Glancing over 20 Papers on Modeling Feature Interactions

Bokai Cao
12/30/2017

Multi-View Learning

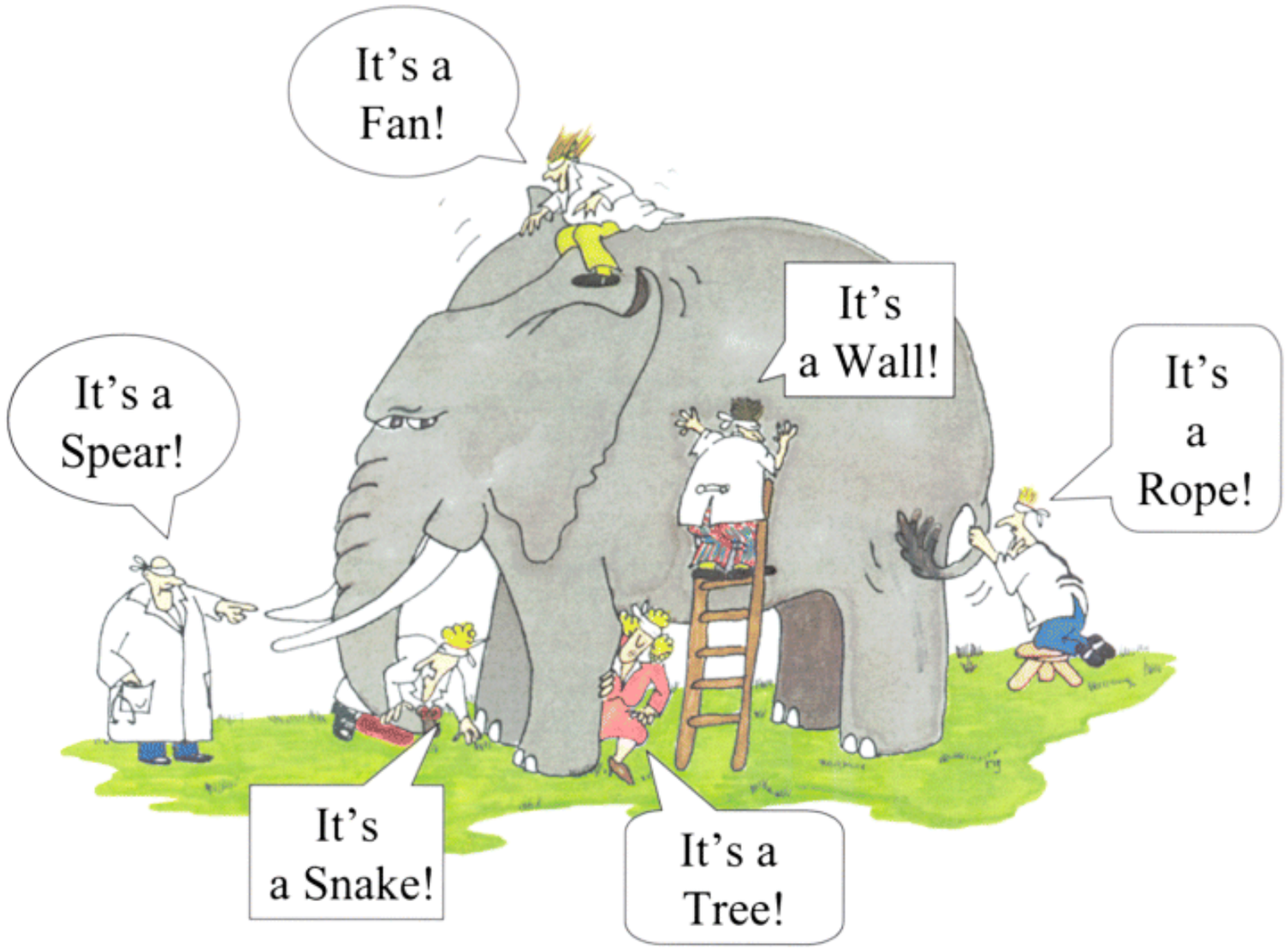
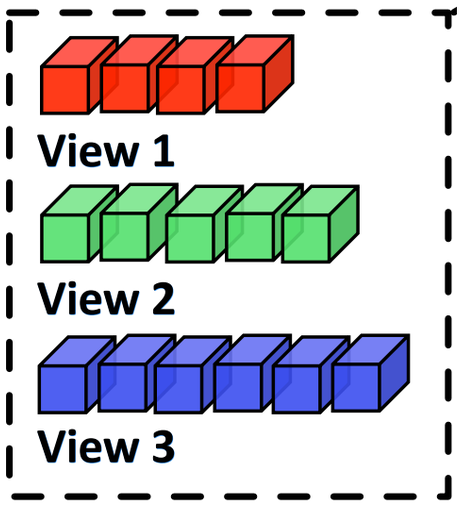


Image source: blind men and the elephant

Multi-View Learning

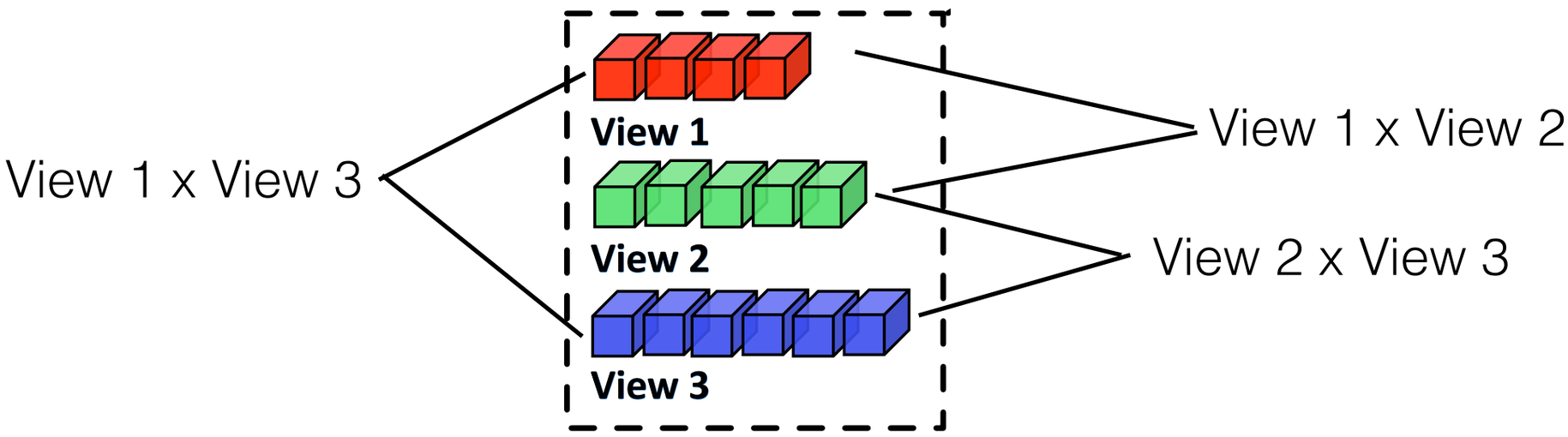
- (Definition) An instance is represented by multiple groups of features, and different feature groups are usually collected from different data sources, thereby having different physical meanings and statistical distributions.

An instance

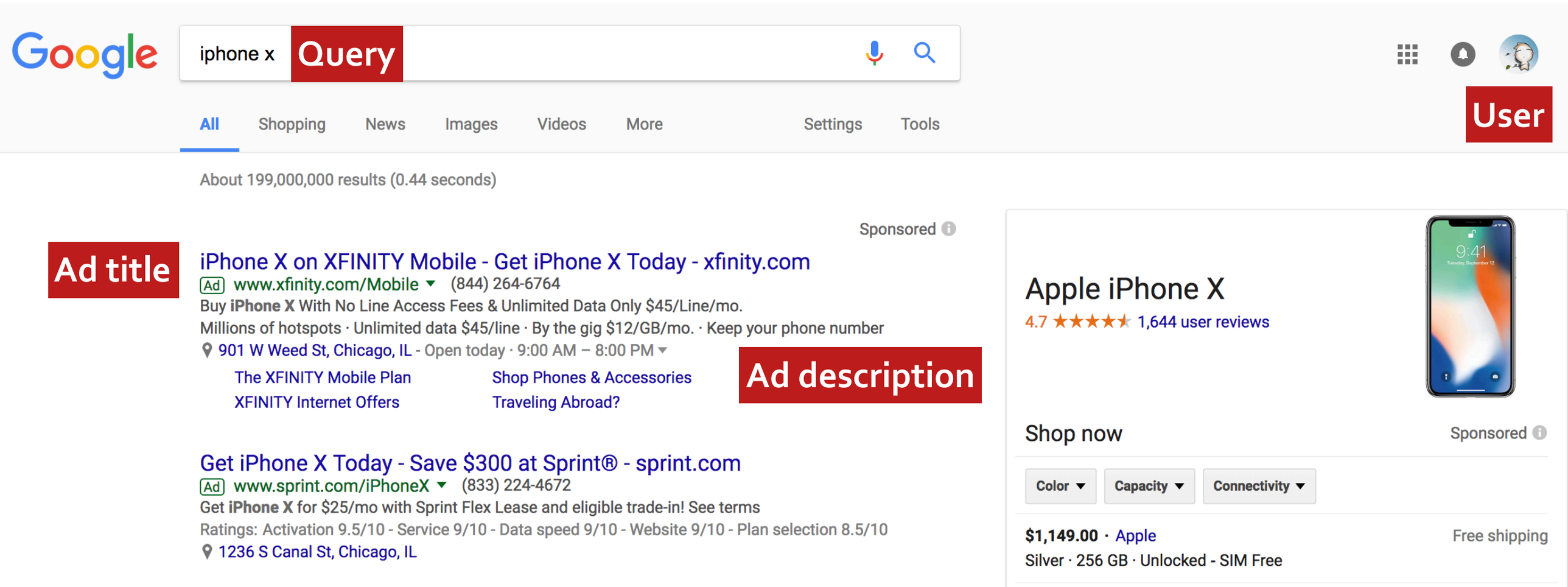


Multi-View Learning

- (Assumption) Features within the same view are usually correlated, while features from different views are complementary. Therefore, it is critical to consider inter-view feature interactions.



