes, p different mode-models $\rightarrow |\mathcal{D}| = mp = 2p$

$$\mathcal{D} = \left\{ \begin{array}{l} \{(d_{1,1}, \ldots, d_{1,p}), \\ (d_{2,1}, \ldots, d_{2,p})\} \end{array} : d_{1,i_1} = 1, d_{2,i_2} = 1 \; \forall i_2 > i_1, 0 \; else \right\},$$

GFM II: Predictive Model Selection

- Specify $\mathcal{D}$, i.e. select a specific model

Extended Matrix Factorization

- $m = 2$ different modes, p different mode-models $\rightarrow |\mathcal{D}| = mp = 2p$

$$\mathcal{D} = \left\{ \begin{array}{l} \{(d_{1,1}, \ldots, d_{1,p}), \\ (d_{2,1}, \ldots, d_{2,p})\} \end{array} : d_{1,i_1} = 1, d_{2,i_2} = 1 \ \forall i_2 > i_1, 0 \ else \right\},$$

- **ex.1:** $\mathbf{d}_1 = (1, 0, 0, \ldots, 0), \ \mathbf{d}_2 = (0, 1, 1, 1, \ldots, 1)$
- **ex.2:** $\mathbf{d}_1 = (0, 1, 0, \ldots, 0), \ \mathbf{d}_2 = (0, 0, 1, 1, \ldots, 1)$

$$f(\mathbf{x}_\ell | V) = \mu_\ell = \sum_{i_1=1}^{p} \sum_{i_2>i_1}^{p} x_{\ell,i_1} x_{\ell,i_2} \sum_{f=1}^{k} v_{i_1,f} \, v_{i_2,f}$$

Selecting a prior distribution:

So far:

- $y_\ell | \mathbf{x}_\ell \sim N(\mu_\ell = f(\mathbf{x}_\ell | V), 1)$
- More general $\mathbf{x}_\ell$ unifies recent factorization model enhancements

Selecting a prior distribution:

So far:

- $y_\ell | \mathbf{x}_\ell \sim N(\mu_\ell = f(\mathbf{x}_\ell | V), 1)$
- More general $\mathbf{x}_\ell$ unifies recent factorization model enhancements

Next step:

- Selecting a prior distribution for model parameters V, using prior knowledge:
  - If all tensor elements $\in \mathbb{R}^k \rightarrow k$ linearly independent real valued latent dimensions

Selecting a prior distribution:

So far:

- $y_\ell | \mathbf{x}_\ell \sim N(\mu_\ell = f(\mathbf{x}_\ell | V), 1)$
- More general $\mathbf{x}_\ell$ unifies recent factorization model enhancements

Next step:

- Selecting a prior distribution for model parameters V, using prior knowledge:
  - If all tensor elements $\in \mathbb{R}^k \rightarrow k$ linearly independent real valued latent dimensions
  - Each tensor element is a sum over k linear functions of the same $\mathbf{x}_\ell$
  - Different variances σ_f^2 along k latent dimensions

Selecting a prior distribution:

So far:

- $y_\ell | \mathbf{x}_\ell \sim N(\mu_\ell = f(\mathbf{x}_\ell | V), 1)$
- More general $\mathbf{x}_\ell$ unifies recent factorization model enhancements

Next step:

- Selecting a prior distribution for model parameters V, using prior knowledge:
  - If all tensor elements $\in \mathbb{R}^k \rightarrow k$ linearly independent real valued latent dimensions
  - Each tensor element is a sum over k linear functions of the same $\mathbf{x}_\ell$
  - Different variances σ_f^2 along k latent dimensions
  - Centered at some dimension dependent mean μ_f

Selecting a prior distribution:

So far:

- $y_\ell | \mathbf{x}_\ell \sim N(\mu_\ell = f(\mathbf{x}_\ell | V), 1)$
- More general $\mathbf{x}_\ell$ unifies recent factorization model enhancements

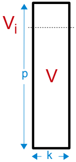
Next step:

