Latent Factor Model

– From a Probabilistic Perspective

Parameter Estimation

- **We face two inference problems:**
  - to estimate values for a set of distribution parameters ϑ that can best explain a set of observations X.
  - to calculate the probability of new observations $\tilde{x}$ given previous observations, i.e., to find $p(\tilde{x}|\mathcal{X})$.
- **The data set $\mathcal{X} \triangleq \{x_i\}_{i=1}^{|\mathcal{X}|}$ can be considered a sequence of independent and identically distributed (i.i.d.) realizations of a random variable (r.v.) X.**
- **For these data and parameters, a couple of probability functions are ubiquitous in Bayesian statistics. They are best introduced as parts of Bayes' rule, which is:**

$$p(\vartheta|\mathcal{X}) = \frac{p(\mathcal{X}|\vartheta) \cdot p(\vartheta)}{p(\mathcal{X})}, \quad \Longleftrightarrow \quad \text{posterior} = \frac{\text{likelihood} \cdot \text{prior}}{\text{evidence}}.$$

Maximum Likelihood Estimation (MLE)

- Maximum likelihood (ML) estimation tries to find parameters that maximize the likelihood:

$$L(\vartheta|\mathcal{X}) \triangleq p(\mathcal{X}|\vartheta) = \bigcap_{x\in\mathcal{X}} \{X = x|\vartheta\} = \prod_{x\in\mathcal{X}} p(x|\vartheta),$$

- The ML estimation problem then can be written as:

$$\hat{\vartheta}_{\mathrm{ML}} = \underset{\vartheta}{\operatorname{argmax}}\ \mathcal{L}(\vartheta|\mathcal{X}) = \underset{\vartheta}{\operatorname{argmax}} \sum_{x\in\mathcal{X}} \log p(x|\vartheta).$$

- The common way to obtain the parameter estimates is to solve the system:

$$\frac{\partial \mathcal{L}(\vartheta|\mathcal{X})}{\partial \vartheta_k} \stackrel{!}{=} 0 \quad \forall \vartheta_k \in \vartheta.$$

- The probability of a new observation given the data X can now be found using the approximation:

$$\begin{aligned} p(\tilde{x}|\mathcal{X}) &= \int_{\vartheta\in\Theta} p(\tilde{x}|\vartheta)\, p(\vartheta|\mathcal{X})\, \mathrm{d}\vartheta \\ &\approx \int_{\vartheta\in\Theta} p(\tilde{x}|\hat{\vartheta}_{\mathrm{ML}})\, p(\vartheta|\mathcal{X})\, \mathrm{d}\vartheta = p(\tilde{x}|\hat{\vartheta}_{\mathrm{ML}}), \end{aligned}$$

An Example of MLE

- **Consider a set C of N Bernoulli experiments with unknown parameter p, e.g., realized by tossing a deformed coin. The Bernoulli density function for the r.v. C for one experiment is:**

$$p(C=c|p) = p^c \ (1-p)^{1-c} \triangleq \mathrm{Bern}(c|p)$$

- **Building an ML estimator for the parameter p can be done by expressing the (log) likelihood as a function of the data:**

$$\begin{aligned}\mathcal{L} &= \log \prod_{i=1}^{N} p(C=c_i|p) = \sum_{i=1}^{N} \log p(C=c_i|p) \\ &= n^{(1)} \log p(C=1|p) + n^{(0)} \log p(C=0|p) \\ &= n^{(1)} \log p + n^{(0)} \log(1-p)\end{aligned}$$

- **Differentiating with respect to (w.r.t.) the parameter p yields:**

$$\frac{\partial \mathcal{L}}{\partial p} = \frac{n^{(1)}}{p} - \frac{n^{(0)}}{1-p} \overset{!}{=} 0 \quad \Leftrightarrow \quad \hat{p}_{\mathrm{ML}} = \frac{n^{(1)}}{n^{(1)} + n^{(0)}} = \frac{n^{(1)}}{N},$$

Maximum a Posteriori Estimation (MAP)