An Example



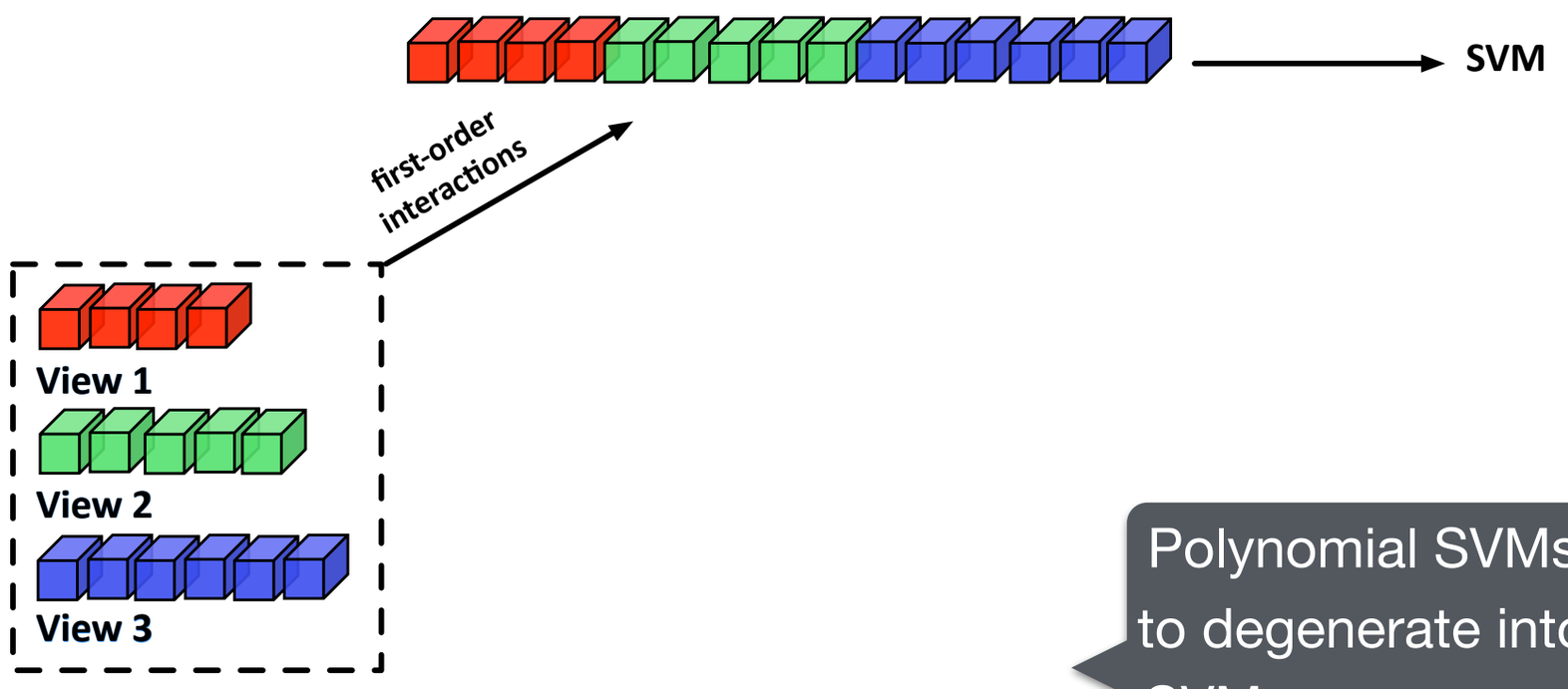
Given a query, how likely a user will click on an ad?

Feature Interactions

- Problem: How to model feature interactions?
- A common solution: Manually create combinatorial/ cross features and then feed them into downstream machine learning models.
  - Query X Ad title, Query X User, or Query X Ad description X User, …
  - Expensive to create, maintain, and improve.
  - Need domain knowledge.

Support Vector Machines (SVM)

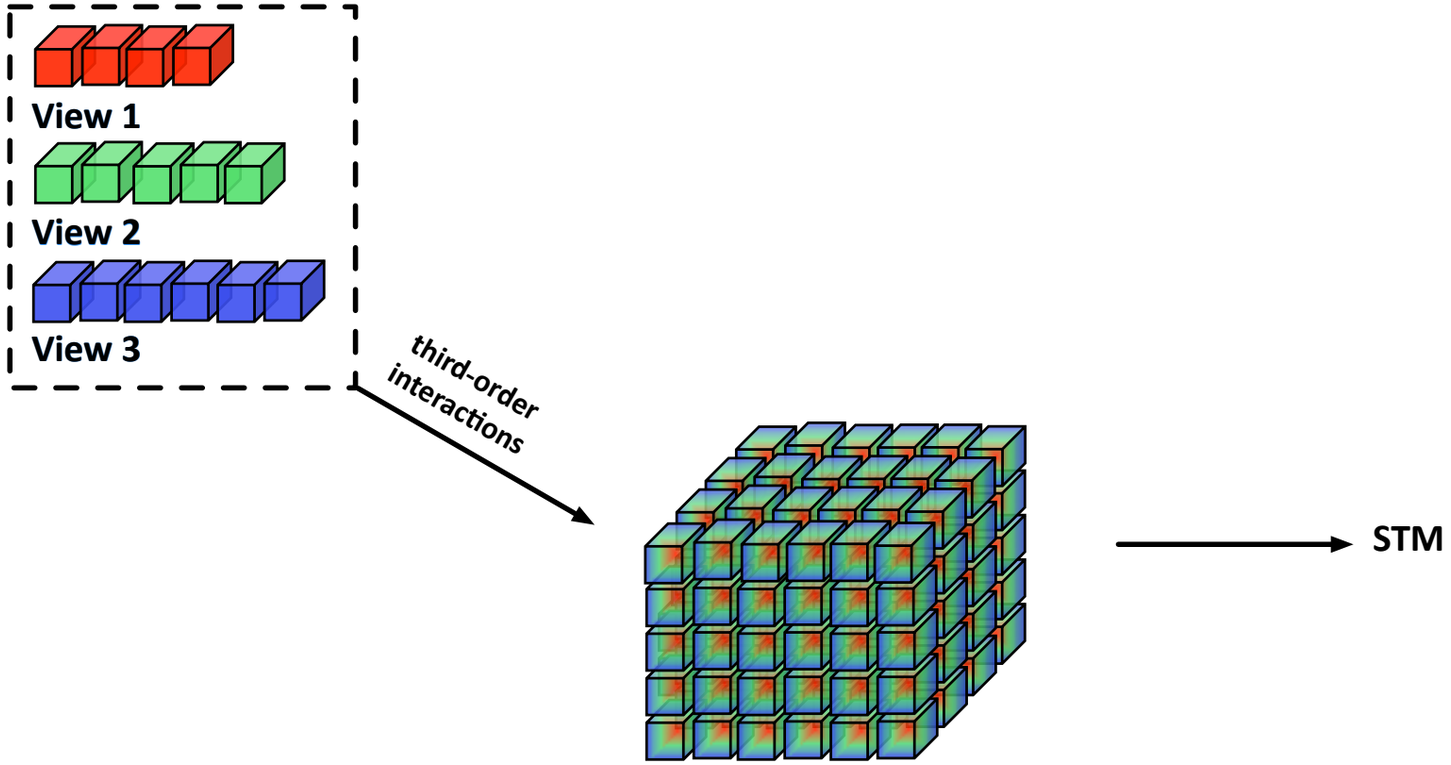
- The linear SVM model (and LR etc.) is limited to the first-order feature interactions.



Polynomial SVMs are likely to degenerate into linear SVMs on sparse datasets.

Support Tensor Machines (STM)

- Work with hinge loss only.
- Strong assumption: rank-1 weight tensor.

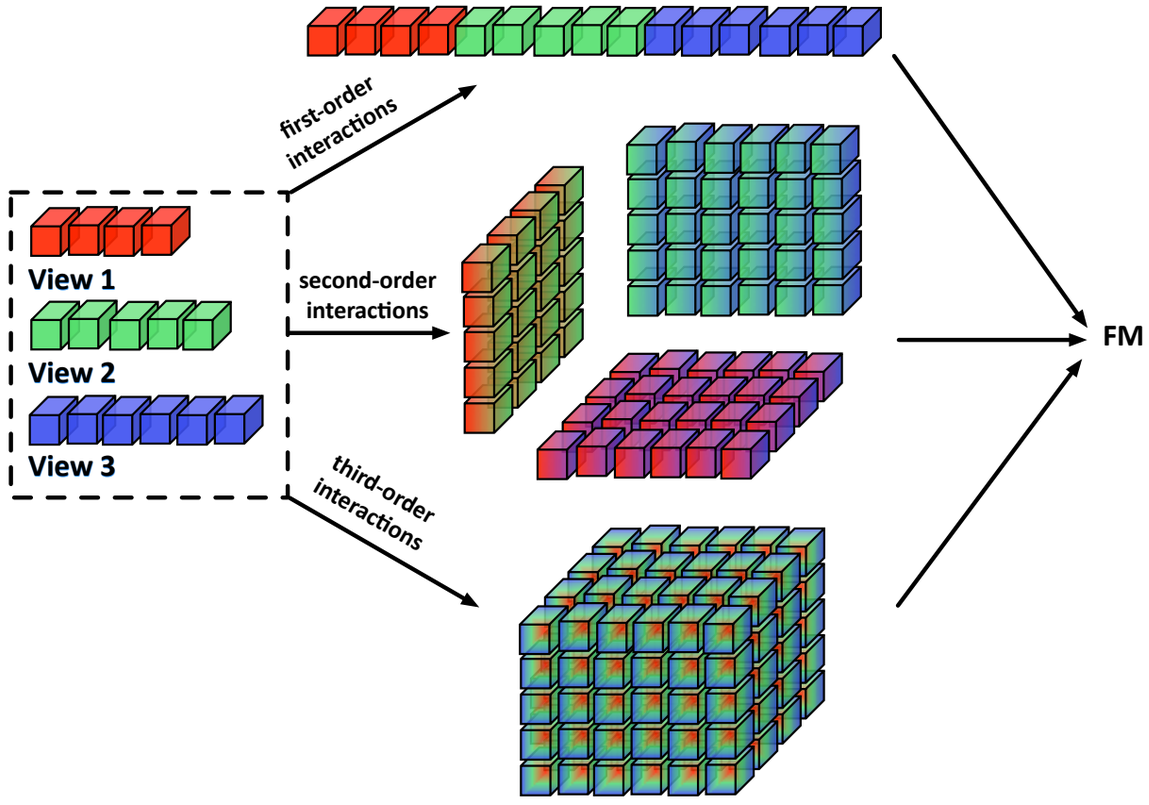


Factorization Machines (FM)

大阪大学
OSAKA UNIVERSITY

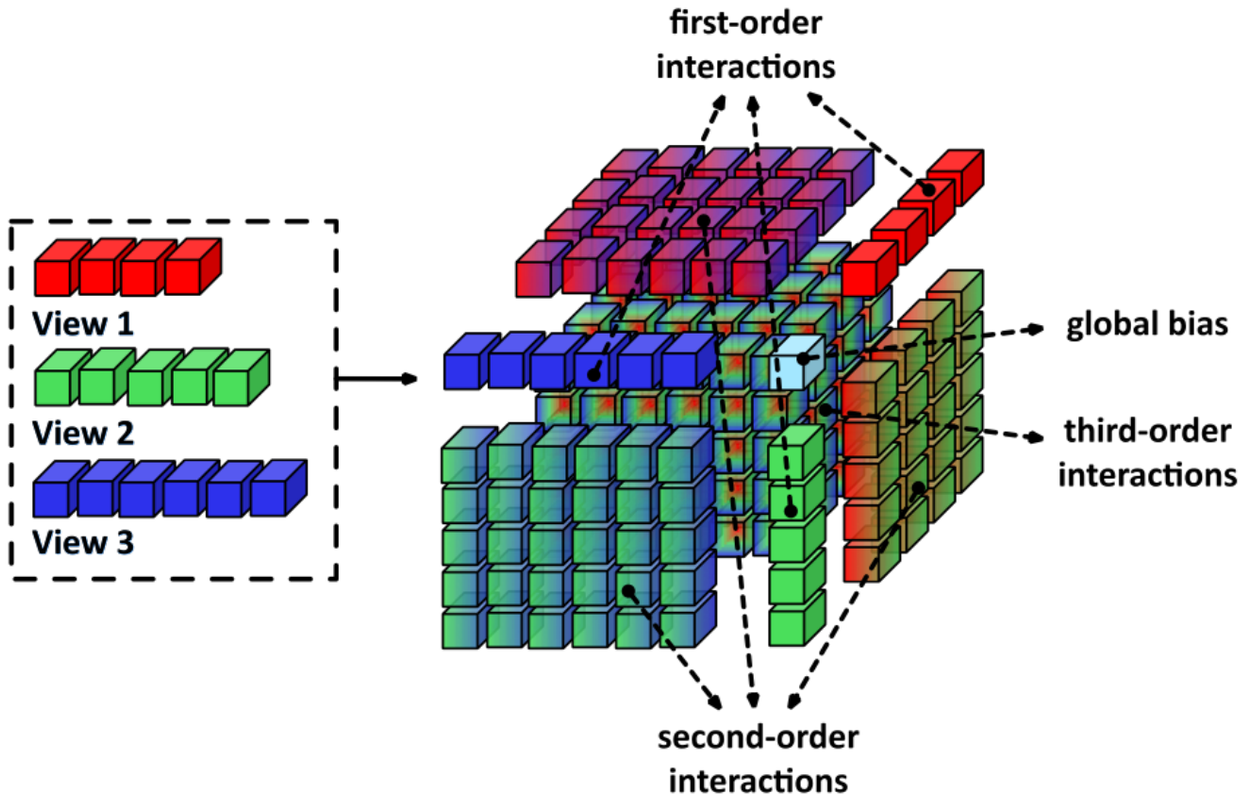
- Use separate parameters to approximate interactions in different orders.

