# Factorization Machines

Steffen Rendle

Current affiliation: Google Inc.
Work was done at University of Konstanz

MLConf, November 14, 2014

# Outline

- Factorization Models & Polynomial Regression
  - Factorization Models
  - Linear/ Polynomial Regression
  - Comparison
- Factorization Machines
- Applications
- Summary

# Matrix Factorization

Example for data:

| | | Movie | | | | |
|---|---|---|---|---|---|---|
| | | TI | NH | SW | ST | ... |
| User | A | 5 | 3 | 1 | ? | ... |
| | B | ? | ? | 4 | 5 | ... |
| | C | 1 | ? | 5 | ? | ... |
| | ... | ... | ... | ... | ... | ... |

Matrix Factorization:

$$\hat{Y} := W\,H^t, \quad W \in \mathbb{R}^{|U| \times k}, H \in \mathbb{R}^{|I| \times k}$$

$k$ is the rank of the reconstruction.

# Matrix Factorization

Example for data:

Movie

| | | TI | NH | SW | ST | ... |
|---|---|---|---|---|---|---|
| User | A | 5 | 3 | 1 | ? | ... |
| | B | ? | ? | 4 | 5 | ... |
| | C | 1 | ? | 5 | ? | ... |
| | ... | ... | ... | ... | ... | ... |

Matrix Factorization:

$$\hat{Y} := W\,H^t, \quad W \in \mathbb{R}^{|U|\times k}, H \in \mathbb{R}^{|I|\times k}$$

$$\hat{y}(u,i) = \hat{y}_{u,i} = \sum_{f=1}^{k} w_{u,f} h_{i,f} = \langle \mathbf{w}_u, \mathbf{h}_i \rangle$$

$k$ is the rank of the reconstruction.

# Matrix Factorization & Extensions

Example for data:

Movie

| | | TI | NH | SW | ST | ... |
|---|---|---|---|---|---|---|
| User | A | 5 | 3 | 1 | ? | ... |
| | B | ? | ? | 4 | 5 | ... |
| | C | 1 | ? | 5 | ? | ... |
| | ... | ... | ... | ... | ... | ... |

Examples for models:

$$\hat{y}^{\mathrm{MF}}(u,i) := \sum_{f=1}^{k} v_{u,f}\, v_{i,f} = \langle \mathbf{v}_u, \mathbf{v}_i \rangle$$

# Matrix Factorization & Extensions

Example for data:

Movie

| | | TI | NH | SW | ST | ... |
|---|---|---|---|---|---|---|
| User | A | 5 | 3 | 1 | ? | ... |
| | B | ? | ? | 4 | 5 | ... |
| | C | 1 | ? | 5 | ? | ... |
| | ... | ... | ... | ... | ... | ... |

Examples for models:

$$\hat{y}^{\mathrm{MF}}(u,i) := \sum_{f=1}^{k} v_{u,f}\, v_{i,f} = \langle \mathbf{v}_u, \mathbf{v}_i \rangle$$

$$\hat{y}^{\mathrm{SVD++}}(u,i) := \left\langle \mathbf{v}_u + \sum_{j \in N(u)} \mathbf{v}_j, \mathbf{v}_i \right\rangle$$

$$\hat{y}^{\mathrm{Fact\text{-}KNN}}(u,i) := \frac{1}{|R(u)|} \sum_{j \in R(u)} r_{u,j} \langle \mathbf{v}_i, \mathbf{v}_j \rangle$$

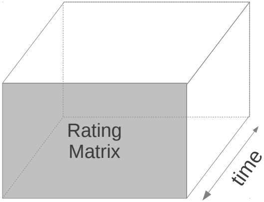
# Matrix Factorization & Extensions

Example for data:

Movie

User

|   | TI | NH | SW | ST | ... |
|---|---|---|---|---|---|
| A | 5 | 3 | 1 | ? | ... |
| B | ? | ? | 4 | 5 | ... |
| C | 1 | ? | 5 | ? | ... |
| ... | ... | ... | ... | ... | ... |



Examples for models:

$$\hat{y}^{\text{MF}}(u,i) := \sum_{f=1}^{k} v_{u,f}\, v_{i,f} = \langle \mathbf{v}_u, \mathbf{v}_i \rangle$$

$$\hat{y}^{\text{SVD++}}(u,i) := \left\langle \mathbf{v}_u + \sum_{j \in N(u)} \mathbf{v}_j, \mathbf{v}_i \right\rangle$$

$$\hat{y}^{\text{Fact-KNN}}(u,i) := \frac{1}{|R(u)|} \sum_{j \in R(u)} r_{u,j} \langle \mathbf{v}_i, \mathbf{v}_j \rangle$$

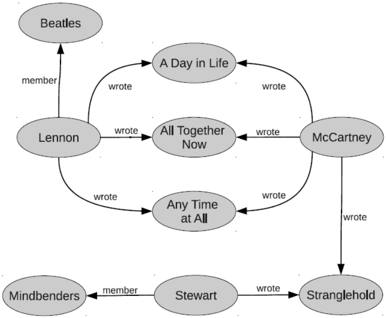
$$\hat{y}^{\text{timeSVD}}(u,i,t) := \langle \mathbf{v}_u + \mathbf{v}_{u,t}, \mathbf{v}_i \rangle$$

$$\hat{y}^{\text{timeTF}}(u,i,t) := \sum_{f=1}^{k} v_{u,f}\, v_{i,f}\, v_{t,f}$$

$$\ldots$$

# Tensor Factorization

Example for data:



Triples of **S**ubject, **P**redicate, **O**bject

Examples for models:

$$\hat{y}^{\text{PARAFAC}}(s,p,o) := \sum_{f=1}^{k} v_{s,f}\, v_{p,f}\, v_{o,f}$$

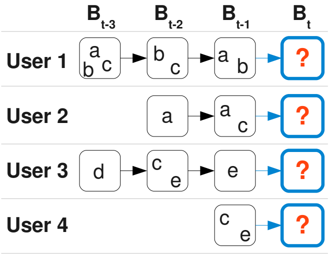
$$\hat{y}^{\text{PITF}}(s,p,o) := \langle \mathbf{v}_s, \mathbf{v}_p \rangle + \langle \mathbf{v}_s, \mathbf{v}_o \rangle + \langle \mathbf{v}_p, \mathbf{v}_o \rangle$$

$$\ldots$$

[illustration from Drumond et al. 2012]

# Sequential Factorization Models

Example for data:



Examples for models:

$$\hat{y}^{\text{FMC}}(u,i,t) := \sum_{l \in B_{t-1}} \langle \mathbf{v}_i, \mathbf{v}_l \rangle$$

$$\hat{y}^{\text{FPMC}}(u,i,t) := \langle \mathbf{v}_u, \mathbf{v}_i \rangle + \sum_{l \in B_{t-1}} \langle \mathbf{v}_i, \mathbf{v}_l \rangle$$

$$\ldots$$

# Factorization Models: Discussion

- Advantages
  - Can estimate interactions between two (or more) variables even if the cross is not observed.
  - E.g. user × movie, current product × next product, user × query × url, . . .

# Factorization Models: Discussion

- Advantages
  - Can estimate interactions between two (or more) variables even if the cross is not observed.
  - E.g. user × movie, current product × next product, user × query × url, . . .
- Downsides
  - Factorization models are usually build specifically for each problem.
  - Learning algorithms and implementations are tailored to individual models.

# Outline

- Factorization Models & Polynomial Regression
  - Factorization Models
  - Linear/ Polynomial Regression
  - Comparison
- Factorization Machines
- Applications
- Summary

# Data and Variable Representation

Many standard ML approaches work with real valued feature vectors as input. It allows to represent, e.g.:

- any number of variables
- categorical domains by using dummy indicator variables
- numerical domains
- set-categorical domains by using dummy indicator variables

Using this representation allows to apply a wide variety of standard models (e.g. linear regression, SVM, etc.).