- **Maximum a posteriori (MAP) estimation is very similar to ML estimation but allows to include some a priori belief on the parameters by weighting them with a prior distribution p(ϑ).**
- **The name derives from the objective to maximize the posterior of the parameters given the data:**

$$\hat{\vartheta}_{\mathrm{MAP}} = \underset{\vartheta}{\operatorname{argmax}} \; p(\vartheta|\mathcal{X}).$$

Maximum a Posteriori Estimation (MAP)

- **By using Bayes' rule, this can be rewritten to:**

$$\begin{aligned}\hat{\vartheta}_{\mathrm{MAP}} &= \underset{\vartheta}{\operatorname{argmax}} \frac{p(\mathcal{X}|\vartheta)p(\vartheta)}{p(\mathcal{X})} \qquad \Big| \; p(\mathcal{X}) \neq f(\vartheta) \\ &= \underset{\vartheta}{\operatorname{argmax}} \; p(\mathcal{X}|\vartheta)p(\vartheta) = \underset{\vartheta}{\operatorname{argmax}} \; \{\mathcal{L}(\vartheta|\mathcal{X}) + \log p(\vartheta)\} \\ &= \underset{\vartheta}{\operatorname{argmax}} \Big\{ \sum_{x \in \mathcal{X}} \log p(x|\vartheta) \; + \; \log p(\vartheta) \Big\}.\end{aligned}$$

- **The probability of a new observation given the data X can now be found using the approximation:**

$$p(\tilde{x}|\mathcal{X}) \approx \int_{\vartheta \in \Theta} p(\tilde{x}|\hat{\vartheta}_{\mathrm{MAP}})\, p(\vartheta|\mathcal{X})\, \mathrm{d}\vartheta = p(\tilde{x}|\hat{\vartheta}_{\mathrm{MAP}}).$$

An Example of MAP

- **Consider the above experiment, but now there are values for p that we believe to be more likely, e.g., we believe that a coin usually is fair. This can be expressed as a prior distribution that has a high probability around 0.5. We choose the beta distribution:**

$$p(p|\alpha,\beta) = \frac{1}{\mathrm{B}(\alpha,\beta)} p^{\alpha-1}(1-p)^{\beta-1} \triangleq \mathrm{Beta}(p|\alpha,\beta),$$

$$\mathrm{B}(\alpha,\beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)} \qquad \Gamma(x) \text{ is the Gamma function}$$

- **The optimization problem now becomes:**

$$\begin{aligned}
\mathcal{L} &= \log \prod_{i=1}^{N} p(C{=}c_i|p) = \sum_{i=1}^{N} \log p(C{=}c_i|p) \\
&= n^{(1)} \log p(C{=}1|p) + n^{(0)} \log p(C{=}0|p) \\
&= n^{(1)} \log p + n^{(0)} \log(1-p)
\end{aligned}$$

$$\frac{\partial}{\partial p}\mathcal{L} + \log p(p) = \frac{n^{(1)}}{p} - \frac{n^{(0)}}{1-p} + \frac{\alpha-1}{p} - \frac{\beta-1}{1-p} \stackrel{!}{=} 0$$

$$\Leftrightarrow \quad \hat{p}_{\mathrm{MAP}} = \frac{n^{(1)}+\alpha-1}{n^{(1)}+n^{(0)}+\alpha+\beta-2}$$

Probabilistic Matrix Factorization

Ruslan Salakhutdinov and Andriy Mnih
Department of Computer Science, University of Toronto
6 King's College Rd, M5S 3G4, Canada
{rsalakhu,amnih}@cs.toronto.edu

Probabilistic Matrix Factorization

- **Some notations:**
  - We have *M* movies, *N* users.
  - $R_{i,j}$ represents the rating of user *i* for movie *j*.
  - Two matrices:
    - User $U \in R^{D\times N}$
    - Movie $V \in R^{D\times M}$

- **Probability of observed ratings:**

$$p(R|U,V,\sigma^2) = \prod_{i=1}^{N}\prod_{j=1}^{M}\left[\mathcal{N}(R_{ij}|U_i^T V_j, \sigma^2)\right]^{I_{ij}}$$

  - $\mathcal{N}(x|\mu,\sigma^2)$ corresponds to Gaussian distribution.