- Selecting a prior distribution for model parameters V, using prior knowledge:
  - If all tensor elements $\in \mathbb{R}^k \rightarrow k$ linearly independent real valued latent dimensions
  - Each tensor element is a sum over k linear functions of the same $\mathbf{x}_\ell$
  - Different variances σ_f^2 along k latent dimensions
  - Centered at some dimension dependent mean μ_f
  - Center μ_f and variance σ_f^2 may differ for different modes

Selecting a prior distribution:

$\rightarrow$ **Conjugate Gaussian prior** distribution:
Each latent k-dim representation $\mathbf{v}_i \in \mathbb{R}^k$, $i = 1, \ldots, p$ is an independent draw

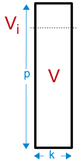
$$\mathbf{v}_i \sim N((\mu_1, \ldots, \mu_k), \text{diag}(\sigma_1^2, \ldots, \sigma_k^2))$$



Selecting a prior distribution:

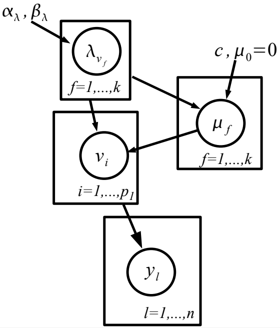
→ **Conjugate Gaussian prior** distribution:
Each latent k-dim representation $\mathbf{v}_i \in \mathbb{R}^k$, $i = 1, \ldots, p$ is an independent draw

$$\mathbf{v}_i \sim N((\mu_1, \ldots, \mu_k), \mathrm{diag}(\sigma_1^2, \ldots, \sigma_k^2))$$



→ **Conjugate Gaussian hyperprior**:
Each prior mean μ_f is considered an independent realization of

$$\mu_f \sim N(\mu_0, \frac{c}{\lambda_f})$$



→ **Conjugate Gamma hyperprior**:
Each precision $\lambda_f = \sigma_f^{-2}$ is considered an independent realization of

$$\lambda_f \sim G(\alpha_0, \beta_0)$$

Learning Generalized Factorization Models

- Stochastic Gradient Descent: scalable, simple

Learning Generalized Factorization Models

- Stochastic Gradient Descent: scalable, simple
- Alternating Least Squares

Learning Generalized Factorization Models

- Stochastic Gradient Descent: scalable, simple
- Alternating Least Squares
- Bayesian Analysis, i.e. standard Gibbs sampling:
  - Easy to derive for multi-linear models like

$$f(\mathbf{x}_\ell | V) = \sum_{i_1=1}^{p} \sum_{i_2>i_1}^{p} x_{\ell,i_1} x_{\ell,i_2} \sum_{f=1}^{k} v_{i_1,f}\, v_{i_2,f}$$

Learning Generalized Factorization Models

- Stochastic Gradient Descent: scalable, simple
- Alternating Least Squares
- Bayesian Analysis, i.e. standard Gibbs sampling:
  - Easy to derive for multi-linear models like

$$f(\mathbf{x}_\ell | V) = \sum_{i_1=1}^{p} \sum_{i_2>i_1}^{p} x_{\ell,i_1} x_{\ell,i_2} \sum_{f=1}^{k} v_{i_1,f}\, v_{i_2,f}$$

  - Learning on large datasets: block size $= 1 \rightarrow\ O(N_z k)$

Outline

Generalized Factorization Model

Relations to standard Models

Empirical Evaluation

Conclusion and Discussion

Polynomial Regression

$$f(\mathbf{x}_\ell|V) = \sum_{i_1=1}^{p} x_{\ell,i_1}\beta_{i_1} + \sum_{i_1=1}^{p}\sum_{i_2\geq i_1}^{p} x_{\ell,i_1}x_{\ell,i_2}\beta_{i_1,i_2} + \ldots + \sum_{i_1=1}^{p}\cdots\sum_{i_o\geq i_{o-1}}^{p} x_{\ell,i_1}\ldots x_{\ell,i_1}\beta_{i_1,\ldots,i_o}$$

- Order-o polynomial regression models

Polynomial Regression

$$f(\mathbf{x}_\ell|V) = \sum_{i_1=1}^{p} x_{\ell,i_1}\beta_{i_1} + \sum_{i_1=1}^{p}\sum_{i_2\geq i_1}^{p} x_{\ell,i_1}x_{\ell,i_2}\beta_{i_1,i_2} + \ldots + \sum_{i_1=1}^{p}\cdots\sum_{i_o\geq i_{o-1}}^{p} x_{\ell,i_1}\ldots x_{\ell,i_1}\beta_{i_1,\ldots,i_o}$$