- In practice, 2-way FM is usually used.



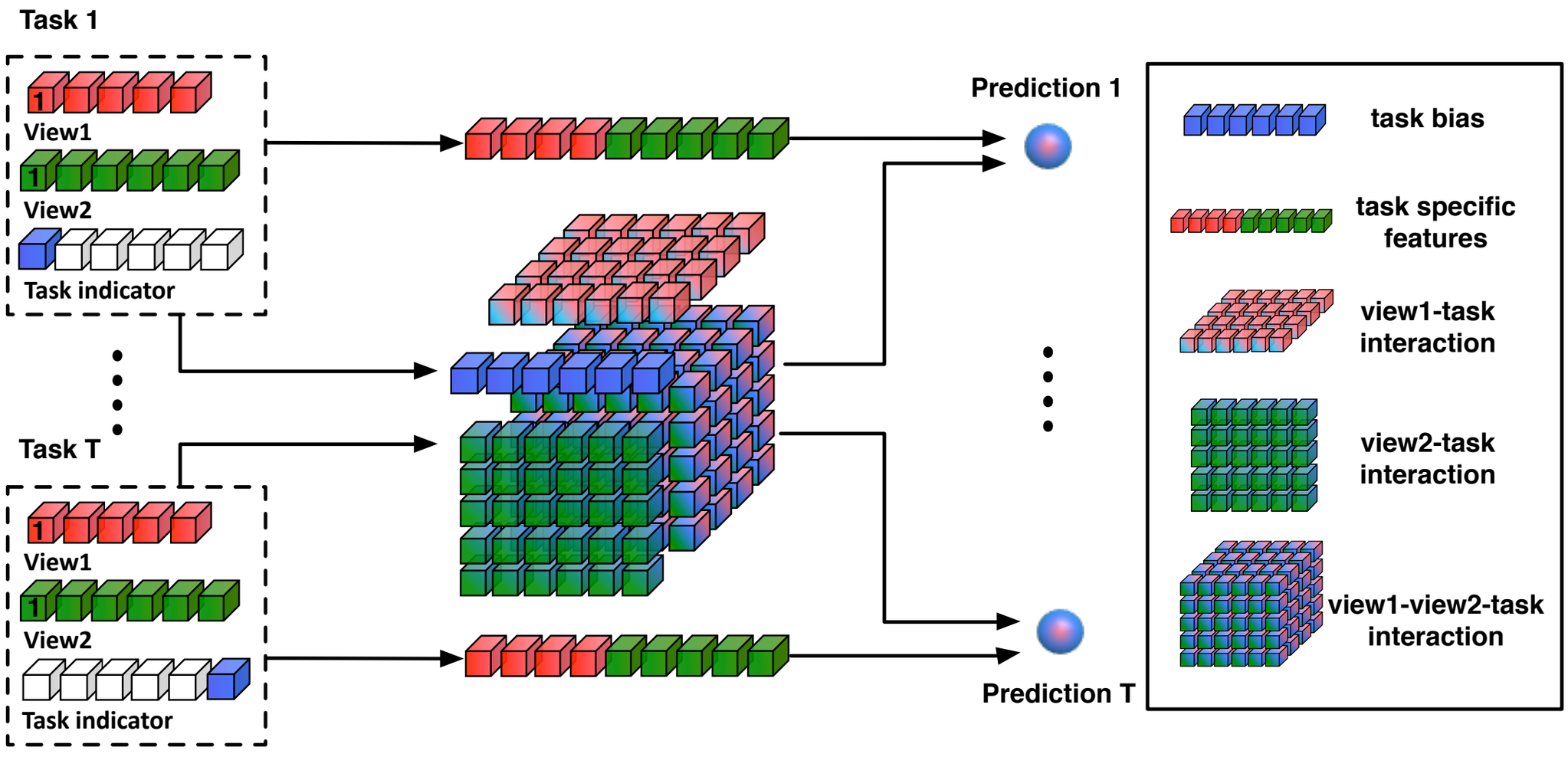
Multi-View Machines (MVM)

- Jointly model all interactions of every order.
- CP factorization on the weight tensor.



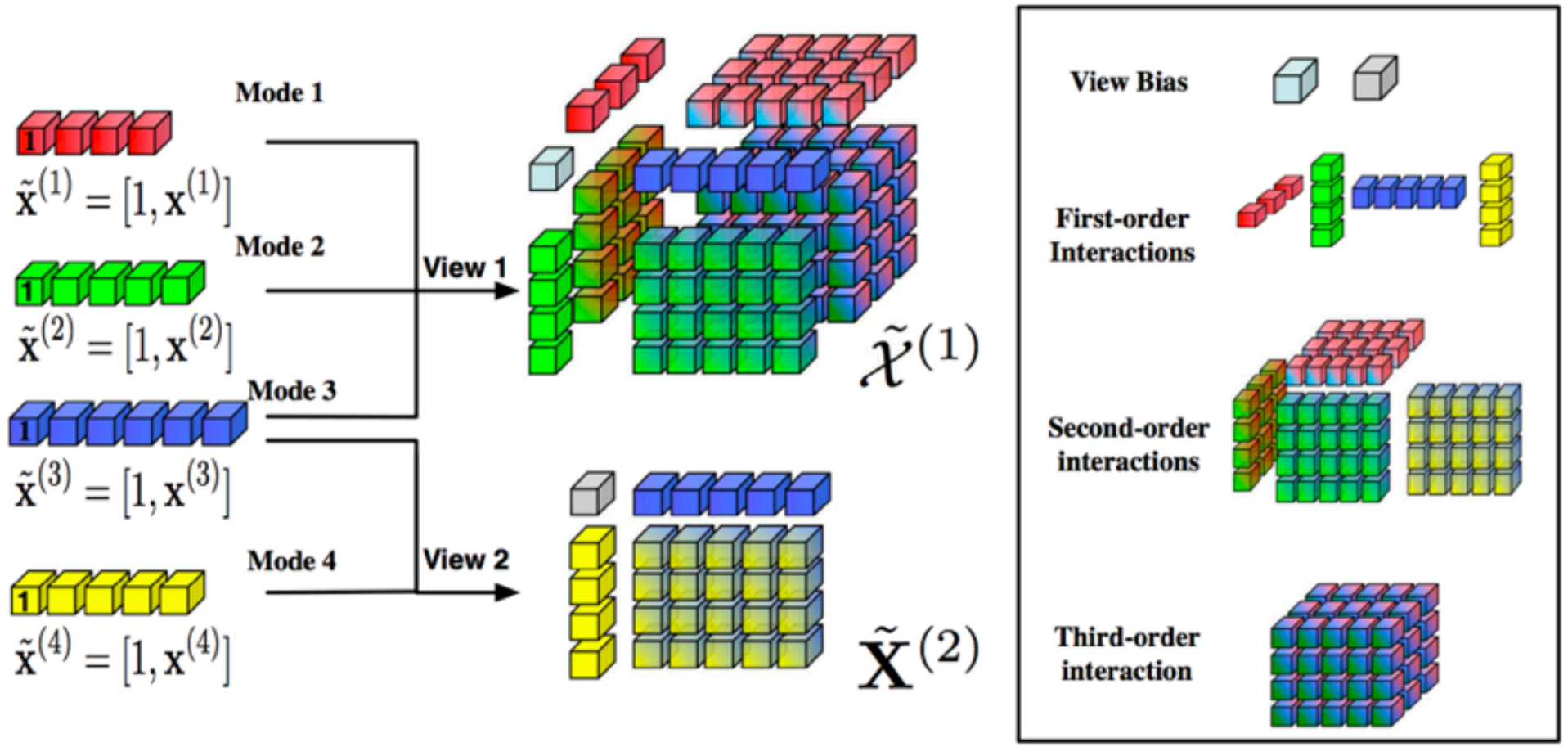
Multilinear Factorization Machines (MFM)

- Extend to multi-task learning.



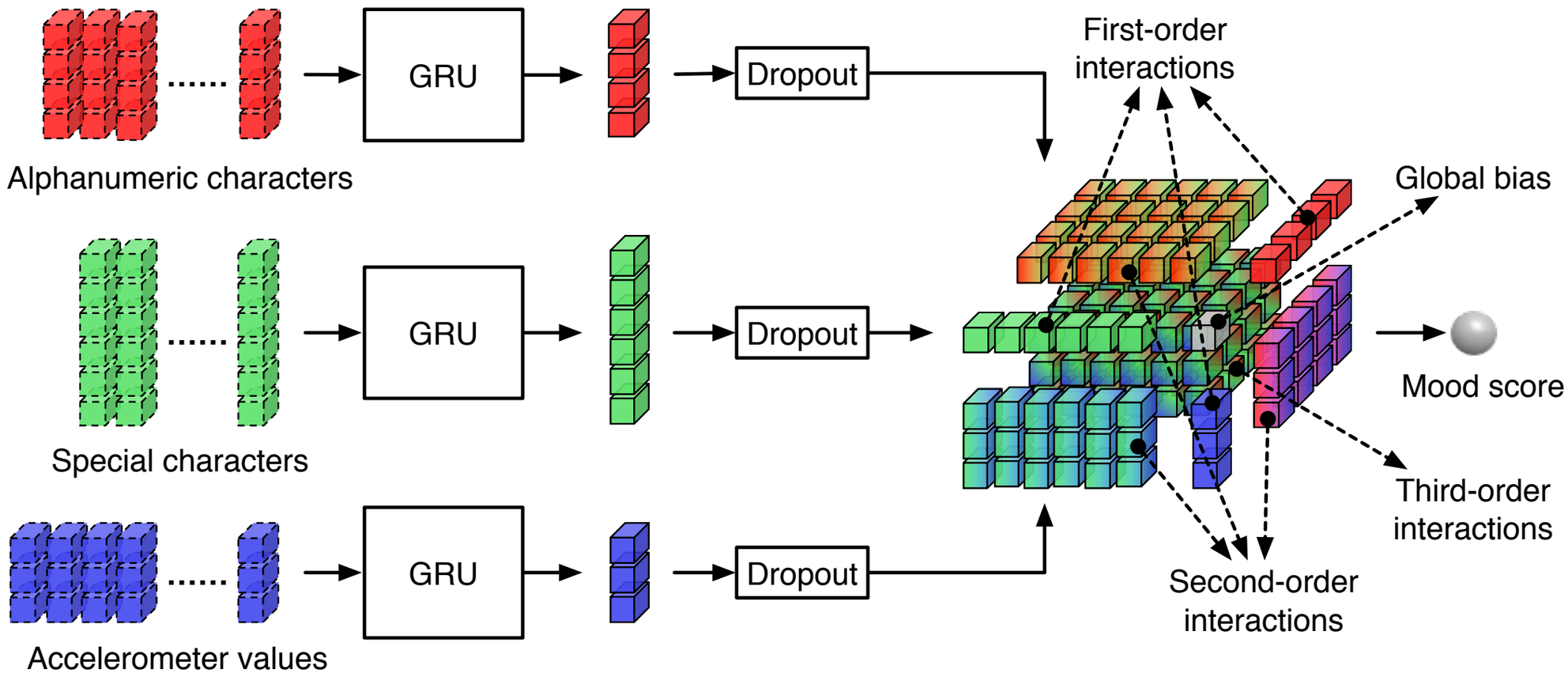
Structural Factorization Machines (SFM)

- Overlapping groups of views.



