

ERATO感謝祭

Convex Factorization Machine for Toxicogenomics Prediction

Makoto Yamada, Wenzhao Lian, Amit Goyal, Jianhui Chen,
Kishan Wimalawarne, Suleiman Khan,
Samuel Kaski, Hiroshi Mamitsuka, Yi Chang



- Mail
- News
- Finance
- Sports
- Fantasy Football
- Politics
- Celebrity
- View
- TV
- Movies
- Style
- Beauty
- Shopping
- Tech
- More on Yahoo

Make YAHOO! your homepage

2016 Big Data Trends

Top 8 Trends in Big Data Whitepaper

Close Ad

Ad recommendation

Fisher, Ford had affair during Star Wars

Cerie Fisher says she was 19 when she had for her 33-year-old married co-star while filming the original movie. "It was so intense!"

Oxford Dictionaries' word of the year

New Balance shoes caught in controversy

Report: 3 NBA teams boycotting Trump hotels

'Catch' star Jake Harris has cracked skull

Image recommendation

Trending Now

1. Beastie Boys
2. Kim Kardashian
3. Tiffany Trump
4. Blake Lively
5. Dolly Parton
6. Luxury SUV Deals
7. Amber Heard
8. Rheumatoid Arthrit...
9. Kareena Kapoor
10. 2016 Cars

YAHOO! MAIL

One inbox. Every email.

Yahoo, Gmail, Hotmail, and AOL together at last.

Get the mail app

Science

Stephen Hawking Puts An Expiry Date On Humanity

The famed physicist said Tuesday that humanity has less than a thousand years left on Earth before a mass extinction occurs.

Politics

Megyn Kelly Believes President Donald Trump Could Be 'Dangerous'

Report: 3 NBA teams boycotting Trump hotels

This rare technique is making Japanese investors a fortune in 2016...

News recommendation

Yahoo!社ホームページ
https://www.yahoo.com/
から転載

Recommendation

- User-item matrix.

$A =$

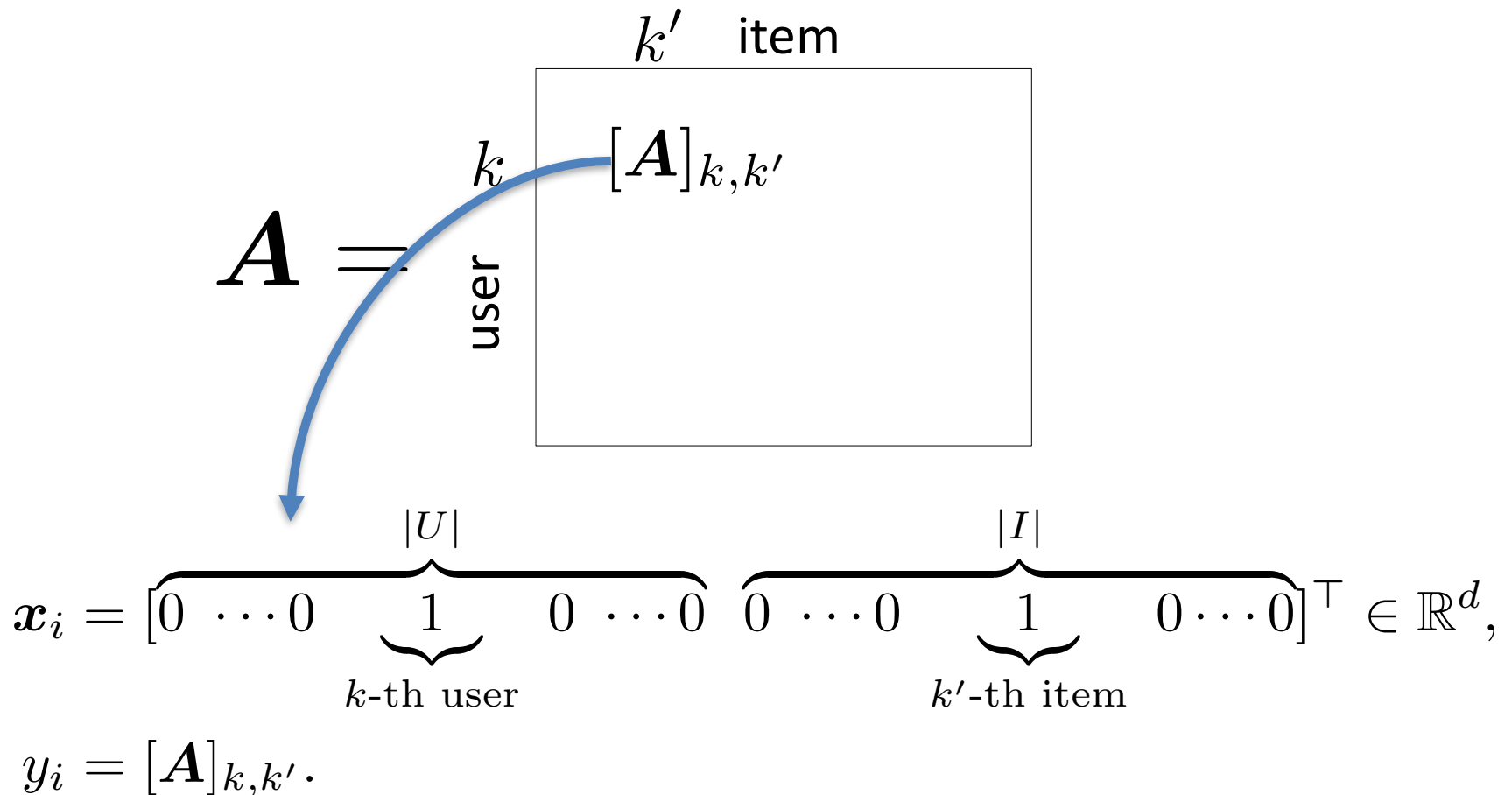
	Item1	Item2	Item3	Item4	Item5	Item6
User1	5		1		2	2
User2		3	2	1	3	5
User3	3	4		3		
User4			4	5	4	
User5	1		1			3