- Order-*o* polynomial regression models

$$f^{GFM}(\mathbf{x}_\ell|V) = \sum_{f=1}^{k}\sum_{\{\mathbf{d}_1,\ldots,\mathbf{d}_m\}\in\mathcal{D}}\prod_{j=1}^{m}\sum_{i=1}^{p} x_{\ell,i}d_{j,i}v_{i,f}$$

- Generalized Factorization Model of *m*-mode tensor

Polynomial Regression

$$f(\mathbf{x}_\ell|V) = \sum_{i_1=1}^{p} x_{\ell,i_1}\beta_{i_1} + \sum_{i_1=1}^{p}\sum_{i_2\geq i_1}^{p} x_{\ell,i_1}x_{\ell,i_2}\beta_{i_1,i_2} + \ldots + \sum_{i_1=1}^{p}\cdots\sum_{i_o\geq i_{o-1}}^{p} x_{\ell,i_1}\ldots x_{\ell,i_1}\beta_{i_1,\ldots,i_o}$$

- Order-o polynomial regression models

$$f^{GFM}(\mathbf{x}_\ell|V) = \sum_{f=1}^{k}\sum_{\{\mathbf{d}_1,\ldots,\mathbf{d}_m\}\in\mathcal{D}}\prod_{j=1}^{m}\sum_{i=1}^{p} x_{\ell,i}d_{j,i}v_{i,f}$$

- Generalized Factorization Model of m-mode tensor
  - Define one of the p predictor variables as constant, e.g. $x_{\ell,1} = 1$ $\forall\ell = 1,\ldots,n$
  - Fix corresponding parameter vector $\mathbf{v}_1 \in V = \underbrace{(1,\ldots,1)}_{k}$

Polynomial Regression

$$f(\mathbf{x}_\ell|V) = \sum_{i_1=1}^{p} x_{\ell,i_1}\beta_{i_1} + \sum_{i_1=1}^{p}\sum_{i_2\geq i_1}^{p} x_{\ell,i_1}x_{\ell,i_2}\beta_{i_1,i_2} + \ldots + \sum_{i_1=1}^{p}\cdots\sum_{i_o\geq i_{o-1}}^{p} x_{\ell,i_1}\ldots x_{\ell,i_1}\beta_{i_1,\ldots,i_o}$$

- Order-o polynomial regression models

$$f^{GFM}(\mathbf{x}_\ell|V) = \sum_{f=1}^{k}\sum_{\{\mathbf{d}_1,\ldots,\mathbf{d}_m\}\in\mathcal{D}}\prod_{j=1}^{m}\sum_{i=1}^{p} x_{\ell,i}d_{j,i}v_{i,f}$$

- Generalized Factorization Model of m-mode tensor
  - Define one of the p predictor variables as constant, e.g. $x_{\ell,1} = 1$ $\forall \ell = 1,\ldots,n$
  - Fix corresponding parameter vector $\mathbf{v}_1 \in V = \underbrace{(1,\ldots,1)}_{k}$
  - Selecting $\mathcal{D}$ accordingly gives

$$f^{GFM}(\mathbf{x}_\ell|V) = \sum_{i_1=1}^{p} x_{\ell,i_1}\underbrace{\sum_{f=1}^{k} v_{i_1,f}}_{\beta_{i_1}} + \sum_{i_1=1}^{p}\sum_{i_2\geq i_1}^{p} x_{\ell,i_1}x_{\ell,i_2}\underbrace{\sum_{f=1}^{k} v_{i_1,f}v_{i_2,f}}_{\beta_{i_1,i_2}} + \ldots + \sum_{i_1=1}^{p}\cdots\sum_{i_m\geq i_{m-1}}^{p} x_{\ell,i_1}\ldots x_{\ell,i_m}\underbrace{\sum_{f=1}^{k} v_{i_1,f}\ldots v_{i_m,f}}_{\beta_{i_1,\ldots,i_m}}$$