DeepMood

- Present a neural network view of FM and MVM.



Field-aware Factorization Machines (FFM)

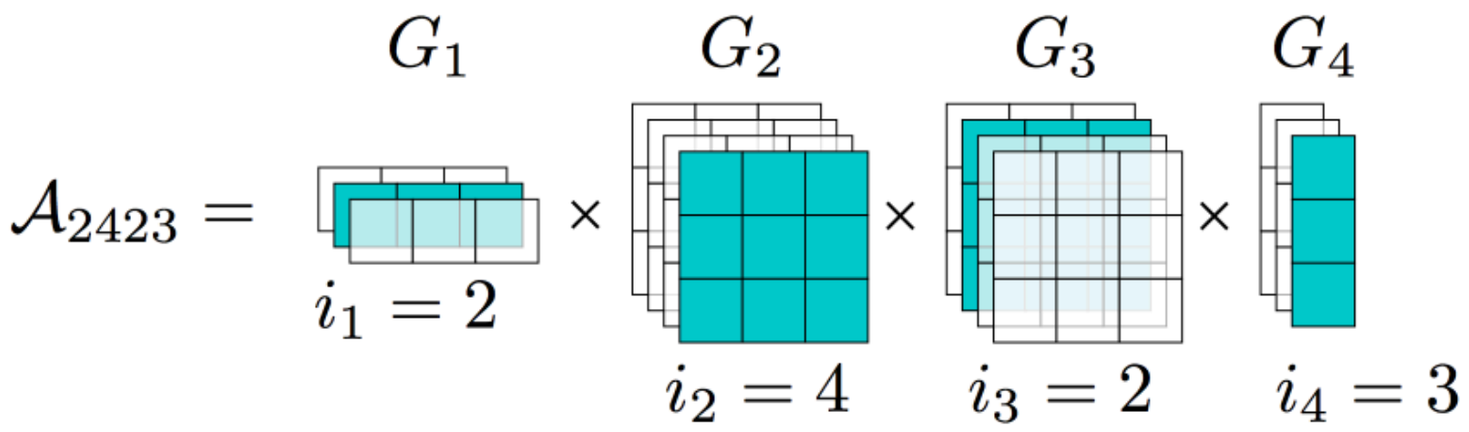
criteo

$$\phi_{\mathrm{FM}}(\boldsymbol{w}, \boldsymbol{x}) = \sum_{j_1=1}^{n} \sum_{j_2=j_1+1}^{n} (\boldsymbol{w}_{j_1} \cdot \boldsymbol{w}_{j_2}) x_{j_1} x_{j_2}$$

$$\phi_{\mathrm{FFM}}(\boldsymbol{w}, \boldsymbol{x}) = \sum_{j_1=1}^{n} \sum_{j_2=j_1+1}^{n} (\boldsymbol{w}_{j_1, f_2} \cdot \boldsymbol{w}_{j_2, f_1}) x_{j_1} x_{j_2}$$

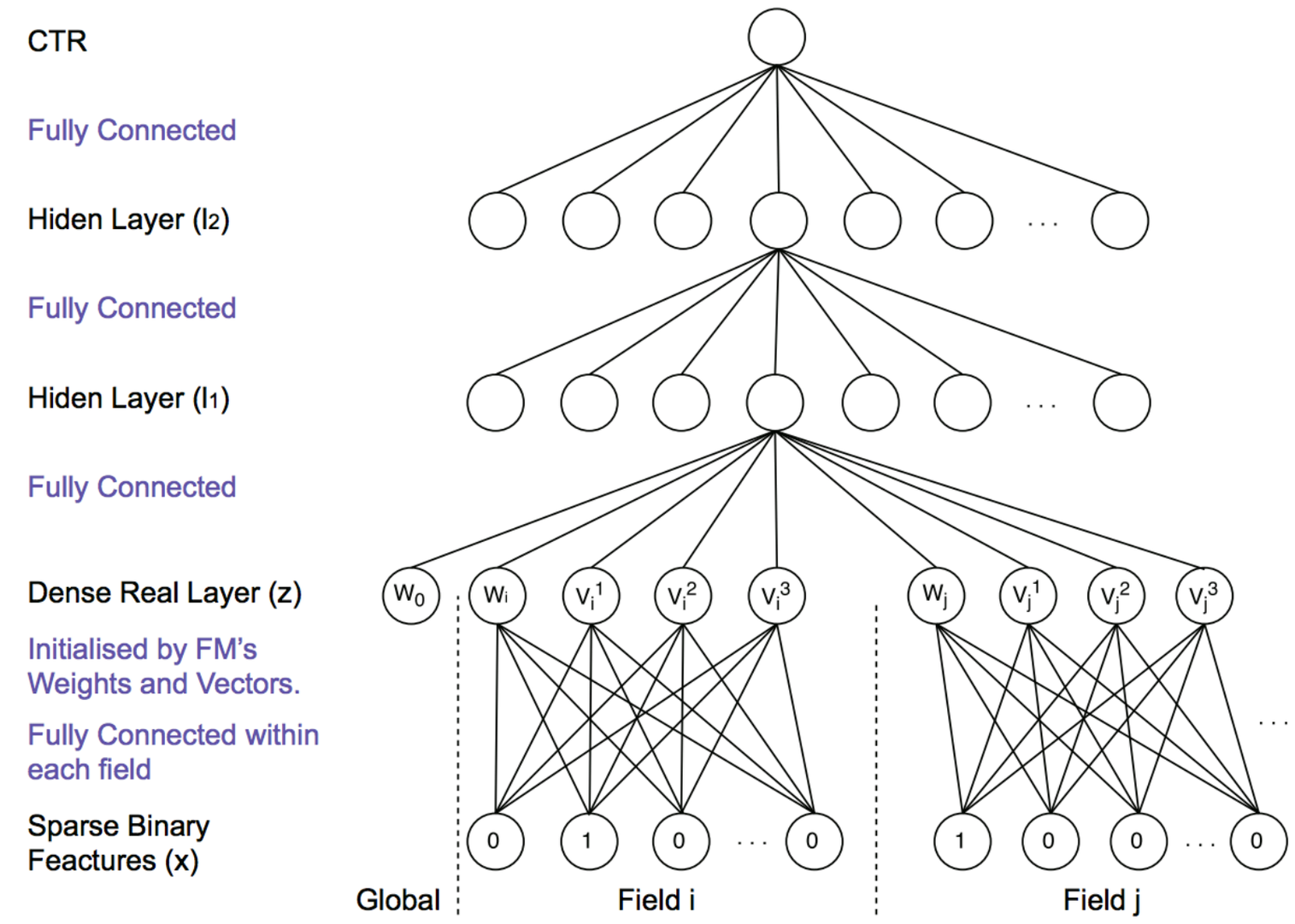
Exponential Machines (ExM)

- Model all interactions of every order.
- Represent the weight tensor in a factorized format called Tensor Train (TT).



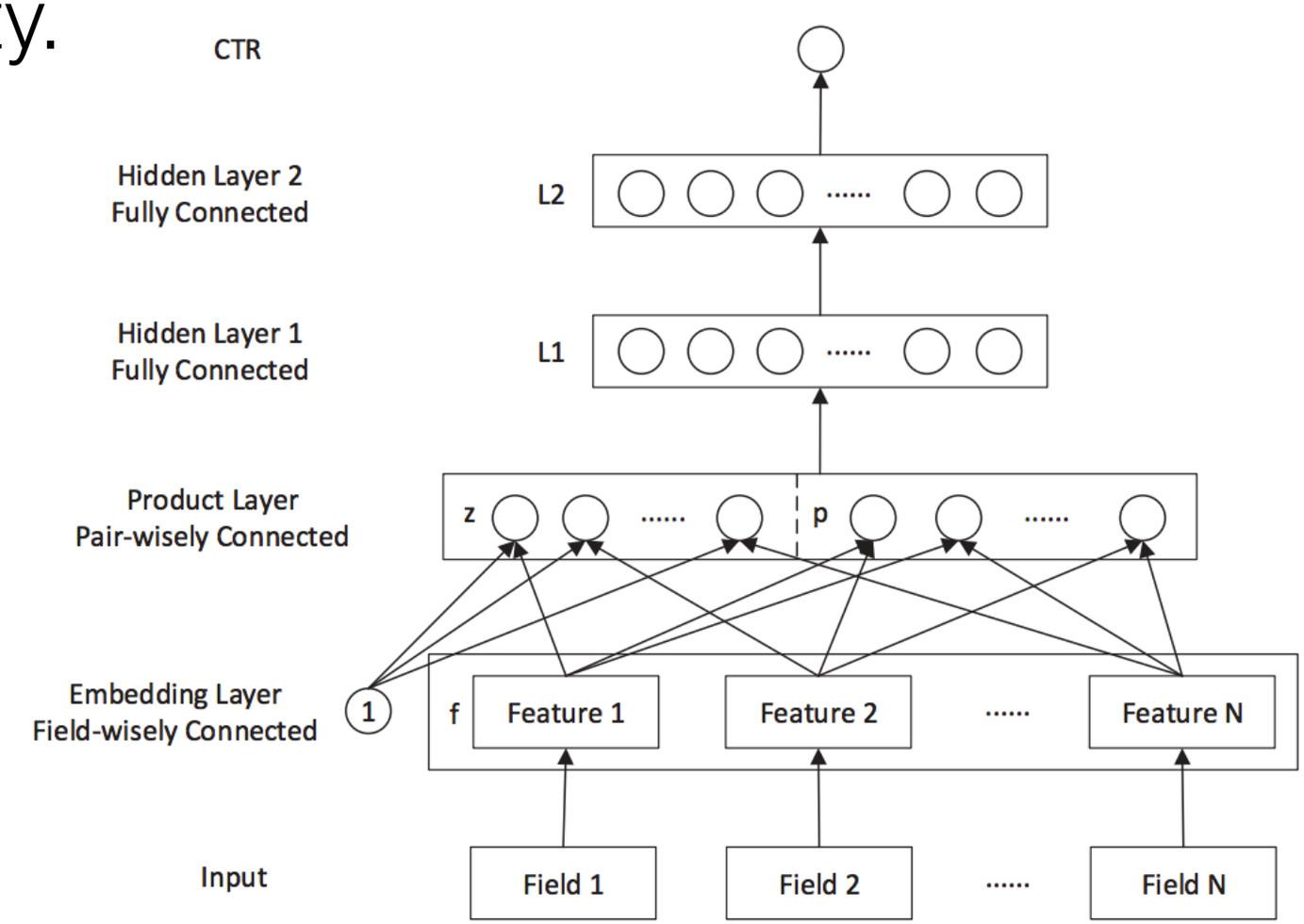
Factorization-Machine supported Neural Networks (FNN)

- Initialize the embedding layer by FM.