Probabilistic Matrix Factorization

- **Add prior distributions to user and item matrices**

$$p(U|\sigma_U^2) = \prod_{i=1}^{N} \mathcal{N}(U_i|0, \sigma_U^2 \mathbf{I}), \qquad p(V|\sigma_V^2) = \prod_{j=1}^{M} \mathcal{N}(V_j|0, \sigma_V^2 \mathbf{I}).$$

- **Posterior distribution of user and item matrices**

$$\boldsymbol{P}\left(\boldsymbol{U}, \boldsymbol{V} \middle| \boldsymbol{R}, \boldsymbol{\sigma}^2, \boldsymbol{\sigma}_V^2, \boldsymbol{\sigma}_U^2\right) = \frac{\boldsymbol{P}(\boldsymbol{R}|\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{\sigma}^2) \times \boldsymbol{P}(\boldsymbol{U}|\boldsymbol{\sigma}_U^2) \times \boldsymbol{P}(\boldsymbol{V}|\boldsymbol{\sigma}_V^2)}{\boldsymbol{P}(\boldsymbol{R}|\boldsymbol{\sigma}^2, \boldsymbol{\sigma}_V^2, \boldsymbol{\sigma}_U^2)}$$

Probabilistic Matrix Factorization

- **MAP estimation for matrix factorization**

$$\ln p(U,V|R,\sigma^2,\sigma_V^2,\sigma_U^2) = -\frac{1}{2\sigma^2}\sum_{i=1}^{N}\sum_{j=1}^{M} I_{ij}(R_{ij}-U_i^T V_j)^2 - \frac{1}{2\sigma_U^2}\sum_{i=1}^{N} U_i^T U_i - \frac{1}{2\sigma_V^2}\sum_{j=1}^{M} V_j^T V_j$$
$$-\frac{1}{2}\left(\left(\sum_{i=1}^{N}\sum_{j=1}^{M} I_{ij}\right)\ln\sigma^2 + ND\ln\sigma_U^2 + MD\ln\sigma_V^2\right) + C, \qquad (3)$$

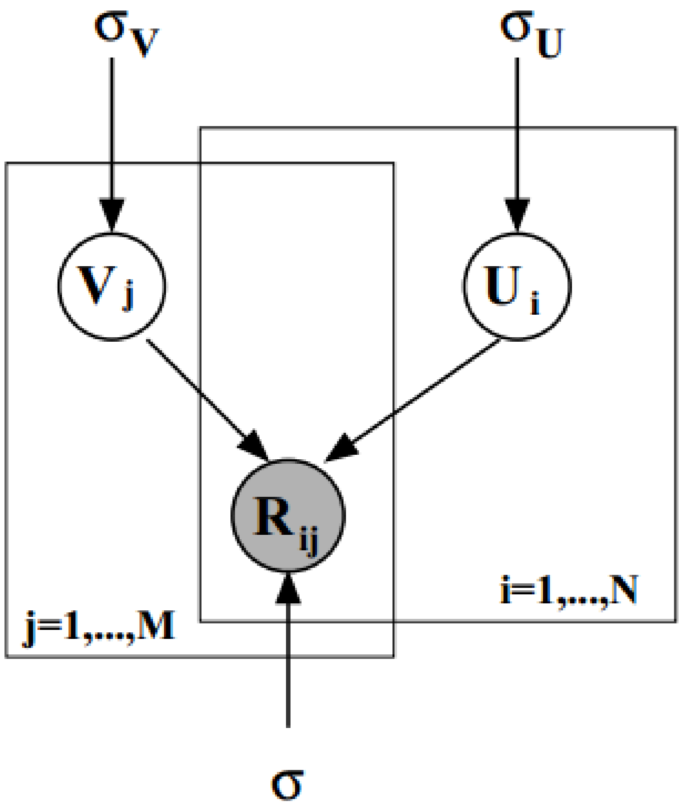
$$\Downarrow$$

$$E = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{M} I_{ij}\left(R_{ij}-U_i^T V_j\right)^2 + \frac{\lambda_U}{2}\sum_{i=1}^{N} \| U_i \|_{Fro}^2 + \frac{\lambda_V}{2}\sum_{j=1}^{M} \| V_j \|_{Fro}^2$$