- **Goal:** impute un-observed elements from the observed data.
- Matrix completion has been widely used.

Factorization Machines

Rendle, ICDM 2010

- FM: A generalization of matrix completion.



Factorization Machines

Rendle, ICDM 2010

- Model

$$f(\mathbf{x}; \mathbf{w}, \mathbf{G}) = w_0 + \overset{\text{User and item bias}}{\mathbf{w}_0^\top \mathbf{x}} + \sum_{\ell=1}^d \sum_{\ell'=\ell+1}^d \overset{\text{User item interaction}}{\mathbf{g}_\ell^\top \mathbf{g}_{\ell'} x_\ell x_{\ell'}}$$

- FM is equivalent to matrix completion algorithm.

$$\hat{\mathbf{A}}_{k,k'} = w_0 + [\mathbf{w}_0]_k + [\mathbf{w}_0]_{|U|+k'} + \mathbf{g}_k^\top \mathbf{g}_{|U|+k'}$$

- We can **incorporate side information** by simply concatenating the information to input vector.
- **However, it is non-convex algorithm** 😞.

Convex Factorization Machine

- Contributions of our paper are
 - We propose a convex variant of FMs, which is convex and can get a **globally optimal solution**.
 - The proposed algorithm is **simple yet efficient**.
 - We show a **convex Tucker-based tensor completion** can be solved through CFM.
 - We applied the CFM to a **toxicogenomics prediction** task. Then, it outperformed a state-of-the-art method.
- **We want to convert all completion problem to FM format and solve it with FM**

Convex Factorization Machine

Blondel et al., ECML 2015

Yamada et al. KDD 2017, arXiv 2015

- Model

$$f(\mathbf{x}; \mathbf{w}, \mathbf{W}) = w_0 + \mathbf{w}_0^\top \mathbf{x} + \sum_{\ell=1}^d \sum_{\ell'=\ell+1}^d \mathbf{W}_{\ell,\ell'} x_\ell x_{\ell'}$$

- CFM is equivalent to a convex matrix completion algorithm.

$$\hat{\mathbf{A}}_{k,k'} = w_0 + [\mathbf{w}_0]_k + [\mathbf{w}_0]_{|U|+k'} + [\mathbf{W}]_{k,|U|+k'}$$

- We formulate the optimization problem as a **semi-definite programming** and solve it with **Hazan's algorithm** (Frank-Wolfe).