# Linear Regression

- Let $\mathbf{x} \in \mathbb{R}^p$ be an input vector with $p$ predictor variables.
- Model equation:

$$\hat{y}(\mathbf{x}) := w_0 + \sum_{i=1}^{p} w_i \, x_i$$

- Model parameters:

$$w_0 \in \mathbb{R}, \quad \mathbf{w} \in \mathbb{R}^p$$

$\mathcal{O}(p)$ model parameters.

# Polynomial Regression

- Let $\mathbf{x} \in \mathbb{R}^p$ be an input vector with $p$ predictor variables.
- Model equation (degree 2):

$$\hat{y}(\mathbf{x}) := w_0 + \sum_{i=1}^{p} w_i x_i + \sum_{i=1}^{p} \sum_{j \geq i}^{p} w_{i,j} \, x_i \, x_j$$

- Model parameters:

$$w_0 \in \mathbb{R}, \quad \mathbf{w} \in \mathbb{R}^p, \quad \mathbf{W} \in \mathbb{R}^{p \times p}$$

$\mathcal{O}(p^2)$ model parameters.

# Outline

Factorization Models & Polynomial Regression

- Factorization Models
- Linear/ Polynomial Regression
- Comparison

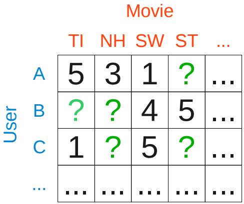
Factorization Machines

Applications

Summary

# Representation: Matrix/ Tensor vs. Feature Vectors

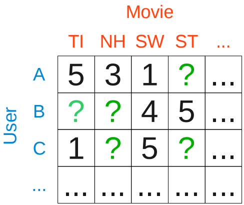
Matrix/ Tensor data can be represented by feature vectors:



| | | Movie | | | | |
|---|---|---|---|---|---|---|
| | | TI | NH | SW | ST | ... |
| User | A | 5 | 3 | 1 | ? | ... |
| | B | ? | ? | 4 | 5 | ... |
| | C | 1 | ? | 5 | ? | ... |
| | ... | ... | ... | ... | ... | ... |

# Representation: Matrix/ Tensor vs. Feature Vectors

Matrix/ Tensor data can be represented by feature vectors:



Movie

User

|  | TI | NH | SW | ST | ... |
|---|---|---|---|---|---|
| A | 5 | 3 | 1 | ? | ... |
| B | ? | ? | 4 | 5 | ... |
| C | 1 | ? | 5 | ? | ... |
| ... | ... | ... | ... | ... | ... |

$\Leftrightarrow$

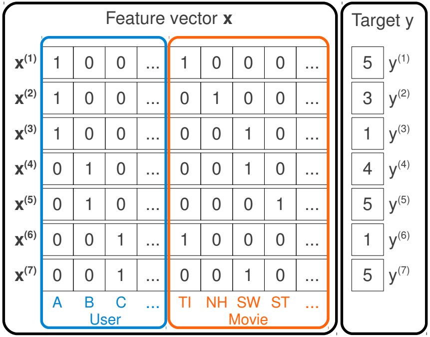
| # | User | Movie | Rating |
|---|---|---|---|
| 1 | **A**lice | **T**itanic | 5 |
| 2 | **A**lice | **N**otting **H**ill | 3 |
| 3 | **A**lice | **S**tar **W**ars | 1 |
| 4 | **B**ob | **S**tar **W**ars | 4 |
| 5 | **B**ob | **S**tar **T**rek | 5 |
| 6 | **C**harlie | **T**itanic | 1 |
| 7 | **C**harlie | **S**tar **W**ars | 5 |
| ... | ... | ... | ... |

# Representation: Matrix/ Tensor vs. Feature Vectors

Matrix/ Tensor data can be represented by feature vectors:

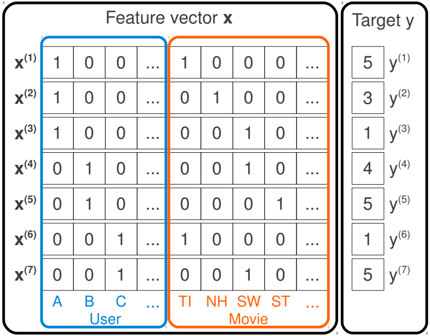
| # | User | Movie | Rating |
|---|---|---|---|
| 1 | **A**lice | **Ti**tanic | 5 |
| 2 | **A**lice | **N**otting **H**ill | 3 |
| 3 | **A**lice | **S**tar **W**ars | 1 |
| 4 | **B**ob | **S**tar **W**ars | 4 |
| 5 | **B**ob | **S**tar **T**rek | 5 |
| 6 | **C**harlie | **Ti**tanic | 1 |
| 7 | **C**harlie | **S**tar **W**ars | 5 |
| ... | ... | ... | ... |

$\Rightarrow$



| | Feature vector **x** | | | | | | | | | Target y | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $\mathbf{x}^{(1)}$ | 1 | 0 | 0 | ... | 1 | 0 | 0 | 0 | ... | 5 | $y^{(1)}$ |
| $\mathbf{x}^{(2)}$ | 1 | 0 | 0 | ... | 0 | 1 | 0 | 0 | ... | 3 | $y^{(2)}$ |
| $\mathbf{x}^{(3)}$ | 1 | 0 | 0 | ... | 0 | 0 | 1 | 0 | ... | 1 | $y^{(3)}$ |
| $\mathbf{x}^{(4)}$ | 0 | 1 | 0 | ... | 0 | 0 | 1 | 0 | ... | 4 | $y^{(4)}$ |
| $\mathbf{x}^{(5)}$ | 0 | 1 | 0 | ... | 0 | 0 | 0 | 1 | ... | 5 | $y^{(5)}$ |
| $\mathbf{x}^{(6)}$ | 0 | 0 | 1 | ... | 1 | 0 | 0 | 0 | ... | 1 | $y^{(6)}$ |
| $\mathbf{x}^{(7)}$ | 0 | 0 | 1 | ... | 0 | 0 | 1 | 0 | ... | 5 | $y^{(7)}$ |
| | A | B | C | ... | TI | NH | SW | ST | ... | | |
| | User | | | | Movie | | | | | | |

# Application to Sparse Feature Vectors

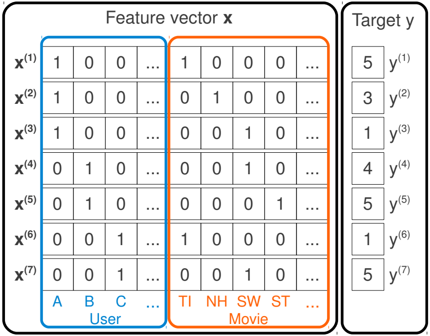


Feature vector **x**

| | User | | | | Movie | | | | | Target y | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | A | B | C | ... | TI | NH | SW | ST | ... | | |
| $\mathbf{x}^{(1)}$ | 1 | 0 | 0 | ... | 1 | 0 | 0 | 0 | ... | 5 | $y^{(1)}$ |
| $\mathbf{x}^{(2)}$ | 1 | 0 | 0 | ... | 0 | 1 | 0 | 0 | ... | 3 | $y^{(2)}$ |
| $\mathbf{x}^{(3)}$ | 1 | 0 | 0 | ... | 0 | 0 | 1 | 0 | ... | 1 | $y^{(3)}$ |
| $\mathbf{x}^{(4)}$ | 0 | 1 | 0 | ... | 0 | 0 | 1 | 0 | ... | 4 | $y^{(4)}$ |
| $\mathbf{x}^{(5)}$ | 0 | 1 | 0 | ... | 0 | 0 | 0 | 1 | ... | 5 | $y^{(5)}$ |
| $\mathbf{x}^{(6)}$ | 0 | 0 | 1 | ... | 1 | 0 | 0 | 0 | ... | 1 | $y^{(6)}$ |
| $\mathbf{x}^{(7)}$ | 0 | 0 | 1 | ... | 0 | 0 | 1 | 0 | ... | 5 | $y^{(7)}$ |

Applying regression models to this data leads to:

# Application to Sparse Feature Vectors



Feature vector **x**

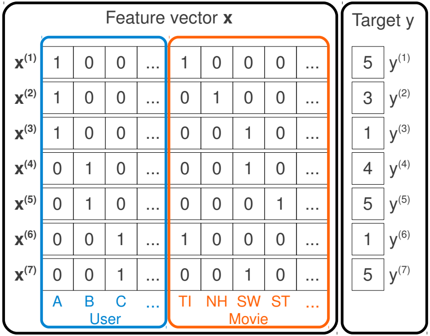
Target y

|  | A | B | C | ... | TI | NH | SW | ST | ... |  |  |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $\mathbf{x}^{(1)}$ | 1 | 0 | 0 | ... | 1 | 0 | 0 | 0 | ... | 5 | $y^{(1)}$ |
| $\mathbf{x}^{(2)}$ | 1 | 0 | 0 | ... | 0 | 1 | 0 | 0 | ... | 3 | $y^{(2)}$ |
| $\mathbf{x}^{(3)}$ | 1 | 0 | 0 | ... | 0 | 0 | 1 | 0 | ... | 1 | $y^{(3)}$ |
| $\mathbf{x}^{(4)}$ | 0 | 1 | 0 | ... | 0 | 0 | 1 | 0 | ... | 4 | $y^{(4)}$ |
| $\mathbf{x}^{(5)}$ | 0 | 1 | 0 | ... | 0 | 0 | 0 | 1 | ... | 5 | $y^{(5)}$ |
| $\mathbf{x}^{(6)}$ | 0 | 0 | 1 | ... | 1 | 0 | 0 | 0 | ... | 1 | $y^{(6)}$ |
| $\mathbf{x}^{(7)}$ | 0 | 0 | 1 | ... | 0 | 0 | 1 | 0 | ... | 5 | $y^{(7)}$ |
|  | User |  |  |  | Movie |  |  |  |  |  |  |

Applying regression models to this data leads to:

Linear regression: $\hat{y}(\mathbf{x}) = w_0 + w_u + w_i$

# Application to Sparse Feature Vectors



Feature vector **x**

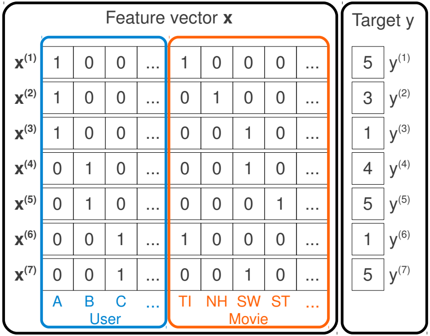
| | User | | | | Movie | | | | | Target y | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | A | B | C | ... | TI | NH | SW | ST | ... | | |
| $\mathbf{x}^{(1)}$ | 1 | 0 | 0 | ... | 1 | 0 | 0 | 0 | ... | 5 | $y^{(1)}$ |
| $\mathbf{x}^{(2)}$ | 1 | 0 | 0 | ... | 0 | 1 | 0 | 0 | ... | 3 | $y^{(2)}$ |
| $\mathbf{x}^{(3)}$ | 1 | 0 | 0 | ... | 0 | 0 | 1 | 0 | ... | 1 | $y^{(3)}$ |
| $\mathbf{x}^{(4)}$ | 0 | 1 | 0 | ... | 0 | 0 | 1 | 0 | ... | 4 | $y^{(4)}$ |
| $\mathbf{x}^{(5)}$ | 0 | 1 | 0 | ... | 0 | 0 | 0 | 1 | ... | 5 | $y^{(5)}$ |
| $\mathbf{x}^{(6)}$ | 0 | 0 | 1 | ... | 1 | 0 | 0 | 0 | ... | 1 | $y^{(6)}$ |
| $\mathbf{x}^{(7)}$ | 0 | 0 | 1 | ... | 0 | 0 | 1 | 0 | ... | 5 | $y^{(7)}$ |

Applying regression models to this data leads to:

Linear regression: $\hat{y}(\mathbf{x}) = w_0 + w_u + w_i$

Polynomial regression: $\hat{y}(\mathbf{x}) = w_0 + w_u + w_i + w_{u,i}$

# Application to Sparse Feature Vectors



Feature vector **x**

| | User | | | | Movie | | | | | Target y | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | A | B | C | ... | TI | NH | SW | ST | ... | | |
| $\mathbf{x}^{(1)}$ | 1 | 0 | 0 | ... | 1 | 0 | 0 | 0 | ... | 5 | $y^{(1)}$ |
| $\mathbf{x}^{(2)}$ | 1 | 0 | 0 | ... | 0 | 1 | 0 | 0 | ... | 3 | $y^{(2)}$ |
| $\mathbf{x}^{(3)}$ | 1 | 0 | 0 | ... | 0 | 0 | 1 | 0 | ... | 1 | $y^{(3)}$ |
| $\mathbf{x}^{(4)}$ | 0 | 1 | 0 | ... | 0 | 0 | 1 | 0 | ... | 4 | $y^{(4)}$ |
| $\mathbf{x}^{(5)}$ | 0 | 1 | 0 | ... | 0 | 0 | 0 | 1 | ... | 5 | $y^{(5)}$ |
| $\mathbf{x}^{(6)}$ | 0 | 0 | 1 | ... | 1 | 0 | 0 | 0 | ... | 1 | $y^{(6)}$ |
| $\mathbf{x}^{(7)}$ | 0 | 0 | 1 | ... | 0 | 0 | 1 | 0 | ... | 5 | $y^{(7)}$ |

Applying regression models to this data leads to:

Linear regression: $\hat{y}(\mathbf{x}) = w_0 + w_u + w_i$

Polynomial regression: $\hat{y}(\mathbf{x}) = w_0 + w_u + w_i + w_{u,i}$

Matrix factorization: $\hat{y}(u,i) = \langle \mathbf{w}_u, \mathbf{h}_i \rangle$

# Application to Sparse Feature Vectors

For the data of the example:

- Linear regression has no user-item interaction.

# Application to Sparse Feature Vectors

For the data of the example:

- Linear regression has no user-item interaction.
    - ⇒ Linear regression is not expressive enough.

# Application to Sparse Feature Vectors

For the data of the example:

- Linear regression has no user-item interaction.
    - ⇒ Linear regression is not expressive enough.

- Polynomial regression includes pairwise interactions but cannot estimate them from the data.

# Application to Sparse Feature Vectors

For the data of the example:

- Linear regression has no user-item interaction.
  - $\Rightarrow$ Linear regression is not expressive enough.
- Polynomial regression includes pairwise interactions but cannot estimate them from the data.
  - $n \ll p^2$: number of cases is much smaller than number of model parameters.

# Application to Sparse Feature Vectors

For the data of the example:

- Linear regression has no user-item interaction.
  - $\Rightarrow$ Linear regression is not expressive enough.
- Polynomial regression includes pairwise interactions but cannot estimate them from the data.
  - $n \ll p^2$: number of cases is much smaller than number of model parameters.
  - Max.-likelihood estimator for a pairwise effect is:

$$w_{i,j} = \begin{cases} y - w_0 - w_i - w_u, & \text{if } (i,j,y) \in S. \\ \text{not defined}, & \text{else} \end{cases}$$

# Application to Sparse Feature Vectors

For the data of the example:

- Linear regression has no user-item interaction.
  - $\Rightarrow$ Linear regression is not expressive enough.

- Polynomial regression includes pairwise interactions but cannot estimate them from the data.
  - $n \ll p^2$: number of cases is much smaller than number of model parameters.
  - Max.-likelihood estimator for a pairwise effect is:

$$w_{i,j} = \begin{cases} y - w_0 - w_i - w_u, & \text{if } (i,j,y) \in S. \\ \text{not defined}, & \text{else} \end{cases}$$

  - Polynomial regression cannot generalize to *any* unobserved pairwise effect.