- **Now, the regularized version of matrix factorization is derived.**

Graphical Representation for PMF

- **Notations:**
- **Solid circle: observed variable**
- **Empty circle: hidden variable**
- **Plate: containing multiple variables**



Factorization Machines

Factorization Machines

Steffen Rendle
Department of Reasoning for Intelligence
The Institute of Scientific and Industrial Research
Osaka University, Japan
rendle@ar.sanken.osaka-u.ac.jp

Now at Google!
Less pubs in recent years.

Handling User/Item Features

- **What if instead of user/item IDs we are given user and item features?**
- **Assume user u and item v have feature vectors**
  - User (f_u): user ID, gender, income, etc.
  - Item (g_v): item ID, category, etc.
- **Some one-hot vectors**

Country=USA	Country=China	Day=26/11/15	Day=1/7/14	Day=19/2/15	Ad_type=Movie	Ad_type=Game
1	0	1	0	0	1	0
0	1	0	1	0	0	1
0	1	0	0	1	0	1

- **How to utilize these feature vectors to build model?**

A Simple Way

- We can consider a regression problem where data instances are,

$$
\begin{array}{cc}
\textbf{Target value} & \textbf{Feature} \\
\vdots & \vdots \\
r_{uv} & \begin{bmatrix} \mathbf{f}_u^T & \mathbf{g}_v^T \end{bmatrix} \\
\vdots & \vdots
\end{array}
$$

- The target is to solve,

$$\min_{\mathbf{w}} \sum_{u,v \in R} \left(R_{u,v} - \mathbf{w}^T \begin{bmatrix} \mathbf{f}_u \\ \mathbf{g}_v \end{bmatrix} \right)^2$$

Feature Combinations

- **The above regression based method does not take the interaction between features into account.**
- **Recap latent factor models and its variants**

$$f(u,i) = \alpha + \beta_u + \beta_i + \boxed{\gamma_u \cdot \gamma_i}$$

$$f(u,i) = \alpha + \beta_u + \beta_i + \boxed{(\gamma_u + \sum_{a \in A(u)} \rho_a) \cdot \gamma_i}$$

  - The interaction between (other) features is missed.

Feature Combinations

- **A solution of interacting features is to generate new features,**

$$(f_u)_t(g_v)_s, t = 1, \ldots, U, s = 1, \ldots V$$

- **In this way,**

$$\min_{w_{t,s}, \forall t,s} \sum_{u,v \in R} \left(r_{u,v} - \sum_{t'=1}^{U} \sum_{s'=1}^{V} w_{t',s'} (f_u)_t (g_v)_s\right)^2$$

Feature Combinations

- **However, this solution fails for sparse features, just like one-hot vectors.**
  - This is because many dimensions of the generated new features equal to 0

$$U = m, J = n,$$
$$\mathbf{f}_i = [\underbrace{0, \ldots, 0}_{i-1}, 1, 0, \ldots, 0]^T$$

  - In this situation, the parameter matrix ***W*** could not be learned well.

$$\min_{w_{t,s}, \forall t,s} \sum_{u,v \in R} (r_{u,v} - \sum_{t'=1}^{U} \sum_{s'=1}^{V} \boxed{w_{t',s'}} (f_u)_t (g_v)_s)^2$$

Feature Combinations

- **The reason why we cannot learn *W* well is because the optimization problem encounters**

$$\# \text{ parameters} = mn \gg \# \text{ instances} = |R|$$

- **Remedy: we can let**

$$W \approx P^T Q,$$

  - where ***P*** and ***Q*** are low-rank matrices. This becomes **matrix factorization**.