Product-based Neural Networks (PNN)

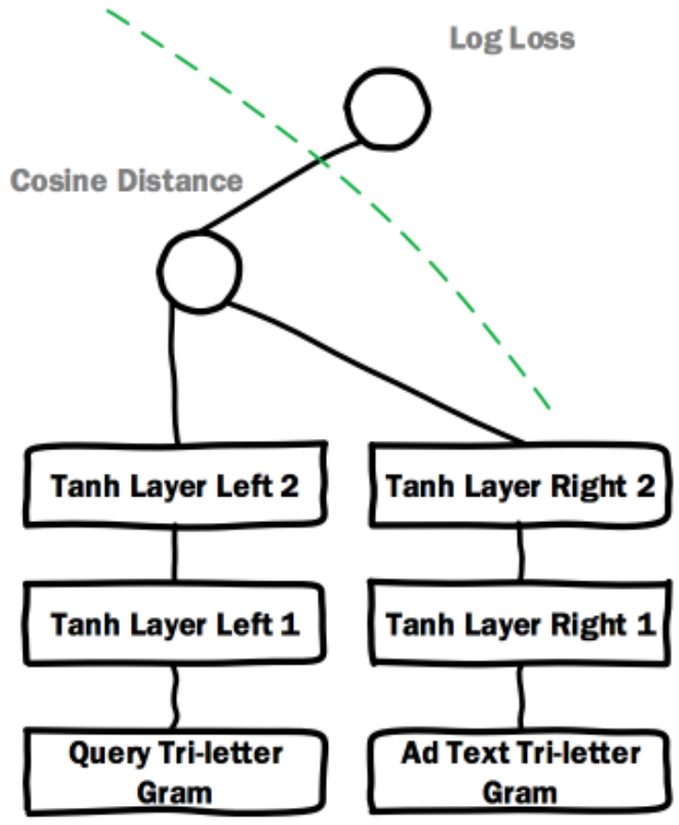
- Add a product layer to increase the model capacity.



Deep Semantic Similarity Model (DSSM)

Microsoft

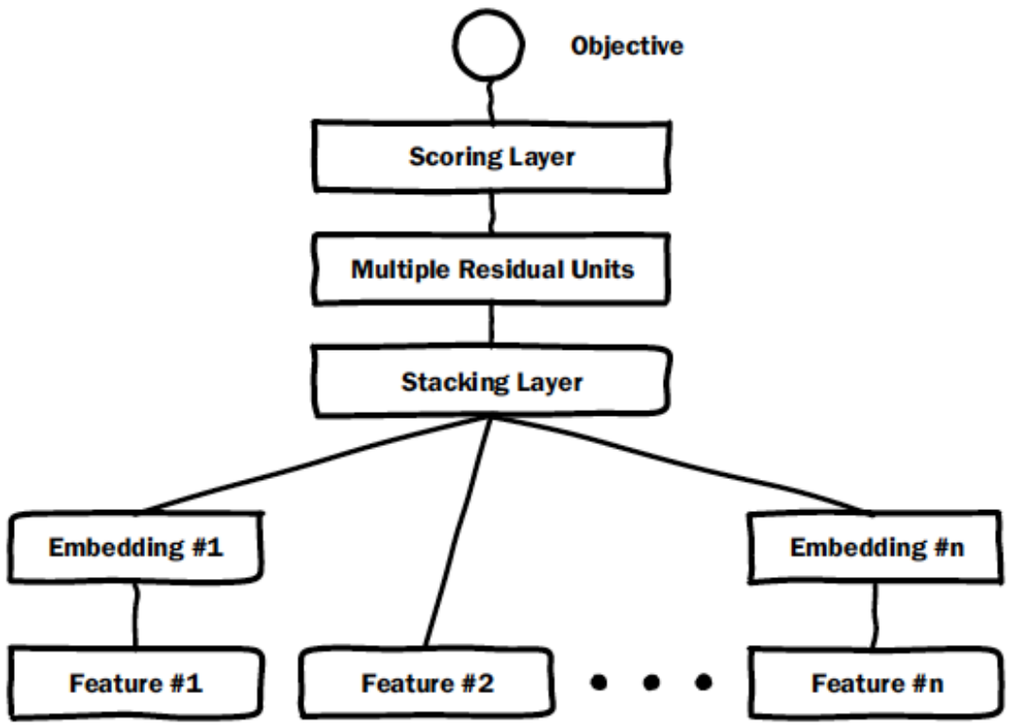
- Late crossing.



Deep Crossing

Microsoft

- Early crossing.

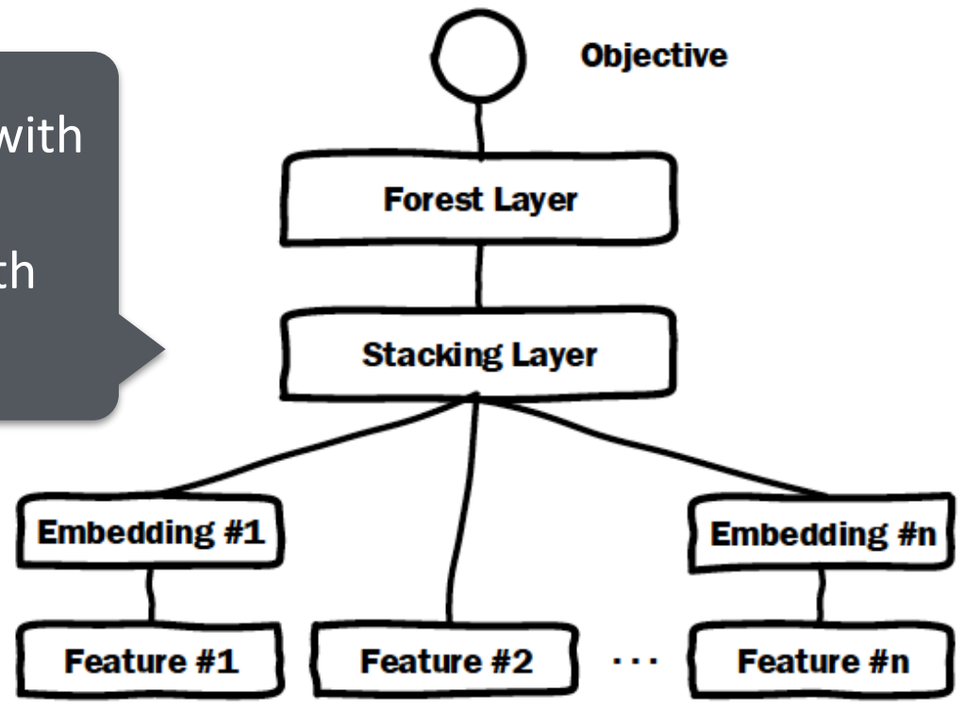


Deep Embedding Forest (DEF)

Microsoft

- Replace residual units in Deep Crossing with a forest/tree-like layer.

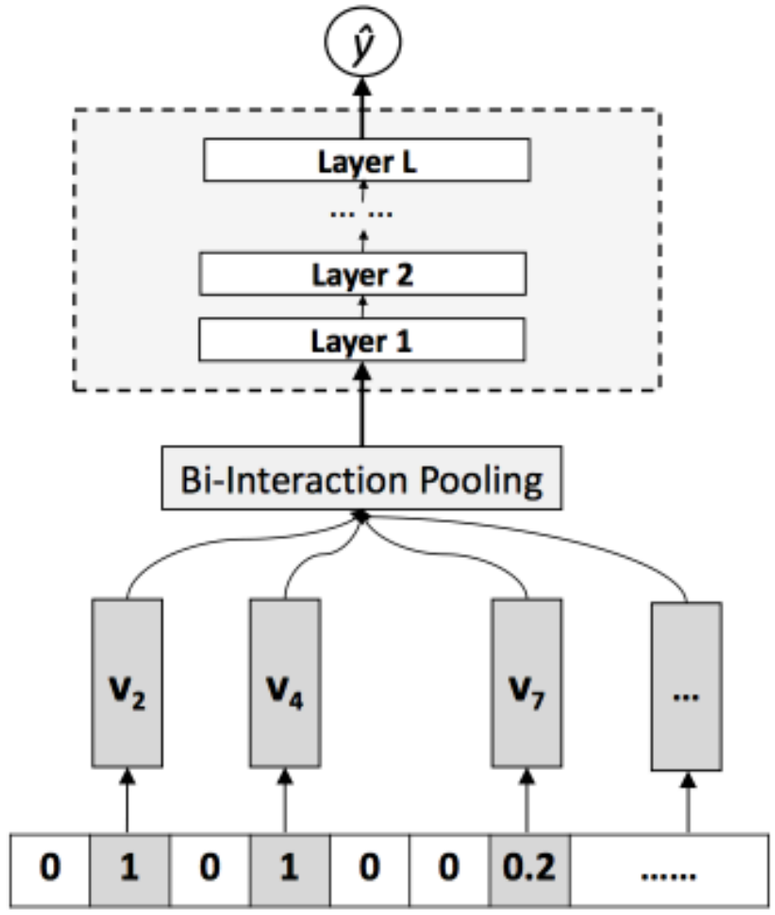
1.Initialize Embedding Layers with Deep Crossing.
2.Initialize the Forest Layer with XGBoost and LightGBM.



Neural Factorization Machines (NFM)

NUS National University of Singapore

- Replace the stacking layer in Deep Crossing with the Bi-Interaction pooling layer.



Prediction Score

Hidden Layers

B-Interaction Layer

Embedding Layer

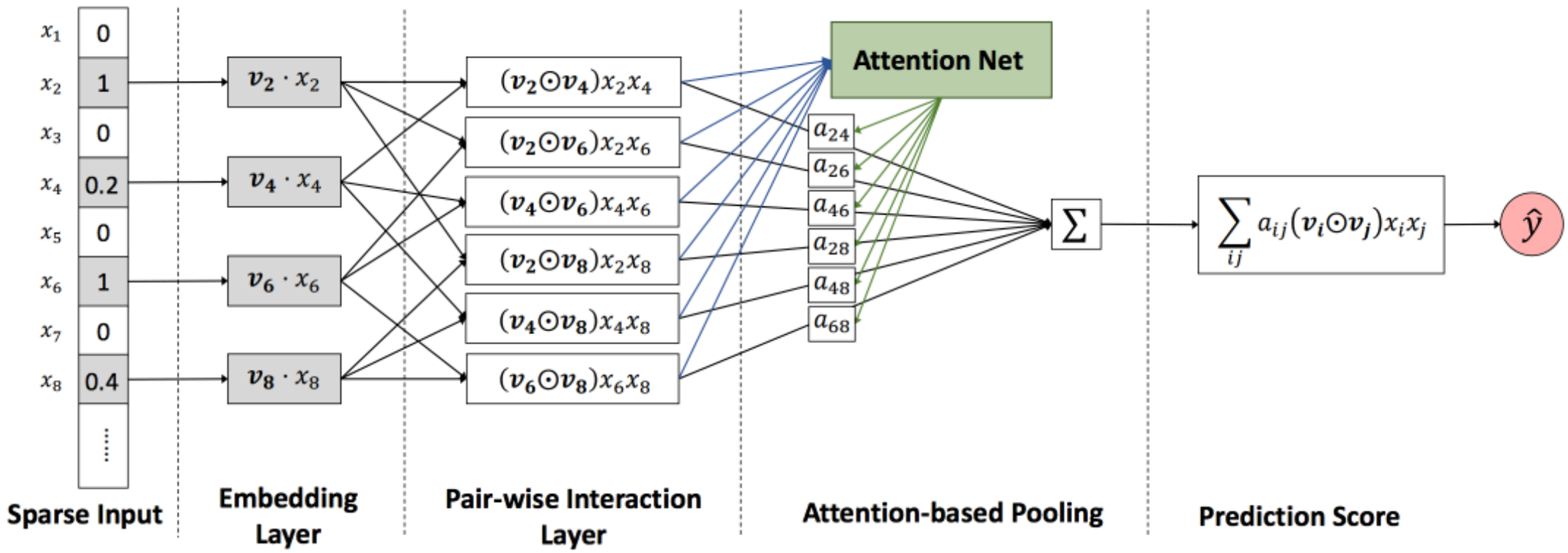
Input Feature Vector (sparse)

FM is a shallow and linear model, which can be seen as a special case of NFM with no hidden layer.