# Outline

# Factorization Machine (FM)

- Let $\mathbf{x} \in \mathbb{R}^p$ be an input vector with $p$ predictor variables.
- Model equation (degree 2):

$$\hat{y}(\mathbf{x}) := w_0 + \sum_{i=1}^{p} w_i \, x_i + \sum_{i=1}^{p} \sum_{j>i}^{p} \langle \mathbf{v}_i, \mathbf{v}_j \rangle \, x_i \, x_j$$

- Model parameters:

$$w_0 \in \mathbb{R}, \quad \mathbf{w} \in \mathbb{R}^p, \quad \mathbf{V} \in \mathbb{R}^{p \times k}$$

[Rendle 2010, Rendle 2012]

# Factorization Machine (FM)

- Let $\mathbf{x} \in \mathbb{R}^p$ be an input vector with $p$ predictor variables.
- Model equation (degree 2):

$$\hat{y}(\mathbf{x}) := w_0 + \sum_{i=1}^{p} w_i \, x_i + \sum_{i=1}^{p} \sum_{j>i}^{p} \langle \mathbf{v}_i, \mathbf{v}_j \rangle \, x_i \, x_j$$

- Model parameters:

$$w_0 \in \mathbb{R}, \quad \mathbf{w} \in \mathbb{R}^p, \quad \mathbf{V} \in \mathbb{R}^{p \times k}$$

Compared to Polynomial regression:

- Model equation (degree 2):

$$\hat{y}(\mathbf{x}) := w_0 + \sum_{i=1}^{p} w_i x_i + \sum_{i=1}^{p} \sum_{j \geq i}^{p} w_{i,j} \, x_i \, x_j$$

- Model parameters:

$$w_0 \in \mathbb{R}, \quad \mathbf{w} \in \mathbb{R}^p, \quad \mathbf{W} \in \mathbb{R}^{p \times p}$$

[Rendle 2010, Rendle 2012]

# Factorization Machine (FM)

- Let $\mathbf{x} \in \mathbb{R}^p$ be an input vector with $p$ predictor variables.
- Model equation (degree 2):

$$\hat{y}(\mathbf{x}) := w_0 + \sum_{i=1}^{p} w_i \, x_i + \sum_{i=1}^{p} \sum_{j>i}^{p} \langle \mathbf{v}_i, \mathbf{v}_j \rangle \, x_i \, x_j$$

- Model parameters:

$$w_0 \in \mathbb{R}, \quad \mathbf{w} \in \mathbb{R}^p, \quad \mathbf{V} \in \mathbb{R}^{p \times k}$$

[Rendle 2010, Rendle 2012]

# Factorization Machine (FM)

- Let $\mathbf{x} \in \mathbb{R}^p$ be an input vector with $p$ predictor variables.
- Model equation (degree 3):

$$\hat{y}(\mathbf{x}) := w_0 + \sum_{i=1}^{p} w_i \, x_i + \sum_{i=1}^{p} \sum_{j>i}^{p} \langle \mathbf{v}_i, \mathbf{v}_j \rangle \, x_i \, x_j$$

$$+ \sum_{i=1}^{p} \sum_{j>i}^{p} \sum_{l>j}^{p} \sum_{f=1}^{k} v_{i,f}^{(3)} \, v_{j,f}^{(3)} \, v_{l,f}^{(3)} \, x_i \, x_j \, x_l$$

- Model parameters:

$$w_0 \in \mathbb{R}, \quad \mathbf{w} \in \mathbb{R}^p, \quad \mathbf{V} \in \mathbb{R}^{p \times k}, \quad \mathbf{V}^{(3)} \in \mathbb{R}^{p \times k}$$

[Rendle 2010, Rendle 2012]

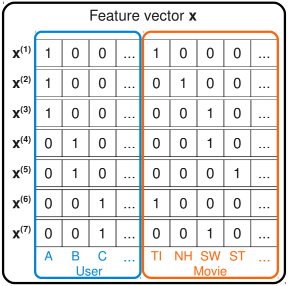
# Factorization Machines: Discussion

- FMs work with real valued input.
- FMs include variable interactions like polynomial regression.
- Model parameters for interactions are factorized.
- Number of model parameters is $\mathcal{O}(k\,p)$ (instead of $\mathcal{O}(p^2)$ for poly. regr.).

# Outline

# Matrix Factorization and Factorization Machines

Two categorical variables encoded with real valued predictor variables:



Feature vector **x**

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| $\mathbf{x}^{(1)}$ | 1 | 0 | 0 | ... | 1 | 0 | 0 | 0 | ... |
| $\mathbf{x}^{(2)}$ | 1 | 0 | 0 | ... | 0 | 1 | 0 | 0 | ... |
| $\mathbf{x}^{(3)}$ | 1 | 0 | 0 | ... | 0 | 0 | 1 | 0 | ... |
| $\mathbf{x}^{(4)}$ | 0 | 1 | 0 | ... | 0 | 0 | 1 | 0 | ... |
| $\mathbf{x}^{(5)}$ | 0 | 1 | 0 | ... | 0 | 0 | 0 | 1 | ... |
| $\mathbf{x}^{(6)}$ | 0 | 0 | 1 | ... | 1 | 0 | 0 | 0 | ... |
| $\mathbf{x}^{(7)}$ | 0 | 0 | 1 | ... | 0 | 0 | 1 | 0 | ... |
| | A | B | C | ... | TI | NH | SW | ST | ... |
| | User | | | | Movie | | | | |

With this data, the FM is identical to MF with biases[1]:

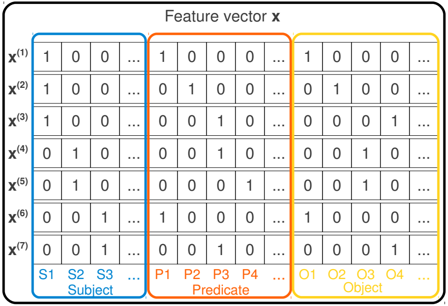
$$\hat{y}(\mathbf{x}) = w_0 + w_u + w_i + \underbrace{\langle \mathbf{v}_u, \mathbf{v}_i \rangle}_{\text{MF}}$$

---

[1] libFM, $k = 128$, MCMC inference, Netflix RMSE=0.8937

# RDF-Triple Prediction with Factorization Machines

Three categorical variables encoded with real valued predictor variables:



Feature vector **x**

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\mathbf{x}^{(1)}$ | 1 | 0 | 0 | ... | 1 | 0 | 0 | 0 | ... | 1 | 0 | 0 | 0 | ... |
| $\mathbf{x}^{(2)}$ | 1 | 0 | 0 | ... | 0 | 1 | 0 | 0 | ... | 0 | 1 | 0 | 0 | ... |
| $\mathbf{x}^{(3)}$ | 1 | 0 | 0 | ... | 0 | 0 | 1 | 0 | ... | 0 | 0 | 0 | 1 | ... |
| $\mathbf{x}^{(4)}$ | 0 | 1 | 0 | ... | 0 | 0 | 1 | 0 | ... | 0 | 0 | 1 | 0 | ... |
| $\mathbf{x}^{(5)}$ | 0 | 1 | 0 | ... | 0 | 0 | 0 | 1 | ... | 0 | 0 | 1 | 0 | ... |
| $\mathbf{x}^{(6)}$ | 0 | 0 | 1 | ... | 1 | 0 | 0 | 0 | ... | 1 | 0 | 0 | 0 | ... |
| $\mathbf{x}^{(7)}$ | 0 | 0 | 1 | ... | 0 | 0 | 1 | 0 | ... | 0 | 0 | 0 | 1 | ... |
| | S1 | S2 | S3 | ... | P1 | P2 | P3 | P4 | ... | O1 | O2 | O3 | O4 | ... |
| | Subject | | | | Predicate | | | | | Object | | | | |

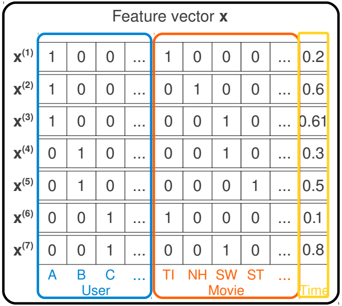
With this data, the FM is equivalent to the PITF model:

$$\hat{y}(\mathbf{x}) := w_0 + w_s + w_p + w_o + \langle \mathbf{v}_s, \mathbf{v}_p \rangle + \langle \mathbf{v}_s, \mathbf{v}_o \rangle + \langle \mathbf{v}_p, \mathbf{v}_o \rangle$$