Polynomial Regression vs. GFM

Generalized Factorization Model include Polynomial Regression

- For factorized parameters, e.g. $\beta_{i_1,i_2} = \sum_{f=1}^{k} v_{i_1,f} v_{i_2,f}$
- If number of modes m equals order o

Factorization Machines[6] vs. GFM

- Very similar to previous factorized polynomial regression model

$$f^{GFM}(\mathbf{x}_\ell|V) = \sum_{i_1=1}^{p} x_{\ell,i_1} \underbrace{\sum_{f=1}^{k} v_{i_1,f}}_{\beta_{i_1}} + \quad \textstyle\sum_{i_1=1}^{p} \sum_{i_2 \geq i_1}^{p} x_{\ell,i_1} x_{\ell,i_2} \underbrace{\displaystyle\sum_{f=1}^{k} v_{i_1,f} v_{i_2,f}}_{\beta_{i_1,i_2}} + \ldots$$

$$f^{FM}(\mathbf{x}_\ell|V) = \sum_{i_1=1}^{p} x_{\ell,i_1} \beta_{i_1} + \quad \textstyle\sum_{i_1=1}^{p} \sum_{i_2 \geq i_1}^{p} x_{\ell,i_1} x_{\ell,i_2} \underbrace{\displaystyle\sum_{f=1}^{k} v_{i_1,f} v_{i_2,f}}_{\beta_{i_1,i_2}} + \ldots$$

[6] Rendle, S.: Factorization Machines. ICDM10.

Factorization Machines[6] vs. GFM

- Very similar to previous factorized polynomial regression model

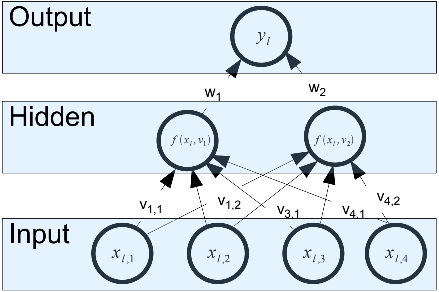
$$f^{GFM}(\mathbf{x}_\ell|V) = \sum_{i_1=1}^{p} x_{\ell,i_1} \underbrace{\sum_{f=1}^{k} v_{i_1,f}}_{\beta_{i_1}} + \quad \textstyle\sum_{i_1=1}^{p}\sum_{i_2\geq i_1}^{p} x_{\ell,i_1}x_{\ell,i_2} \underbrace{\displaystyle\sum_{f=1}^{k} v_{i_1,f}v_{i_2,f}}_{\beta_{i_1,i_2}} + \dots$$

$$f^{FM}(\mathbf{x}_\ell|V) = \sum_{i_1=1}^{p} x_{\ell,i_1}\beta_{i_1} + \quad \textstyle\sum_{i_1=1}^{p}\sum_{i_2\geq i_1}^{p} x_{\ell,i_1}x_{\ell,i_2} \underbrace{\displaystyle\sum_{f=1}^{k} v_{i_1,f}v_{i_2,f}}_{\beta_{i_1,i_2}} + \dots$$

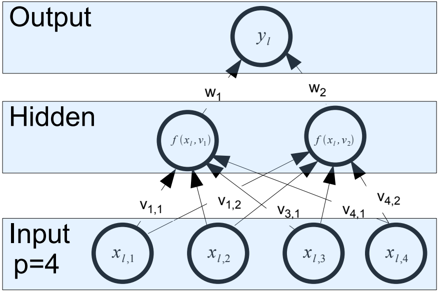
- Factorization Machines using simple Gibbs are successfully applied in several challenges

[6] Rendle, S.: Factorization Machines. ICDM10.