Convex Factorization Machine: Updating formula

- Optimization problem: w, \mathbf{W}

$$\min_{w, \mathbf{W}} \|\mathbf{y} - \mathbf{f}(\mathbf{X}; w, \mathbf{W})\|_2^2 + \lambda_1 \|w\|_2^2$$

$$\text{s.t. } \mathbf{W} \succeq 0 \text{ and } \|\mathbf{W}\|_{\text{tr}} = \eta$$

- Eliminating w vector and only updates w.r.t. \mathbf{W} .

$$w^* = (\mathbf{Z}\mathbf{Z}^\top + \lambda_1 \mathbf{I}_{d+1})^{-1} \mathbf{Z}(\mathbf{y} - \mathbf{f}_Q(\mathbf{X}; \mathbf{W}))$$

$$\mathbf{f}_Q(\mathbf{X}; \mathbf{W}) = [f_Q(\mathbf{x}_1; \mathbf{W}), \dots, f_Q(\mathbf{x}_n; \mathbf{W})]^\top \in \mathbb{R}^n,$$

$$f_Q(\mathbf{x}; \mathbf{W}) = \frac{1}{2} \text{tr}(\mathbf{W}(\mathbf{x}\mathbf{x}^\top - \text{diag}(\mathbf{x} \circ \mathbf{x})))$$

$$\mathbf{z} = [1 \ \mathbf{x}^\top]^\top \in \mathbb{R}^{d+1}, \mathbf{Z} = [\mathbf{z}_1, \dots, \mathbf{z}_n] \in \mathbb{R}^{(d+1) \times n}$$

Convex Factorization Machine: Updating formula

- Optimization problem w.r.t. \mathbf{W}

$$\min_{\mathbf{W}} J(\mathbf{W}) \quad \text{s.t.} \quad \mathbf{W} \succeq 0 \quad \text{and} \quad \|\mathbf{W}\|_{\text{tr}} = \eta$$

$$J(\mathbf{W}) = (\mathbf{y} - \mathbf{f}_Q(\mathbf{X}; \mathbf{W}))^\top \mathbf{C} (\mathbf{y} - \mathbf{f}_Q(\mathbf{X}; \mathbf{W})),$$

$$\mathbf{C} = \mathbf{R}^\top \mathbf{R} + \lambda_1 \mathbf{H}^\top \mathbf{H},$$

$$\mathbf{R} = \mathbf{I}_n - \mathbf{Z}^\top (\mathbf{Z}\mathbf{Z}^\top + \lambda_1 \mathbf{I}_{d+1})^{-1} \mathbf{Z}$$

$$\mathbf{H} = (\mathbf{Z}\mathbf{Z}^\top + \lambda_1 \mathbf{I}_{d+1})^{-1} \mathbf{Z}$$

- We can efficiently solve it with Hazan's algorithm.

Frank-Wolfe algorithm

Frank & Wolfe 1956

Algorithm 1 Frank Wolfe 1956

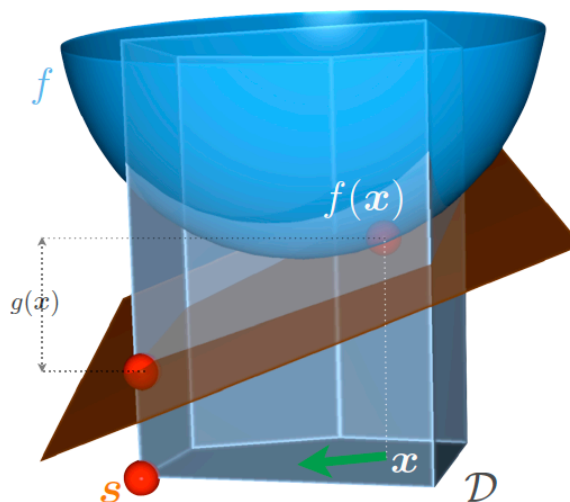
Let $\mathbf{x}^{(0)} \in \mathcal{D}$

for all $t = 0, 1, \dots, T$ **do**

 Compute $\mathbf{s} = \operatorname{argmin}_{\mathbf{s} \in \mathcal{D}} \langle \mathbf{s}, \nabla f(\mathbf{x}) \rangle$.

 Update $\mathbf{x}^{(t+1)} := (1 - \gamma)\mathbf{x}^{(t)} + \gamma\mathbf{s}$, $\gamma ::= \frac{2}{t+2}$.

end for



Jaggi ICML 2013
の図から転載

	Convergence	Computation	Memory
FW	$O\left(\frac{1}{\epsilon}\right)$	$O(n)$	$(U + I)T$
SVT (ADMM)	$O\left(\frac{1}{\sqrt{\epsilon}}\right)$	$O(U I ^2)$	$ U \times I $

Hazan's Algorithm (Frank-Wolfe)

Algorithm 1 CFM with Hazan's Algorithm

Rescale loss function $J_\eta(\mathbf{W}) = J(\eta\mathbf{W})$.