[PITF: Rendle et al. 2010, WSDM Best Student Paper, ECML 2009 Best DC Award]

# Time with Factorization Machines

Two categorical variables and time as linear predictor:



Feature vector **x**

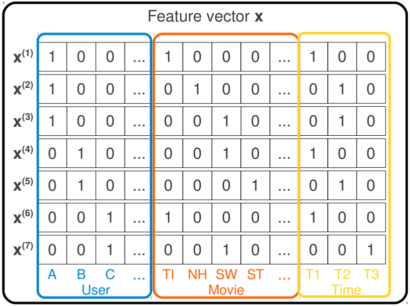
| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| $\mathbf{x}^{(1)}$ | 1 | 0 | 0 | ... | 1 | 0 | 0 | 0 | ... | 0.2 |
| $\mathbf{x}^{(2)}$ | 1 | 0 | 0 | ... | 0 | 1 | 0 | 0 | ... | 0.6 |
| $\mathbf{x}^{(3)}$ | 1 | 0 | 0 | ... | 0 | 0 | 1 | 0 | ... | 0.61 |
| $\mathbf{x}^{(4)}$ | 0 | 1 | 0 | ... | 0 | 0 | 1 | 0 | ... | 0.3 |
| $\mathbf{x}^{(5)}$ | 0 | 1 | 0 | ... | 0 | 0 | 0 | 1 | ... | 0.5 |
| $\mathbf{x}^{(6)}$ | 0 | 0 | 1 | ... | 1 | 0 | 0 | 0 | ... | 0.1 |
| $\mathbf{x}^{(7)}$ | 0 | 0 | 1 | ... | 0 | 0 | 1 | 0 | ... | 0.8 |
| | A | B | C | ... | TI | NH | SW | ST | ... | |
| | User | | | | Movie | | | | | Time |

The FM model would correspond to:

$$\hat{y}(\mathbf{x}) := w_0 + w_i + w_u + t\, w_{\text{time}} + \langle \mathbf{v}_u, \mathbf{v}_i \rangle + t\, \langle \mathbf{v}_u, \mathbf{v}_{\text{time}} \rangle + t\, \langle \mathbf{v}_i, \mathbf{v}_{\text{time}} \rangle$$

# Time with Factorization Machines

Two categorical variables and time discretized in bins ($b(t)$):



Feature vector **x**

| | User | | | | Movie | | | | | Time | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | A | B | C | ... | TI | NH | SW | ST | ... | T1 | T2 | T3 |
| $\mathbf{x}^{(1)}$ | 1 | 0 | 0 | ... | 1 | 0 | 0 | 0 | ... | 1 | 0 | 0 |
| $\mathbf{x}^{(2)}$ | 1 | 0 | 0 | ... | 0 | 1 | 0 | 0 | ... | 0 | 1 | 0 |
| $\mathbf{x}^{(3)}$ | 1 | 0 | 0 | ... | 0 | 0 | 1 | 0 | ... | 0 | 1 | 0 |
| $\mathbf{x}^{(4)}$ | 0 | 1 | 0 | ... | 0 | 0 | 1 | 0 | ... | 1 | 0 | 0 |
| $\mathbf{x}^{(5)}$ | 0 | 1 | 0 | ... | 0 | 0 | 0 | 1 | ... | 0 | 1 | 0 |
| $\mathbf{x}^{(6)}$ | 0 | 0 | 1 | ... | 1 | 0 | 0 | 0 | ... | 1 | 0 | 0 |
| $\mathbf{x}^{(7)}$ | 0 | 0 | 1 | ... | 0 | 0 | 1 | 0 | ... | 0 | 0 | 1 |

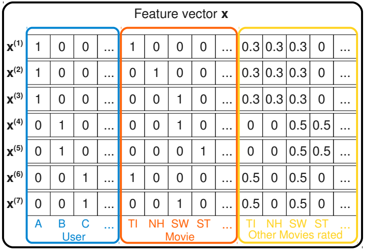
The FM model would correspond to:[2]

$$\hat{y}(\mathbf{x}) := w_0 + w_i + w_u + w_{b(t)} + \langle \mathbf{v}_u, \mathbf{v}_i \rangle + \langle \mathbf{v}_u, \mathbf{v}_{b(t)} \rangle + \langle \mathbf{v}_i, \mathbf{v}_{b(t)} \rangle$$

---

[2] libFM, $k = 128$, MCMC inference, Netflix RMSE=0.8873

# SVD++



Feature vector **x**

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\mathbf{x}^{(1)}$ | 1 | 0 | 0 | ... | 1 | 0 | 0 | 0 | ... | 0.3 | 0.3 | 0.3 | 0 | ... |
| $\mathbf{x}^{(2)}$ | 1 | 0 | 0 | ... | 0 | 1 | 0 | 0 | ... | 0.3 | 0.3 | 0.3 | 0 | ... |
| $\mathbf{x}^{(3)}$ | 1 | 0 | 0 | ... | 0 | 0 | 1 | 0 | ... | 0.3 | 0.3 | 0.3 | 0 | ... |
| $\mathbf{x}^{(4)}$ | 0 | 1 | 0 | ... | 0 | 0 | 1 | 0 | ... | 0 | 0 | 0.5 | 0.5 | ... |
| $\mathbf{x}^{(5)}$ | 0 | 1 | 0 | ... | 0 | 0 | 0 | 1 | ... | 0 | 0 | 0.5 | 0.5 | ... |
| $\mathbf{x}^{(6)}$ | 0 | 0 | 1 | ... | 1 | 0 | 0 | 0 | ... | 0.5 | 0 | 0.5 | 0 | ... |
| $\mathbf{x}^{(7)}$ | 0 | 0 | 1 | ... | 0 | 0 | 1 | 0 | ... | 0.5 | 0 | 0.5 | 0 | ... |
| | A | B | C | ... | TI | NH | SW | ST | ... | TI | NH | SW | ST | ... |
| | User | | | | Movie | | | | | Other Movies rated | | | | |

With this data, the FM[3] is identical to:

$$\hat{y}(\mathbf{x}) = \overbrace{w_0 + w_u + w_i + \langle \mathbf{v}_u, \mathbf{v}_i \rangle + \frac{1}{\sqrt{|N_u|}} \sum_{l \in N_u} \langle \mathbf{v}_i, \mathbf{v}_l \rangle}^{\text{SVD++}}$$

$$+ \frac{1}{\sqrt{|N_u|}} \sum_{l \in N_u} \left( w_l + \langle \mathbf{v}_u, \mathbf{v}_l \rangle + \frac{1}{\sqrt{|N_u|}} \sum_{l' \in N_u, l' > l} \langle \mathbf{v}_l, \mathbf{v}'_l \rangle \right)$$

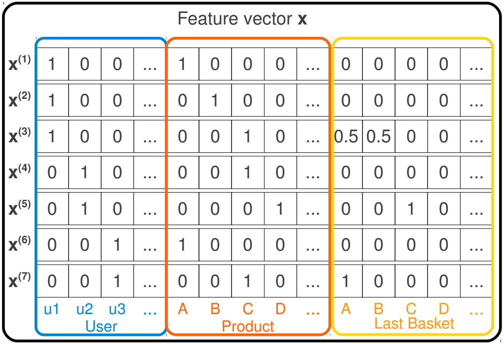
[Koren, 2008]

---

[3] libFM, $k = 128$, MCMC inference, Netflix RMSE=0.8865

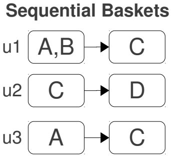
# Factorizing Personalized Markov Chains (FPMC)

Two categorical variables $(u,i)$, one set categorical $(B_{t-1})$:



Feature vector **x**

| | User | | | | Product | | | | | Last Basket | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | u1 | u2 | u3 | ... | A | B | C | D | ... | A | B | C | D | ... |
| $\mathbf{x}^{(1)}$ | 1 | 0 | 0 | ... | 1 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | ... |
| $\mathbf{x}^{(2)}$ | 1 | 0 | 0 | ... | 0 | 1 | 0 | 0 | ... | 0 | 0 | 0 | 0 | ... |
| $\mathbf{x}^{(3)}$ | 1 | 0 | 0 | ... | 0 | 0 | 1 | 0 | ... | 0.5 | 0.5 | 0 | 0 | ... |
| $\mathbf{x}^{(4)}$ | 0 | 1 | 0 | ... | 0 | 0 | 1 | 0 | ... | 0 | 0 | 0 | 0 | ... |
| $\mathbf{x}^{(5)}$ | 0 | 1 | 0 | ... | 0 | 0 | 0 | 1 | ... | 0 | 0 | 1 | 0 | ... |
| $\mathbf{x}^{(6)}$ | 0 | 0 | 1 | ... | 1 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | ... |
| $\mathbf{x}^{(7)}$ | 0 | 0 | 1 | ... | 0 | 0 | 1 | 0 | ... | 1 | 0 | 0 | 0 | ... |



FM is equivalent to

$$\hat{y}(\mathbf{x}) := w_0 + w_u + w_i + \frac{1}{|B_{t-1}|}\sum_{j\in B_{t-1}} w_j + \langle \mathbf{v}_u, \mathbf{v}_i \rangle + \frac{1}{|B_{t-1}|}\sum_{j\in B_{t-1}} \langle \mathbf{v}_i, \mathbf{v}_j \rangle + \ldots$$

[Rendle et al. 2010, WWW Best Paper]

# Outline

Factorization Models & Polynomial Regression

Factorization Machines

- Model
- Examples
- Properties
- Learning
- libFM Software

Applications

Summary

# Computation Complexity

Factorization Machine model equation:

$$\hat{y}(\mathbf{x}) := w_0 + \sum_{i=1}^{p} w_i \, x_i + \sum_{i=1}^{p} \sum_{j>i}^{p} \langle \mathbf{v}_i, \mathbf{v}_j \rangle \, x_i \, x_j$$

- Trivial computation: $\mathcal{O}(p^2 \, k)$

# Computation Complexity

Factorization Machine model equation:

$$\hat{y}(\mathbf{x}) := w_0 + \sum_{i=1}^{p} w_i\, x_i + \sum_{i=1}^{p} \sum_{j>i}^{p} \langle \mathbf{v}_i, \mathbf{v}_j \rangle\, x_i\, x_j$$

- Trivial computation: $\mathcal{O}(p^2\, k)$
- Efficient computation can be done in: $\mathcal{O}(p\, k)$

# Computation Complexity

Factorization Machine model equation:

$$\hat{y}(\mathbf{x}) := w_0 + \sum_{i=1}^{p} w_i \, x_i + \sum_{i=1}^{p} \sum_{j>i}^{p} \langle \mathbf{v}_i, \mathbf{v}_j \rangle \, x_i \, x_j$$

- Trivial computation: $\mathcal{O}(p^2 \, k)$
- Efficient computation can be done in: $\mathcal{O}(p \, k)$
- Making use of many zeros in $\mathbf{x}$ even in: $\mathcal{O}(N_z(\mathbf{x}) \, k)$, where $N_z(\mathbf{x})$ is the number of non-zero elements in vector $\mathbf{x}$.

# Efficient Computation

The model equation of an FM can be computed in $\mathcal{O}(p\,k)$.

# Efficient Computation

The model equation of an FM can be computed in $\mathcal{O}(p\,k)$.

Proof:

$$\begin{aligned}\hat{y}(\mathbf{x}) &:= w_0 + \sum_{i=1}^{p} w_i\,x_i + \sum_{i=1}^{p}\sum_{j>i}^{p}\langle \mathbf{v}_i, \mathbf{v}_j\rangle\,x_i\,x_j \\ &= w_0 + \sum_{i=1}^{p} w_i\,x_i + \frac{1}{2}\sum_{f=1}^{k}\left[\left(\sum_{i=1}^{p} x_i\,v_{i,f}\right)^2 - \sum_{i=1}^{p}\left(x_i\,v_{i,f}\right)^2\right]\end{aligned}$$

# Efficient Computation

The model equation of an FM can be computed in $\mathcal{O}(p\,k)$.

Proof:

$$\hat{y}(\mathbf{x}) := w_0 + \sum_{i=1}^{p} w_i\, x_i + \sum_{i=1}^{p} \sum_{j>i}^{p} \langle \mathbf{v}_i, \mathbf{v}_j \rangle\, x_i\, x_j$$

$$= w_0 + \sum_{i=1}^{p} w_i\, x_i + \frac{1}{2} \sum_{f=1}^{k} \left[ \left( \sum_{i=1}^{p} x_i\, v_{i,f} \right)^2 - \sum_{i=1}^{p} \left(x_i\, v_{i,f}\right)^2 \right]$$

- In the sums over $i$, only non-zero $x_i$ elements have to be summed up $\Rightarrow \mathcal{O}(N_z(\mathbf{x})\,k)$.
- (The complexity of polynomial regression is $\mathcal{O}(N_z(\mathbf{x})^2)$.)

# Multilinearity

FMs are multilinear:

$$\forall \theta \in \Theta = \{w_0, w_1, \ldots, w_p, v_{1,1}, \ldots, v_{p,k}\} : \quad \hat{y}(\mathbf{x}, \theta) = h_{(\theta)}(\mathbf{x})\, \theta + g_{(\theta)}(\mathbf{x})$$

where $g_{(\theta)}$ and $h_{(\theta)}$ do not depend on the value of $\theta$.

# Multilinearity

FMs are multilinear:

$$\forall \theta \in \Theta = \{w_0, w_1, \ldots, w_p, v_{1,1}, \ldots, v_{p,k}\} : \quad \hat{y}(\mathbf{x}, \theta) = h_{(\theta)}(\mathbf{x})\, \theta + g_{(\theta)}(\mathbf{x})$$

where $g_{(\theta)}$ and $h_{(\theta)}$ do not depend on the value of $\theta$.

E.g. for second order effects ($\theta = v_{l,f}$):

$$\hat{y}(\mathbf{x}, v_{l,f}) := \overbrace{w_0 + \sum_{i=1}^{p} w_i\, x_i + \sum_{i=1}^{p} \sum_{j=i+1}^{p} \sum_{\substack{f'=1 \\ (f' \neq f) \vee (l \notin \{i,j\})}}^{k} v_{i,f'}\, v_{j,f'}\, x_i\, x_j}^{g_{(v_{l,f})}(\mathbf{x})} + v_{l,f}\ \underbrace{x_l \sum_{i=1, i \neq l} v_{i,f}\, x_i}_{h_{(v_{l,f})}(\mathbf{x})}$$

# Outline

- Factorization Models & Polynomial Regression
- Factorization Machines
  - Model
  - Examples
  - Properties
  - Learning
  - libFM Software
- Applications
- Summary

# Learning

Using these properties, learning algorithms can be developed:

- L2-regularized regression and classification:
  - Stochastic gradient descent [Rendle, 2010]
  - Alternating least squares/ Coordinate Descent [Rendle et al., 2011, Rendle 2012]
  - Markov Chain Monte Carlo (for Bayesian FMs) [Freudenthaler et al. 2011, Rendle 2012]
- L2-regularized ranking:
  - Stochastic gradient descent [Rendle, 2010]

All the proposed learning algorithms have a runtime of $\mathcal{O}(k\, N_z(X)\, i)$, where $i$ is the number of iterations and $N_z(X)$ the number of non-zero elements in the design matrix $X$.

# Stochastic Gradient Descent (SGD)

- For each training case $(\mathbf{x}, y) \in S$, SGD updates the FM model parameter $\theta$ using:

$$\theta' = \theta - \alpha \, \big((\hat{y}(\mathbf{x}) - y) h_{(\theta)}(\mathbf{x}) + \lambda_{(\theta)} \theta\big)$$

- $\alpha$ is the learning rate / step size.
- $\lambda_{(\theta)}$ is the regularization value of the parameter $\theta$.
- SGD can easily be applied to other loss functions.