Attentional Factorization Machines (AFM)

- Apply the attention mechanism over the pair-wise interaction layer.

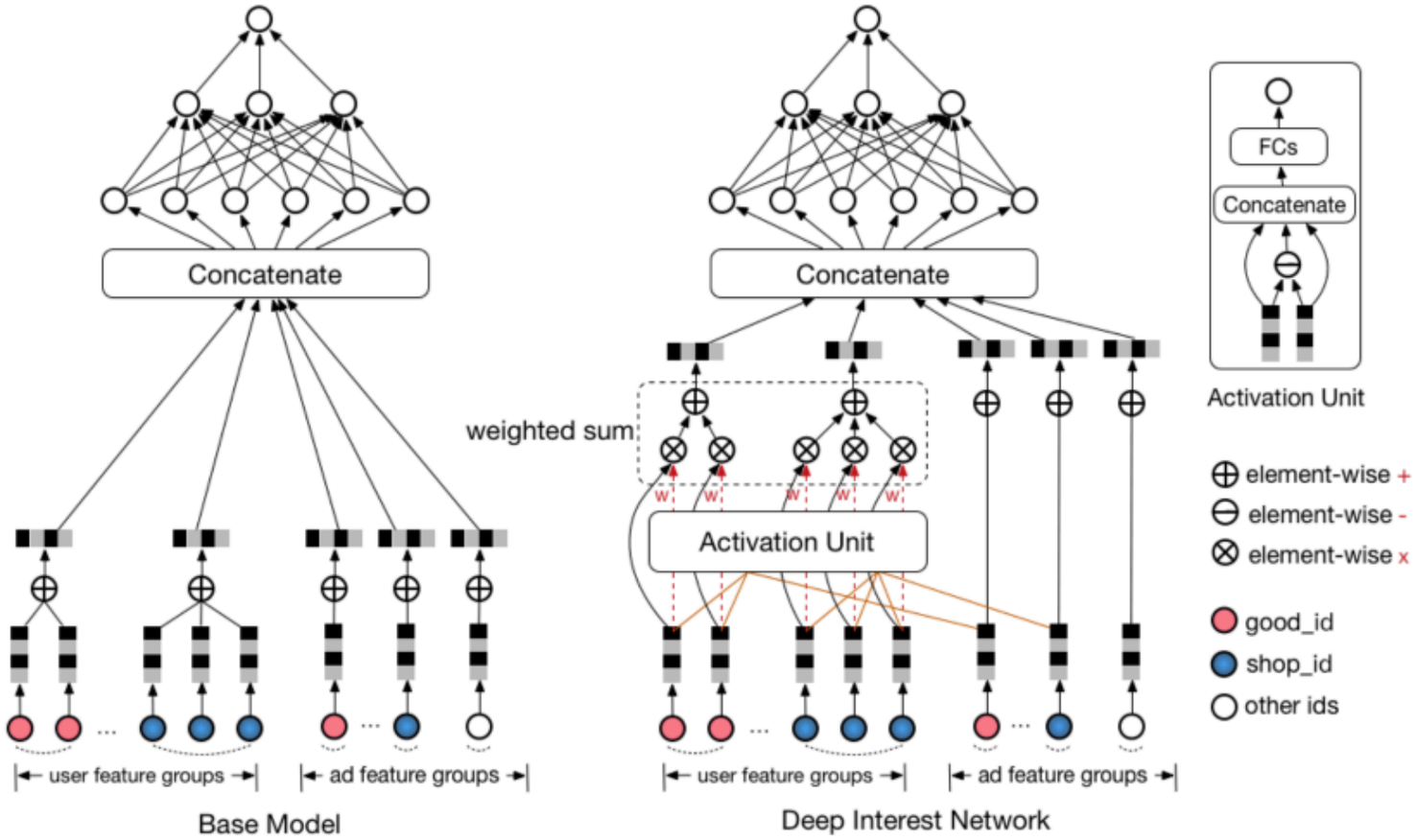
- The Bi-Interaction pooling layer can be seen as a sum pooling.



Deep Interest Network (DIN)

Alibaba Group

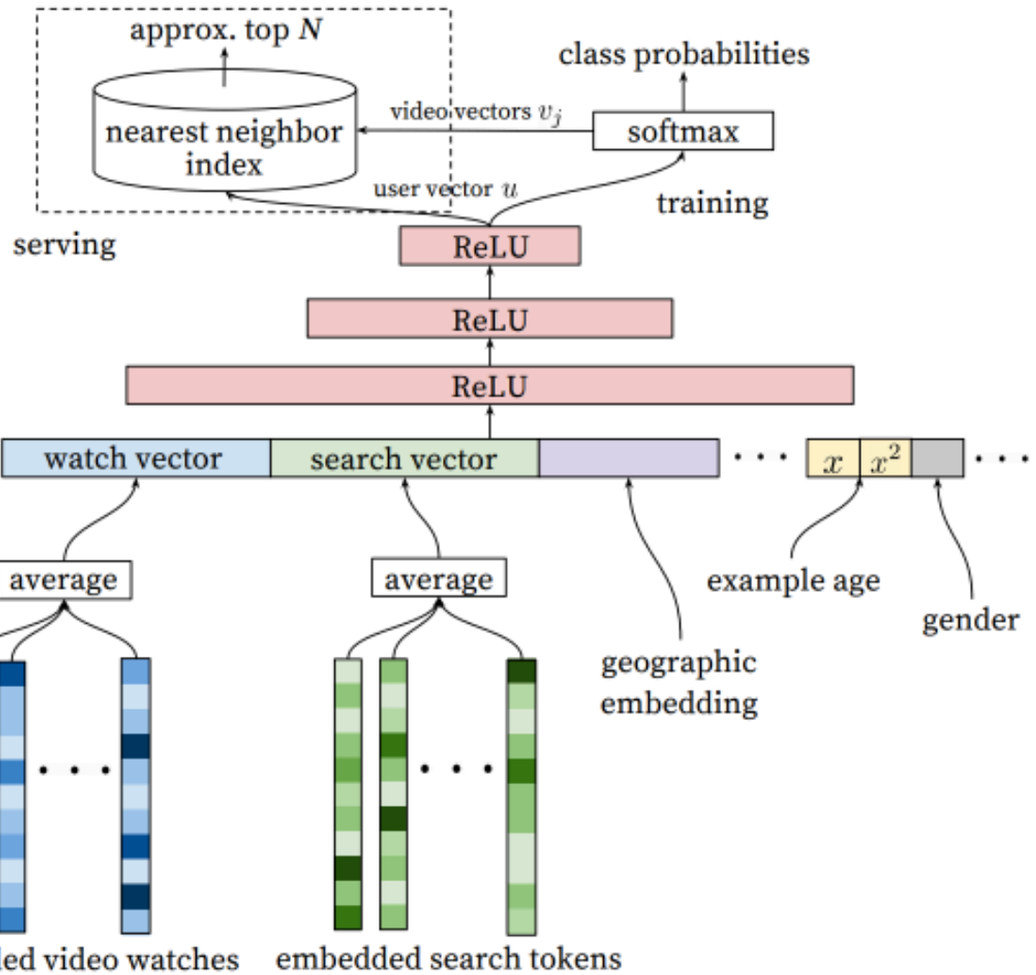
- The embedding vector of a user is a function of the embedding vector of an ad.



YouTube Recommendations

Google

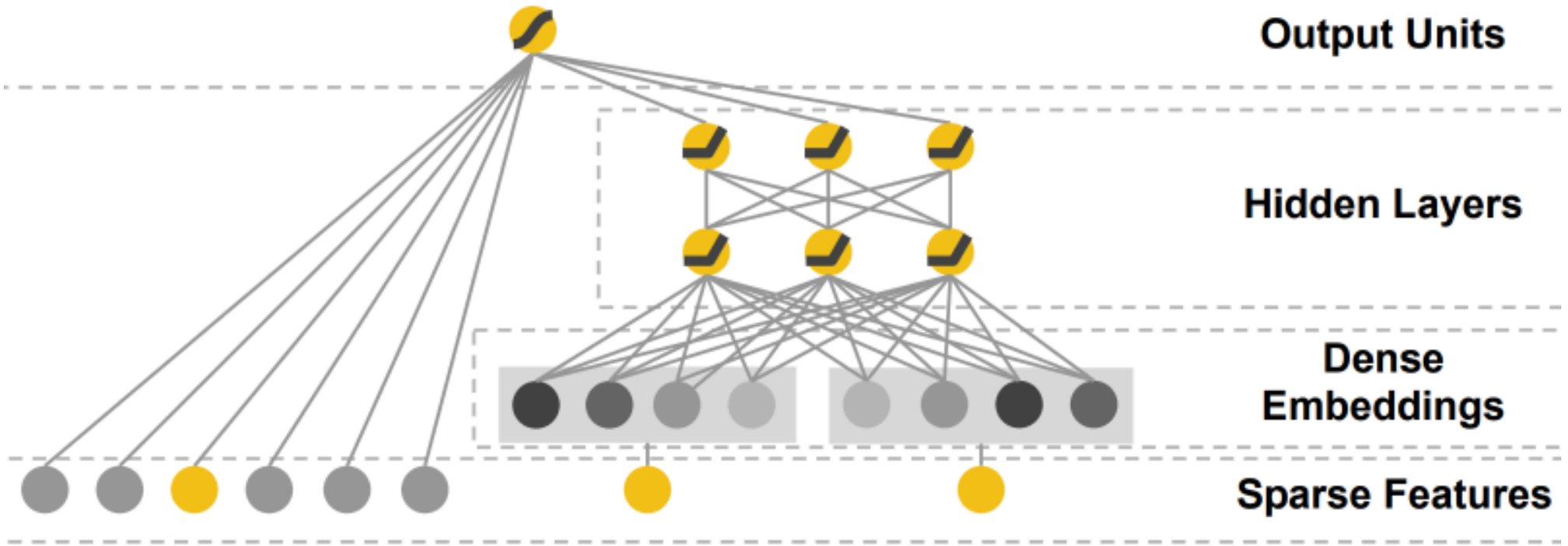
- Replace residual units in Deep Crossing with ReLu.