Initialize $\mathbf{W}^{(1)}$, the curvature parameter $C_f = 1$, and the number of iterations T .

for all $t = 0, 1, \dots, T$ **do**

 Compute $\mathbf{p}^{(t)} = \text{approxEV}\left(-\nabla J_\eta(\mathbf{W}^{(t)}), \frac{C_f}{(t+1)^2}\right)$.

$\hat{\alpha}_t := \frac{2}{t+2}$ (or $\hat{\alpha}_t = \text{argmin}_\alpha J_\eta(\mathbf{W}^{(t)} + \alpha(\mathbf{p}^{(t)}\mathbf{p}^{(t)\top} - \mathbf{W}^{(t)}))$).

$\mathbf{W}^{(t+1)} = \mathbf{W}^{(t)} + \hat{\alpha}_t(\mathbf{p}^{(t)}\mathbf{p}^{(t)\top} - \mathbf{W}^{(t)})$.

end for

return $\mathbf{W}^{(T)}$.

- **Key point:** The approximated eigenvalue decomposition can be estimated efficiently!

$$\nabla J(\mathbf{W}^{(t)}) = \mathbf{X}\mathbf{D}^{(t)}\mathbf{X}^\top, \mathbf{D}^{(t)} = \text{diag}(\bar{\mathbf{y}}^{(t)} - \mathbf{Z}^\top \hat{\mathbf{w}}^{(t)}),$$

$$\hat{\mathbf{w}}^{(t)} = (\mathbf{Z}\mathbf{Z}^\top)^{-1} \mathbf{Z}\bar{\mathbf{y}}^{(t)}, \bar{\mathbf{y}}^{(t)} = \mathbf{y} - \mathbf{f}_Q(\mathbf{X}; \mathbf{W}^{(t)})$$

CFM for tensor completion

- Tucker-based decomposition can be written in FM format!

$$\min_{\{\mathcal{M}^{(m)}\}_{m=0}^3} \sum_{(i,j,k) \in \Omega} \left([\mathcal{Y}]_{i,j,k} - [\mathcal{M}^{(0)}]_{i,j,k} - \sum_{m=1}^3 [\mathcal{M}^{(m)}]_{i,j,k} \right)^2 + \lambda \sum_{m=1}^3 \|\mathbf{M}_{(m)}^{(m)}\|_{\text{tr}}$$

$$\sum_{m=1}^3 [\mathcal{M}^{(m)}]_{i,j,k} = \frac{1}{2} \text{tr} (\mathbf{W} (\mathbf{x}_{i,j,k} \mathbf{x}_{i,j,k}^\top - \text{diag}(\mathbf{x}_{i,j,k} \circ \mathbf{x}_{i,j,k}))),$$

$$\mathbf{x}_{i,j,k}^{(1)} = \left[\underbrace{0 \cdots 0}_{n_1} \underbrace{1}_{i} \underbrace{0 \cdots 0}_{n_2 n_3} \quad \underbrace{0 \cdots 0}_{n_2(k-1)+j} \underbrace{1}_{n_2(k-1)+j} \underbrace{0 \cdots 0}_{n_1 n_3} \right]^\top,$$

$$\mathbf{x}_{i,j,k}^{(2)} = \left[\underbrace{0 \cdots 0}_{n_2} \underbrace{1}_{j} \underbrace{0 \cdots 0}_{n_1 n_3} \quad \underbrace{0 \cdots 0}_{n_3(i-1)+k} \underbrace{1}_{n_3(i-1)+k} \underbrace{0 \cdots 0}_{n_1 n_2} \right]^\top,$$

$$\mathbf{x}_{i,j,k}^{(3)} = \left[\underbrace{0 \cdots 0}_{n_3} \underbrace{1}_{k} \underbrace{0 \cdots 0}_{n_1 n_2} \quad \underbrace{0 \cdots 0}_{n_1(j-1)+i} \underbrace{1}_{n_1(j-1)+i} \underbrace{0 \cdots 0}_{n_2 n_3} \right]^\top,$$

$$\mathbf{x}_{i,j,k} = [\mathbf{x}_{i,j,k}^{(1)\top} \quad \mathbf{x}_{i,j,k}^{(2)\top} \quad \mathbf{x}_{i,j,k}^{(3)\top}]^\top \in \mathbb{R}^d.$$

$$[\mathcal{M}^{(1)}]_{i,j,k} = [\mathbf{M}_{(1)}^{(1)}]_{i, n_2(k-1)+j},$$

$$[\mathcal{M}^{(2)}]_{i,j,k} = [\mathbf{M}_{(2)}^{(2)}]_{j, n_3(i-1)+k},$$

$$[\mathcal{M}^{(3)}]_{i,j,k} = [\mathbf{M}_{(3)}^{(3)}]_{k, n_1(j-1)+i}.$$

- We can also write another tensor completion by FM.