[Rendle, 2010]

# Coordinate Descent (CD)

- CD updates each FM model parameter $\theta$ using:

$$\theta' = \frac{\sum_{(\mathbf{x},y)\in S} \left(y - g_{(\theta)}(\mathbf{x})\right) h_{(\theta)}(\mathbf{x})}{\sum_{(\mathbf{x},y)\in S} h^2_{(\theta)}(\mathbf{x}) + \lambda_{(\theta)}}$$

- Using caches of intermediate results, the runtime for updating all model parameters is $\mathcal{O}(k\, N_z(X))$.
- CD can be extended to classification [Rendle, 2012].

[Rendle et al., 2011]

# Gibbs Sampling (MCMC)

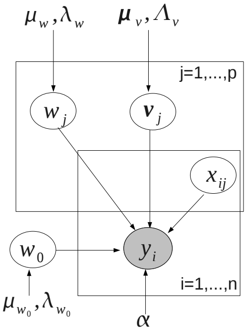
- Gibbs sampling with a block for each FM model parameter $\theta$:

$$\theta | S, \Theta \setminus \{\theta\} \sim \mathcal{N}\left( \frac{\alpha \sum_{(\mathbf{x},y)\in S} \left(y - g_{(\theta)}(\mathbf{x})\right) h_{(\theta)}(\mathbf{x})}{\alpha \sum_{(\mathbf{x},y)\in S} h^2_{(\theta)}(\mathbf{x}) + \lambda_{(\theta)}}, \frac{1}{\alpha \sum_{(\mathbf{x},y)\in S} h^2_{(\theta)}(\mathbf{x}) + \lambda_{(\theta)}} \right)$$

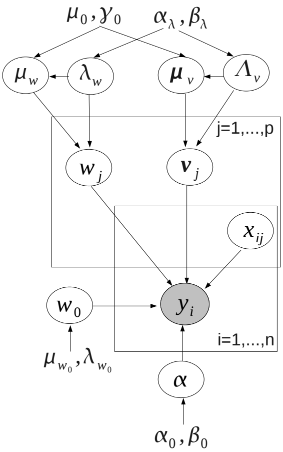
- Mean is the same as for CD $\Rightarrow$ computational complexity is also $\mathcal{O}(k\, N_z(X))$.
- MCMC can be extended to classification using link functions.

[Freudenthaler et al. 2011, Rendle 2012]

# Learning Regularization Values



Standard FM with priors.



Two level FM with hyperpriors.

[Freudenthaler et al., 2011]

# Outline

# libFM Software

libFM is an implementation of FMs

- Model: second-order FMs
- Learning/ inference: SGD, ALS, MCMC
- Classification and regression
- Uses the same data format as LIBSVM, LIBLINEAR [Lin et. al], SVMlight [Joachims].
- Supports variable grouping.
- Open source: GPLv3.

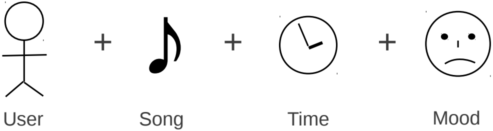
[`http://www.libfm.org/`]

# Outline

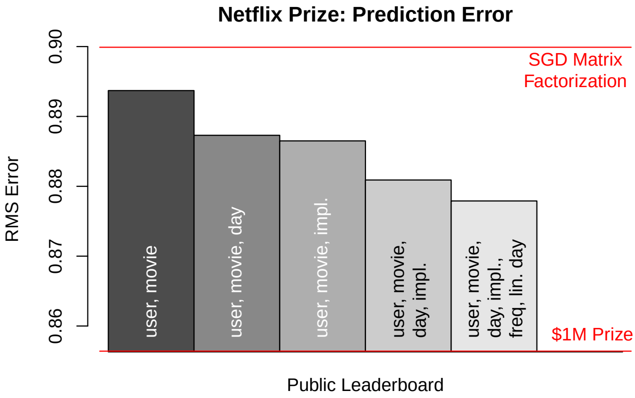
- Factorization Models & Polynomial Regression
- Factorization Machines
- Applications
  - Recommender Systems
  - Link Prediction in Social Networks
  - Clickthrough Prediction
  - Personalized Ranking
  - Student Performance Prediction
  - Kaggle Competitions
- Summary

# (Context-aware) Rating Prediction

- Main variables:
  - User ID (categorical)
  - Item ID (categorical)
- Additional variables:
  - time
  - mood
  - user profile
  - item meta data
  - ...
- Examples: Netflix prize, Movielens, KDDCup 2011



# Netflix Prize



- $k = 128$ factors, 512 MCMC samples (no burnin phase, initialization from random)
- MCMC inference (no hyperparameters (learning rate, regularization) to specify)

# Netflix Prize

| Method (Name) | Ref. | Learning Method | k | Quiz RMSE |
|---|---|---|---|---|
| *Models using user ID and item ID* | | | | |
| Probabilistic Matrix Factorization | [14, 13] | Batch GD | 40 | *0.9170 |
| Probabilistic Matrix Factorization | [14, 13] | Batch GD | 150 | 0.9211 |
| Matrix Factorization | [6] | Variational Bayes | 30 | *0.9141 |
| Matchbox | [15] | Variational Bayes | 50 | *0.9100 |
| ALS-MF | [7] | ALS | 100 | 0.9079 |
| ALS-MF | [7] | ALS | 1000 | *0.9018 |
| SVD/ MF | [3] | SGD | 100 | 0.9025 |
| SVD/ MF | [3] | SGD | 200 | *0.9009 |
| Bayesian Probablistic Matrix Factorization (BPMF) | [13] | MCMC | 150 | 0.8965 |
| Bayesian Probablistic Matrix Factorization (BPMF) | [13] | MCMC | 300 | *0.8954 |
| **FM, pred. var: user ID, movie ID** | - | MCMC | 128 | 0.8937 |
| *Models using implicit feedback* | | | | |
| Probabilistic Matrix Factorization with Constraints | [14] | Batch GD | 30 | *0.9016 |
| SVD++ | [3] | SGD | 100 | 0.8924 |
| SVD++ | [3] | SGD | 200 | *0.8911 |
| BSRM/F | [18] | MCMC | 100 | 0.8926 |
| BSRM/F | [18] | MCMC | 400 | *0.8874 |
| **FM, pred. var: user ID, movie ID, impl.** | - | MCMC | 128 | 0.8865 |

# Netflix Prize

| Method (Name) | Ref. | Learning Method | k | Quiz RMSE |
|---|---|---|---|---|
| *Models using time information* | | | | |
| Bayesian Probabilistic Tensor Factorization (BPTF) | [17] | MCMC | 30 | *0.9044 |
| **FM, pred. var: user ID, movie ID, day** | - | MCMC | 128 | 0.8873 |
| *Models using time and implicit feedback* | | | | |
| timeSVD++ | [5] | SGD | 100 | 0.8805 |
| timeSVD++ | [5] | SGD | 200 | *0.8799 |
| **FM, pred. var: user ID, movie ID, day, impl.** | - | MCMC | 128 | 0.8809 |
| **FM, pred. var: user ID, movie ID, day, impl.** | - | MCMC | 256 | 0.8794 |
| *Assorted models* | | | | |
| BRISMF/UM NB corrected | [16] | SGD | 1000 | *0.8904 |
| BMFSI plus side information | [8] | MCMC | 100 | *0.8875 |
| timeSVD++ plus frequencies | [4] | SGD | 200 | 0.8777 |
| timeSVD++ plus frequencies | [4] | SGD | 2000 | *0.8762 |
| **FM, pred. var: user ID, movie ID, day, impl., freq., lin. day** | - | MCMC | 128 | 0.8779 |
| **FM, pred. var: user ID, movie ID, day, impl., freq., lin. day** | - | MCMC | 256 | 0.8771 |