Wide & Deep Models

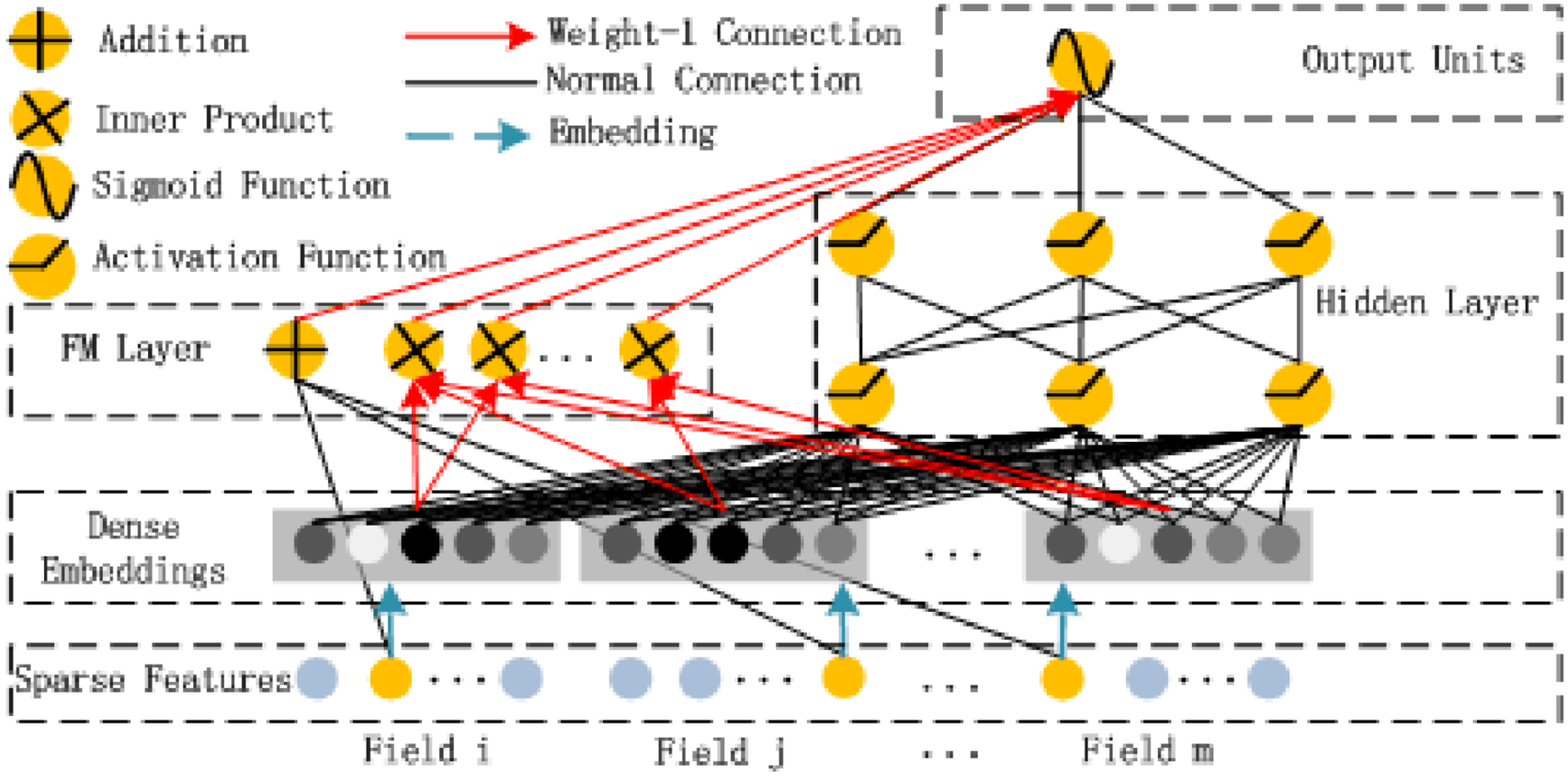
Google

- The wide part: cross features (memorization).
- The deep part: embedding layers (generalization).



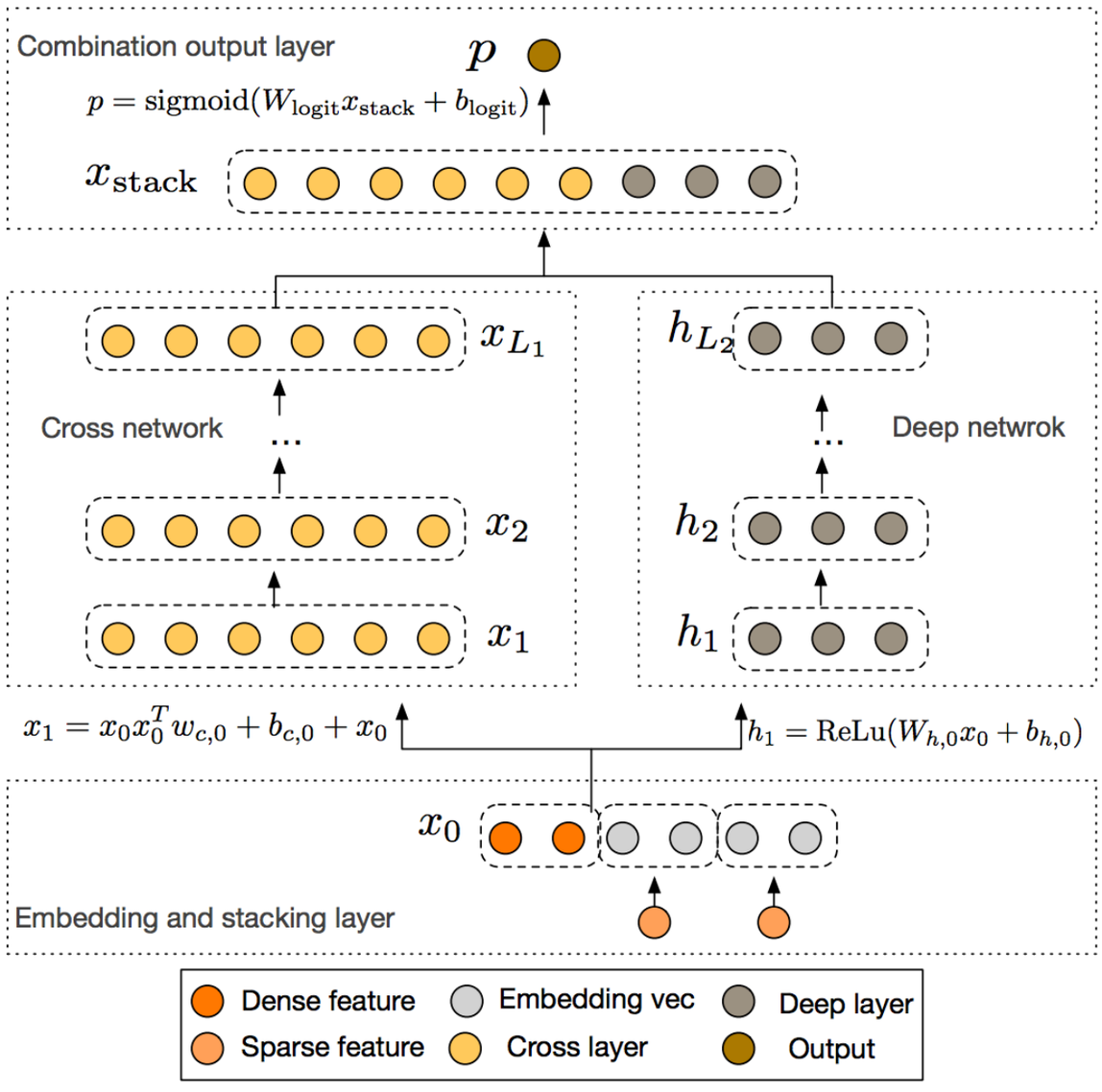
DeepFM

- Replace LR in Wide & Deep with FM.
- Share embeddings between the FM and deep part.



Deep & Cross Network (DCN)

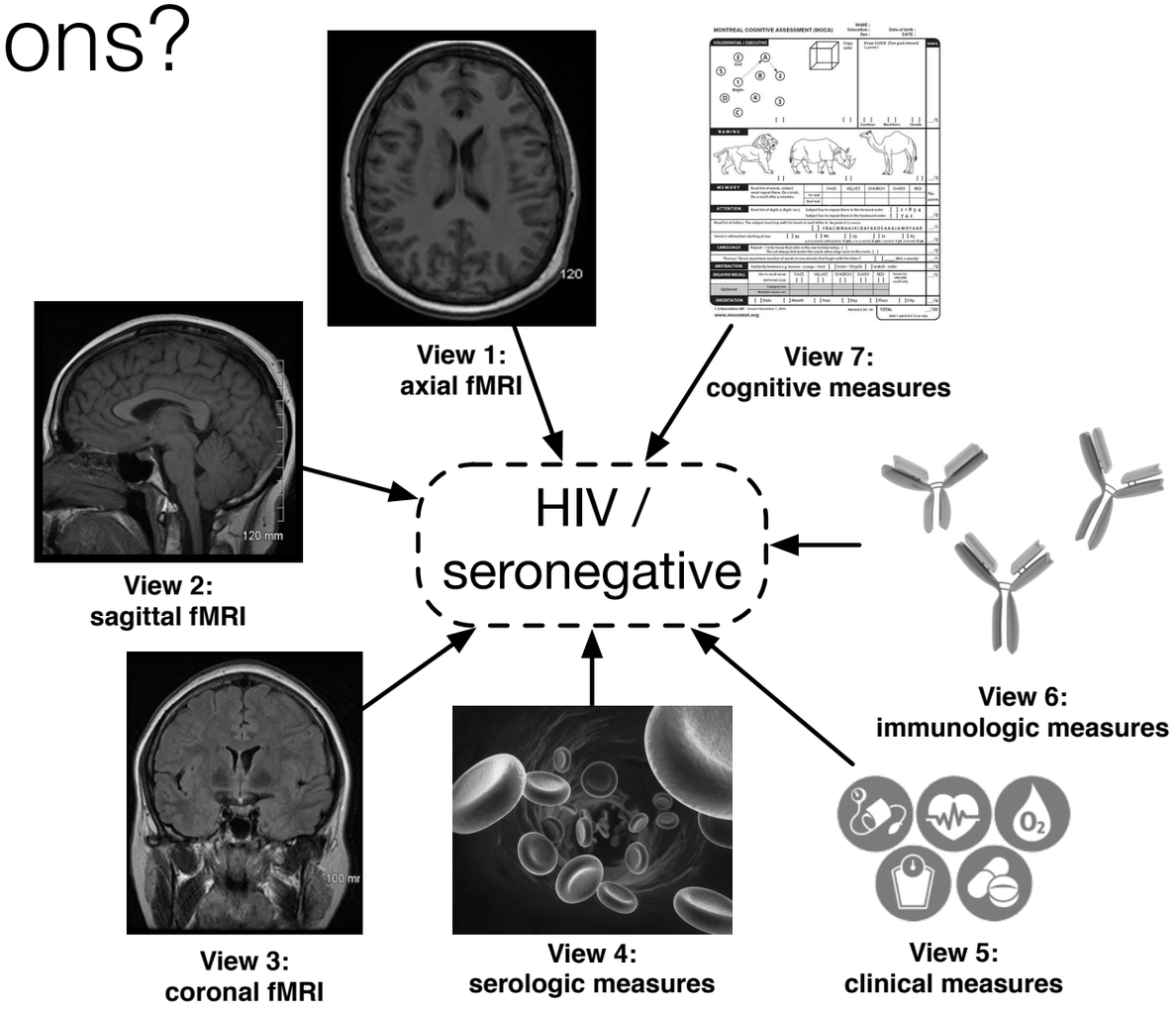
- Learn explicit cross features of bounded degree.