- **In other words, now each feature could be associated with a vector, leading to factorization machines.**

Factorization Machine

- **Illustrative example**

Feature vector **x** | Target y

	User				Movie					Other Movies rated					Time	Last Movie rated					Target y	
	A	B	C	...	TI	NH	SW	ST	...	TI	NH	SW	ST	...	Time	TI	NH	SW	ST	...		
$\mathbf{x}^{(1)}$	1	0	0	...	1	0	0	0	...	0.3	0.3	0.3	0	...	13	0	0	0	0	...	5	$y^{(1)}$
$\mathbf{x}^{(2)}$	1	0	0	...	0	1	0	0	...	0.3	0.3	0.3	0	...	14	1	0	0	0	...	3	$y^{(2)}$
$\mathbf{x}^{(3)}$	1	0	0	...	0	0	1	0	...	0.3	0.3	0.3	0	...	16	0	1	0	0	...	1	$y^{(2)}$
$\mathbf{x}^{(4)}$	0	1	0	...	0	0	1	0	...	0	0	0.5	0.5	...	5	0	0	0	0	...	4	$y^{(3)}$
$\mathbf{x}^{(5)}$	0	1	0	...	0	0	0	1	...	0	0	0.5	0.5	...	8	0	0	1	0	...	5	$y^{(4)}$
$\mathbf{x}^{(6)}$	0	0	1	...	1	0	0	0	...	0.5	0	0.5	0	...	9	0	0	0	0	...	1	$y^{(5)}$
$\mathbf{x}^{(7)}$	0	0	1	...	0	0	1	0	...	0.5	0	0.5	0	...	12	1	0	0	0	...	5	$y^{(6)}$

Factorization Machine

- **Model equation**

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

  - <., .> is the dot product of two vectors
  - $\mathbf{v}_i$ has the dimensional size of *k*.
  - *n* is the dimension of features.

- **Computational complexity**
  - $O(n^2k)$

Factorization Machine

- **Fast computation**

$$
\begin{aligned}
&\sum_{i=1}^{n} \sum_{j=i+1}^{n} \langle \mathbf{v}_i, \mathbf{v}_j \rangle \, x_i \, x_j \\
=&\frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \langle \mathbf{v}_i, \mathbf{v}_j \rangle \, x_i \, x_j - \frac{1}{2} \sum_{i=1}^{n} \langle \mathbf{v}_i, \mathbf{v}_i \rangle \, x_i \, x_i \\
=&\frac{1}{2} \left(\sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{f=1}^{k} v_{i,f} \, v_{j,f} \, x_i \, x_j - \sum_{i=1}^{n} \sum_{f=1}^{k} v_{i,f} \, v_{i,f} \, x_i \, x_i \right) \\
=&\frac{1}{2} \sum_{f=1}^{k} \left(\left(\sum_{i=1}^{n} v_{i,f} \, x_i \right) \left(\sum_{j=1}^{n} v_{j,f} \, x_j \right) - \sum_{i=1}^{n} v_{i,f}^2 \, x_i^2 \right) \\
=&\frac{1}{2} \sum_{f=1}^{k} \left(\left(\sum_{i=1}^{n} v_{i,f} \, x_i \right)^2 - \sum_{i=1}^{n} v_{i,f}^2 \, x_i^2 \right)
\end{aligned}
$$

Complexity: O(kn)

Factorization Machine

- **Learning factorization machines**
  - Stochastic gradient descent

$$\frac{\partial}{\partial\theta}\hat{y}(\mathbf{x}) = \begin{cases} 1, & \text{if } \theta \text{ is } w_0 \\ x_i, & \text{if } \theta \text{ is } w_i \\ x_i \sum_{j=1}^{n} v_{j,f} x_j - v_{i,f} x_i^2, & \text{if } \theta \text{ is } v_{i,f} \end{cases}$$

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