# Outline

Factorization Models & Polynomial Regression

Factorization Machines

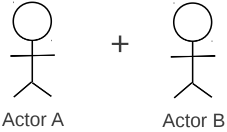
Applications

- Recommender Systems
- Link Prediction in Social Networks
- Clickthrough Prediction
- Personalized Ranking
- Student Performance Prediction
- Kaggle Competitions

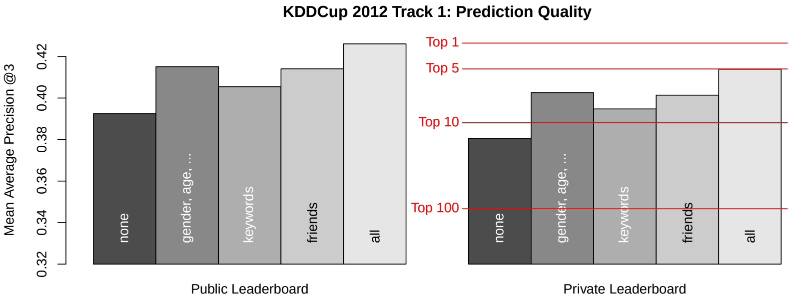
Summary

# Link Prediction in Social Networks

- Main variables:
  - Actor A ID
  - Actor B ID
- Additional variables:
  - profiles
  - actions
  - ...



# KDDCup 2012: Track 1



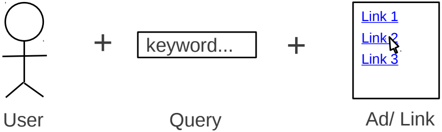
- $k = 22$ factors, 512 MCMC samples (no burnin phase, initialization from random)
- MCMC inference (no hyperparameters (learning rate, regularization) to specify)

[Awarded 2nd place (out of 658 teams)]

# Outline

# Clickthrough Prediction

- Main variables:
    - User ID
    - Query ID
    - Ad/ Link ID
- Additional variables:
    - query tokens
    - user profile
    - . . .



# KDDCup 2012: Track 2

| Model | Inference | wAUC (public) | wAUC (private) |
|---|---|---|---|
| ID-based model ($k = 0$) | SGD | 0.78050 | 0.78086 |
| Attribute-based model ($k = 8$) | MCMC | 0.77409 | 0.77555 |
| Mixed model ($k = 8$) | SGD | 0.79011 | 0.79321 |
| Final ensemble | n/a | 0.79857 | **0.80178** |

**Ensemble**

- Rank positions (not predicted clickthrough rates) are used.
- The MCMC attribute-based model and different variations of the SGD models are included.

[Awarded 3rd place (out of 171 teams)]

# Outline

Factorization Models & Polynomial Regression

Factorization Machines

Applications
- Recommender Systems
- Link Prediction in Social Networks
- Clickthrough Prediction
- Personalized Ranking
- Student Performance Prediction
- Kaggle Competitions

Summary

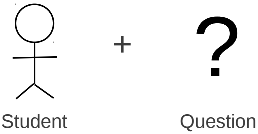
# ECML/PKDD Discovery Challenge 2013

- Problem: Recommend given names.
- Main variables:
  - User ID
  - Name ID
- Additional variables:
  - session info
  - string representation for each name
  - ...
- FM approach won 1st place (online track) and 2nd (offline track).

# Outline

# Student Performance Prediction

- Main variables:
    - Student ID
    - Question ID
- Additional variables:
    - question hierarchy
    - sequence of questions
    - skills required
    - ...
- Examples: KDDCup 2010, Grockit Challenge[4] (FM placed 1st/241)



---

[4] `http://www.kaggle.com/c/WhatDoYouKnow`

# Outline

Factorization Models & Polynomial Regression

Factorization Machines

Applications
- Recommender Systems
- Link Prediction in Social Networks
- Clickthrough Prediction
- Personalized Ranking
- Student Performance Prediction
- Kaggle Competitions

Summary

# Kaggle Competitions

FMs have been successfully applied to several Kaggle competitions:

- Criteon Display Advertising Challenge: 1st place (team '3 idiots').
- Blue Book for Bulldozers: 1st place (team 'Leustagos & Titericz').
- EMI Music Data Science Hackathon: 2nd place (team 'lns').

# Summary

- Factorization machines combine linear/polynomial regression with factorization models.
- Feature interactions are learned with a low rank representation.
- Estimation of unobserved interactions is possible.
- Factorization machines can be computed efficiently and have high prediction quality.

L. Drumond, S. Rendle, and L. Schmidt-Thieme.
Predicting rdf triples in incomplete knowledge bases with tensor factorization.
In *Proceedings of the 27th Annual ACM Symposium on Applied Computing*, SAC '12, pages 326–331, New York, NY, USA, 2012. ACM.

C. Freudenthaler, L. Schmidt-Thieme, and S. Rendle.
Bayesian factorization machines.
In *NIPS workshop on Sparse Representation and Low-rank Approximation*, 2011.

Y. Koren.
Factorization meets the neighborhood: a multifaceted collaborative filtering model.
In *KDD '08: Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 426–434, New York, NY, USA, 2008. ACM.

Y. Koren.
The bellkor solution to the netflix grand prize.
2009.

Y. Koren.
Collaborative filtering with temporal dynamics.
In *KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 447–456, New York, NY, USA, 2009. ACM.

Y. J. Lim and Y. W. Teh.
Variational Bayesian approach to movie rating prediction.
In *Proceedings of KDD Cup and Workshop*, 2007.

I. Pilászy, D. Zibriczky, and D. Tikk.
Fast als-based matrix factorization for explicit and implicit feedback datasets.
In *RecSys '10: Proceedings of the fourth ACM conference on Recommender systems*, pages 71–78, New York, NY, USA, 2010. ACM.

I. Porteous, A. Asuncion, and M. Welling.
Bayesian matrix factorization with side information and dirichlet process mixtures.
In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, AAAI 2010, pages 563–568, 2010.

S. Rendle.
Factorization machines.
In *Proceedings of the 2010 IEEE International Conference on Data Mining*, ICDM '10, pages 995–1000, Washington, DC, USA, 2010. IEEE Computer Society.

S. Rendle.
Factorization machines with libFM.
*ACM Trans. Intell. Syst. Technol.*, 3(3):57:1–57:22, May 2012.

S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme.
Factorizing personalized markov chains for next-basket recommendation.
In *WWW '10: Proceedings of the 19th international conference on World wide web*, pages 811–820, New York, NY, USA, 2010. ACM.

S. Rendle, Z. Gantner, C. Freudenthaler, and L. Schmidt-Thieme.
Fast context-aware recommendations with factorization machines.
In *Proceedings of the 34th ACM SIGIR Conference on Reasearch and Development in Information Retrieval*. ACM, 2011.

R. Salakhutdinov and A. Mnih.

Bayesian probabilistic matrix factorization using Markov chain Monte Carlo.
In *Proceedings of the 25th international conference on Machine learning*, ICML '08, pages 880–887, New York, NY, USA, 2008. ACM.

R. Salakhutdinov and A. Mnih.
Probabilistic matrix factorization.
In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 1257–1264, Cambridge, MA, 2008. MIT Press.

D. H. Stern, R. Herbrich, and T. Graepel.
Matchbox: large scale online bayesian recommendations.
In *Proceedings of the 18th international conference on World wide web*, WWW '09, pages 111–120, New York, NY, USA, 2009. ACM.

G. Takács, I. Pilászy, B. Németh, and D. Tikk.
Scalable collaborative filtering approaches for large recommender systems.
*J. Mach. Learn. Res.*, 10:623–656, June 2009.

L. Xiong, X. Chen, T.-K. Huang, J. Schneider, and J. G. Carbonell.
Temporal collaborative filtering with bayesian probabilistic tensor factorization.
In *Proceedings of the SIAM International Conference on Data Mining*, pages 211–222. SIAM, 2010.

S. Zhu, K. Yu, and Y. Gong.
Stochastic relational models for large-scale dyadic data using MCMC.
In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 1993–2000, 2009.