Neural Networks vs. GFM

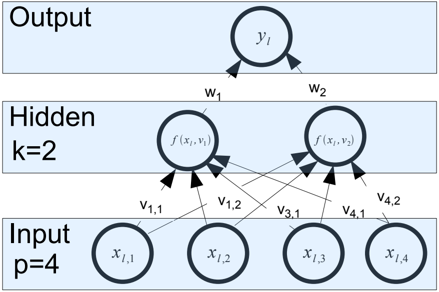


Neural Networks vs. GFM



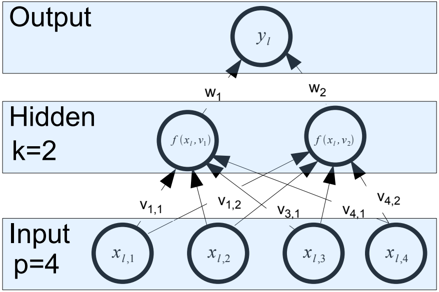
- **Input:** predictor vector $\mathbf{x}_\ell \in \mathbb{R}^p$, $\ell = 1, \ldots, n$

Neural Networks vs. GFM



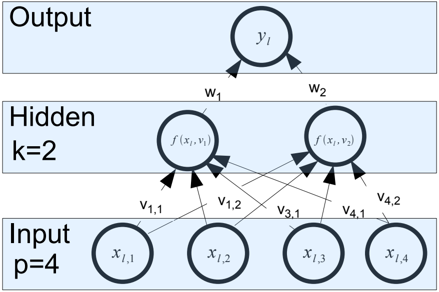
- **Input:** predictor vector $\mathbf{x}_\ell \in \mathbb{R}^p$, $\ell = 1, \ldots, n$
- **Hidden:** Hidden nodes correspond to latent dimensions $\rightarrow$ k nodes

Neural Networks vs. GFM



- **Input:** predictor vector $\mathbf{x}_\ell \in \mathbb{R}^p$, $\ell = 1, \ldots, n$
- **Hidden:** Hidden nodes correspond to latent dimensions $\rightarrow$ k nodes
- **Activation Function:** $f(\mathbf{x}_\ell, \mathbf{v}_f) := \sum_{\{\mathbf{d}_1,\ldots,\mathbf{d}_m\}\in\mathcal{D}} \prod_{j=1}^{m} \sum_{i=1}^{p} x_{\ell,i} d_{j,i} v_{i,f}$

Neural Networks vs. GFM



- **Input:** predictor vector $\mathbf{x}_\ell \in \mathbb{R}^p$, $\ell = 1, \ldots, n$
- **Hidden:** Hidden nodes correspond to latent dimensions $\rightarrow$ k nodes
- **Activation Function:** $f(\mathbf{x}_\ell, \mathbf{v}_f) := \sum_{\{\mathbf{d}_1,\ldots,\mathbf{d}_m\}\in\mathcal{D}} \prod_{j=1}^{m} \sum_{i=1}^{p} x_{\ell,i} d_{j,i} v_{i,f}$
- **Output:** Weights $w_f = 1$

Outline

Generalized Factorization Model

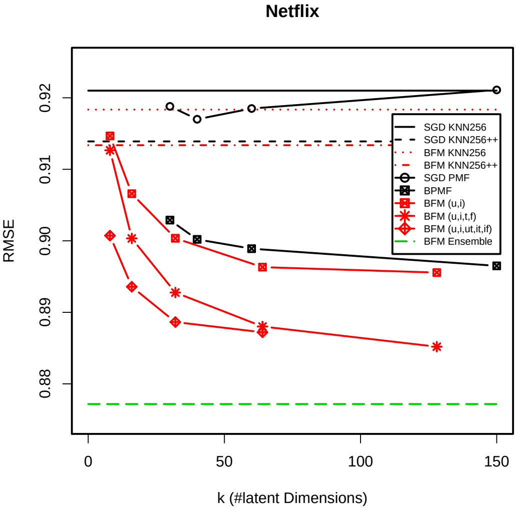
Relations to standard Models

Empirical Evaluation

Conclusion and Discussion

Empirical Evaluation on Recommender System Data

- **Task:** movie rating prediction
- **Data**[7]**:** 100M ratings of 480k users on 18k items



Bayesian Factorization Machines - Kaggle

- **Task:** student performance prediction on GMAT (Graduate Management Admission Test), SAT and ACT (college admission test)
- **118** contestants
- **Data**[8]**:**