Comparison to existing work

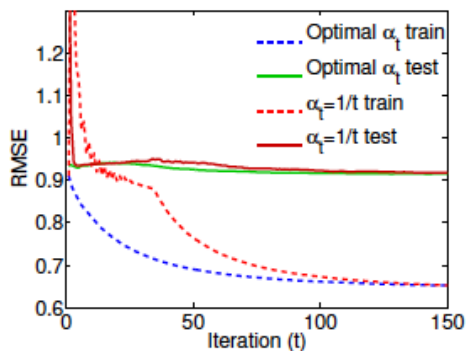
- Comparison to Blondel et al. 2015

	Loss function	Optimization	Complexity	Tensor Completion
This work	Regression	Frank Wolfe \mathbf{W}	$O(Tn)$	\circ
Blondel et al. 2015	Regression/Classification	Coordinate descent w, \mathbf{W}	$O(T_{out}T_{in}n)$??

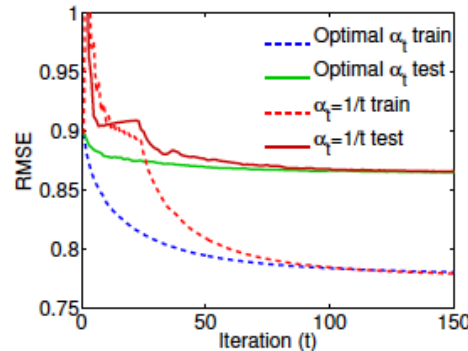
- Our method is much more simpler.
- Can be further speed up easily if we have better Frank Wolfe algorithms.
- We applied CFM for biology data.

Movie Lens

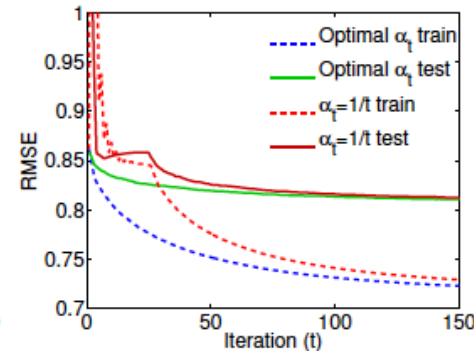
- Movielens (100000 ~ 20million Ratings)



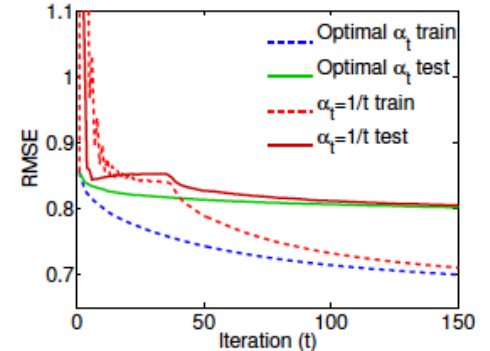
(a) Movielens 100K data.



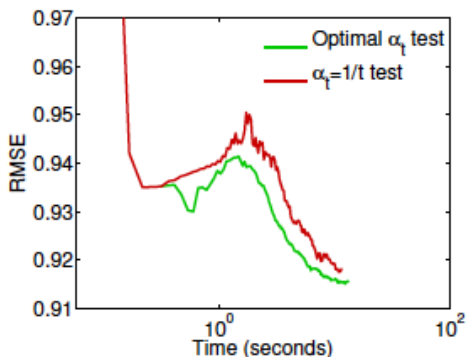
(b) Movielens 1M data.



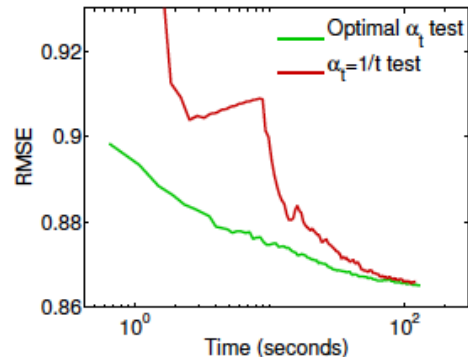
(c) Movielens 10M data.