#	Δ1w	Team Name	Capped Binomial Deviance	Entries	Last Submission UTC (Best Submission - Last)
1	-	Steffen *	0.24727	1	Sun, 04 Dec 2011 18:43:56
2	-	JP *	0.24773	9	Sun, 08 Jan 2012 23:56:28 (-3.1d)
3	-	YetiMan *	0.24795	22	Sat, 31 Dec 2011 16:56:22
4	-	PlanetThanet	0.24971	18	Tue, 10 Jan 2012 19:36:01
5	-	UCSD-Triton	0.25077	40	Mon, 09 Jan 2012 19:29:34 (-13.4d)

[8] http://www.kaggle.com/c/WhatDoYouKnow

Bayesian Factorization Machines - KDDCup'11

- **Task:** music rating prediction
- ≈ **1000** contestants
- **Data**[9]**:** 250M ratings of 1M users on 625k *items* (songs, tracks, album or artists)

Rank	Team Name	Best Score (RMSE)
1	National Taiwan University	21.0147
2	commendo	21.0815
3	InnerPeace	21.2634
4	Aron	21.5721
5	LeBuSiShu	21.8637
6	ICTIRDreamer	22.1813
7	Frantisek Hrdina	22.3367
8	slp008	22.3968
9	Just a guy in a garage	22.4665
10	Try&Go	22.5924
11	UvA AI	22.8131
12	remainder	22.8803
13	yahookddkiddingme	22.8803
14	packy	22.8823
15	[illegible]~[illegible]~[illegible]	22.8824
16	wahaha	22.8920
17	iloveZL	22.9125
18	libFM	22.9523
19	the_dl	22.9694
20	icad	23.0026

[9] http://kddcup.yahoo.com/

Outline

In a nutshell:

- GFM, thus matrix and tensor Factorization are types of regression models

In a nutshell:

- GFM, thus matrix and tensor Factorization are types of regression models
- Generalized Factorization Models relate to
  - Polynomial regression for factorized parameters

In a nutshell:

- GFM, thus matrix and tensor Factorization are types of regression models
- Generalized Factorization Models relate to
  - Polynomial regression for factorized parameters
  - Feed-forward Neural Networks for given activation function

In a nutshell:

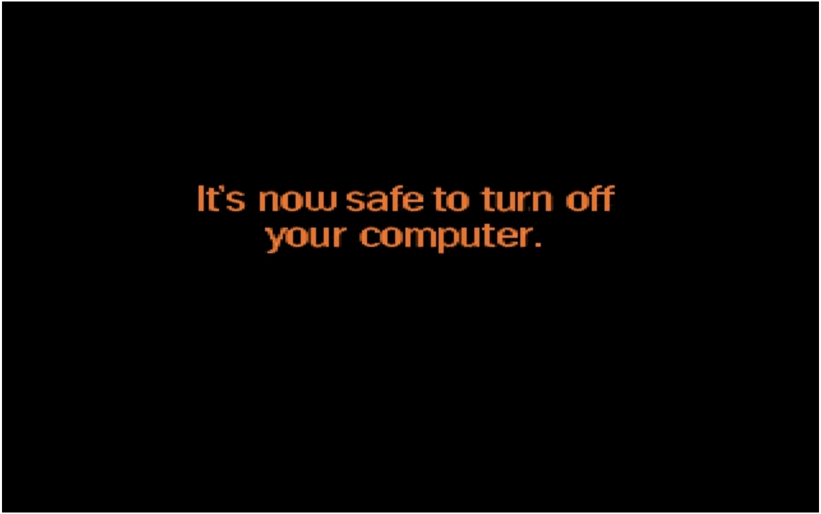
- GFM, thus matrix and tensor Factorization are types of regression models
- Generalized Factorization Models relate to
  - Polynomial regression for factorized parameters
  - Feed-forward Neural Networks for given activation function
- Factorization Machines with simple Gibbs show decent predictive performance

In a nutshell:

- GFM, thus matrix and tensor Factorization are types of regression models
- Generalized Factorization Models relate to
  - Polynomial regression for factorized parameters
  - Feed-forward Neural Networks for given activation function
- Factorization Machines with simple Gibbs show decent predictive performance

Open questions:

- Bayesian learning for models with non-linear functions of V?
- Bayesian model averaging for GFM?
- Efficient Bayesian inference for non-Gaussian likelihood?


It's now safe to turn off
your computer.