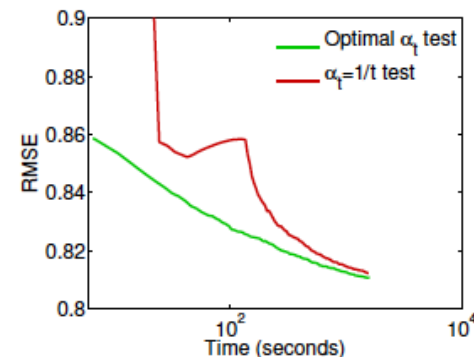
(d) Movielens 20M data.



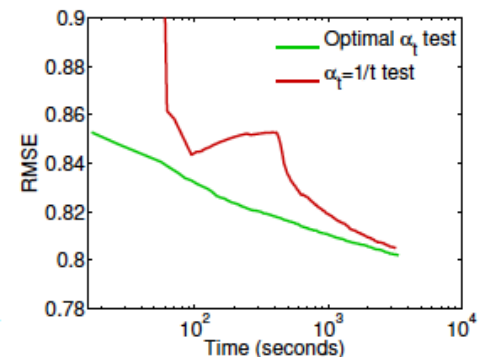
(a) Movielens 100K data.



(b) Movielens 1M data.



(c) Movielens 10M data.



(d) Movielens 20M data.

Toxicogenomics Prediction

- Predict toxicity from data (Completion)

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_1^{(1)} & \mathbf{A}_1^{(2)} \\ \mathbf{A}_2^{(1)} & \mathbf{A}_2^{(2)} \\ \mathbf{A}_3^{(1)} & \mathbf{A}_3^{(2)} \end{bmatrix}$$

Genome data Toxicity

$$\mathbf{x} = \left[\underbrace{0 \dots 0}_{3327} \underbrace{1}_{i\text{-th gene}} \underbrace{0 \dots 0}_{78} \underbrace{0 \dots 0}_{78} \underbrace{1}_{j\text{-th drug}} \underbrace{0 \dots 0}_{2} \underbrace{1}_{1\text{st view}} \underbrace{0}_{1} \right]^T,$$

$$y = [\mathbf{A}]_{i,j},$$

	Multi-view				Single-view		
	CFM	BMTF	GFA	ARDCP	CP	ARDCP	CP
Mean	0.4037	0.4811	0.5223	0.8919	5.3713	0.6438	5.0699
StdError	0.0163	0.0061	0.0041	0.0027	0.0310	0.0047	0.0282

Toxicogenomics Prediction

- Predict toxicity from data (Prediction)

$$A = \begin{bmatrix} A_1^{(1)} & A_1^{(2)} \\ A_2^{(1)} & A_2^{(2)} \\ A_3^{(1)} & A_3^{(2)} \end{bmatrix}$$

Genome data Toxicity

$$x = \left[\underbrace{0 \dots 0}_{78} \underbrace{1}_{i\text{-th drug}} \underbrace{0 \dots 0}_{78} \underbrace{0 \quad 1 \quad 0}_{3} \underbrace{0 \quad 1 \quad 0}_{3} \underbrace{x_{\text{mean}} \quad x_{\text{std}}}_{2} \right]^T,$$

$k\text{-th sensit.}$ $l\text{-th cancer}$

$$y = [A_l^{(2)}]_{i,k}$$

	CFM	CFM (+mean/std features)			CFM (+mean feature)		
		$m = 5$	$m = 10$	$m = 15$	$m = 5$	$m = 10$	$m = 15$
Mean	0.5624	0.5199	0.5207	0.5215	0.5269	0.5234	0.5231
StdError	0.0501	0.0464	0.0451	0.0450	0.0466	0.0454	0.0450

Takehome message

- Convex factorization Machine
 - We propose a convex variant of FMs, which is convex and can get a **globally optimal solution**.
 - The proposed algorithm is **simple yet efficient**.
 - We show a **convex Tucker-based tensor completion** can be solved through CFM.
 - We applied the CFM to a **toxicogenomics prediction** task. Then, it outperformed a state-of-the-art method.
- The python code is available below:
<http://www.makotoyamada-ml.com/cfm.html>