More on our works

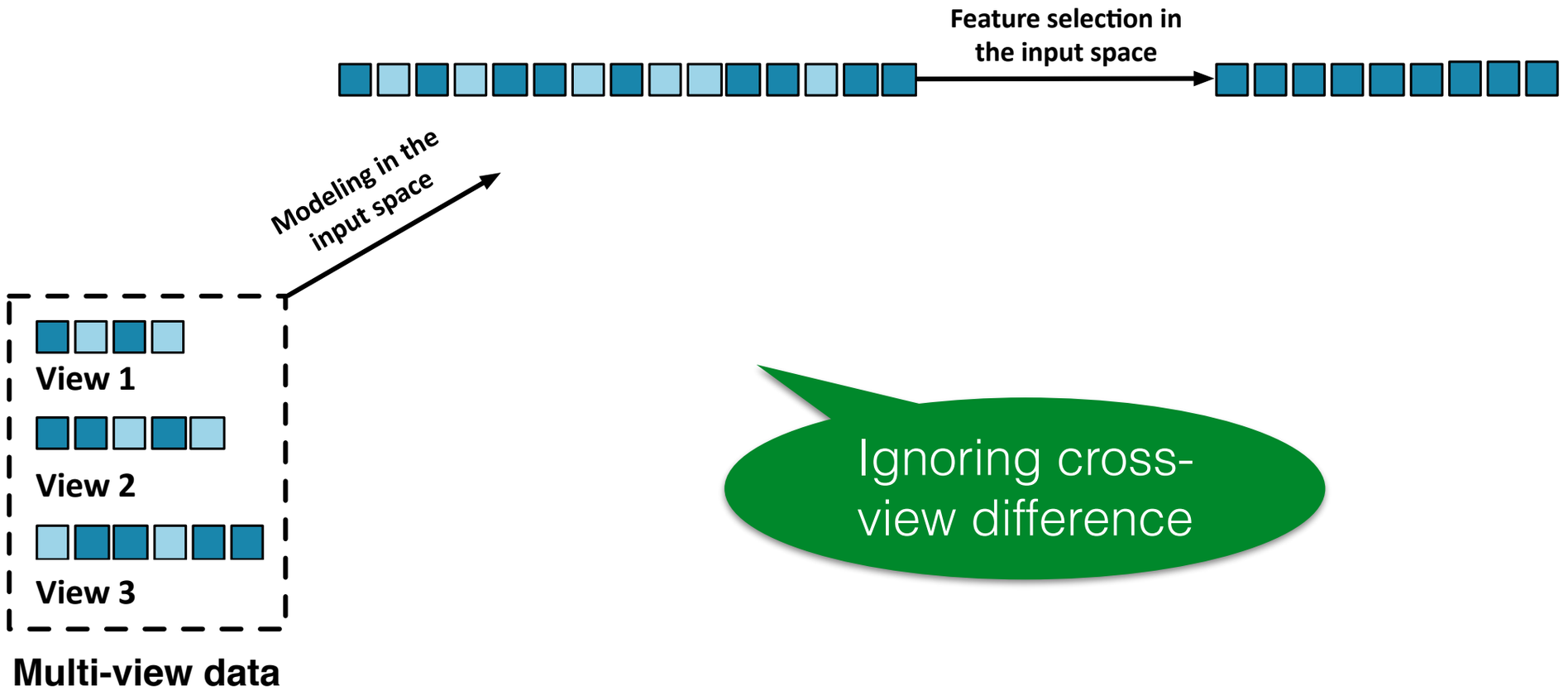
Problem Setting

- (P1) How can we select discriminative features from measures collected from different medical examinations?



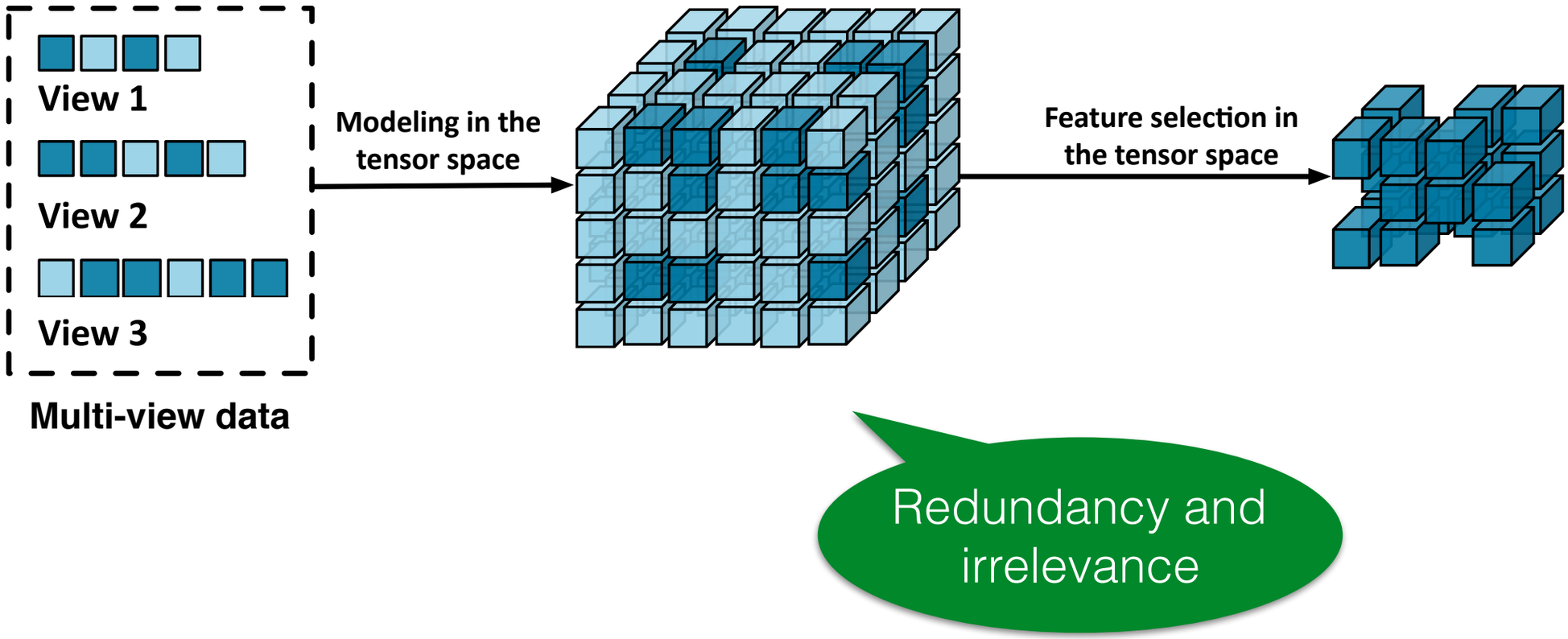
Modeling Multi-View Data

- Vector-based method.



Modeling Multi-View Data

- Tensor-based method.



Modeling Multi-View Data

- Input space.
  - Preserve the interpretability of original data.
  - Fail to explore feature interactions across different views.
- Tensor space.
  - Explore feature interactions across different views.
  - Introduce redundant and irrelevant features.

It motivates us to build a dual mapping between the ***input space*** and ***tensor space***.

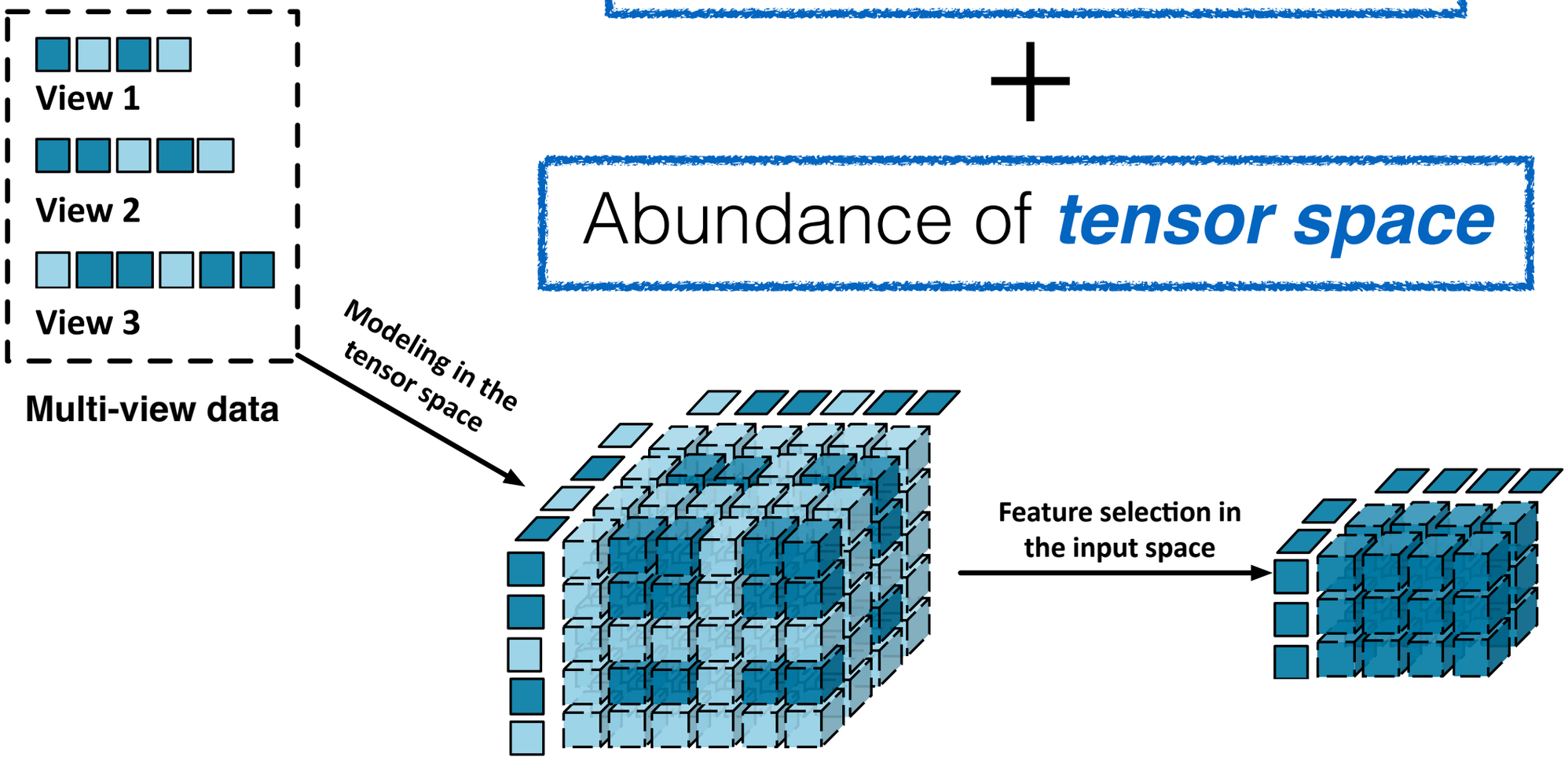
Modeling Multi-View Data

- Dual method.

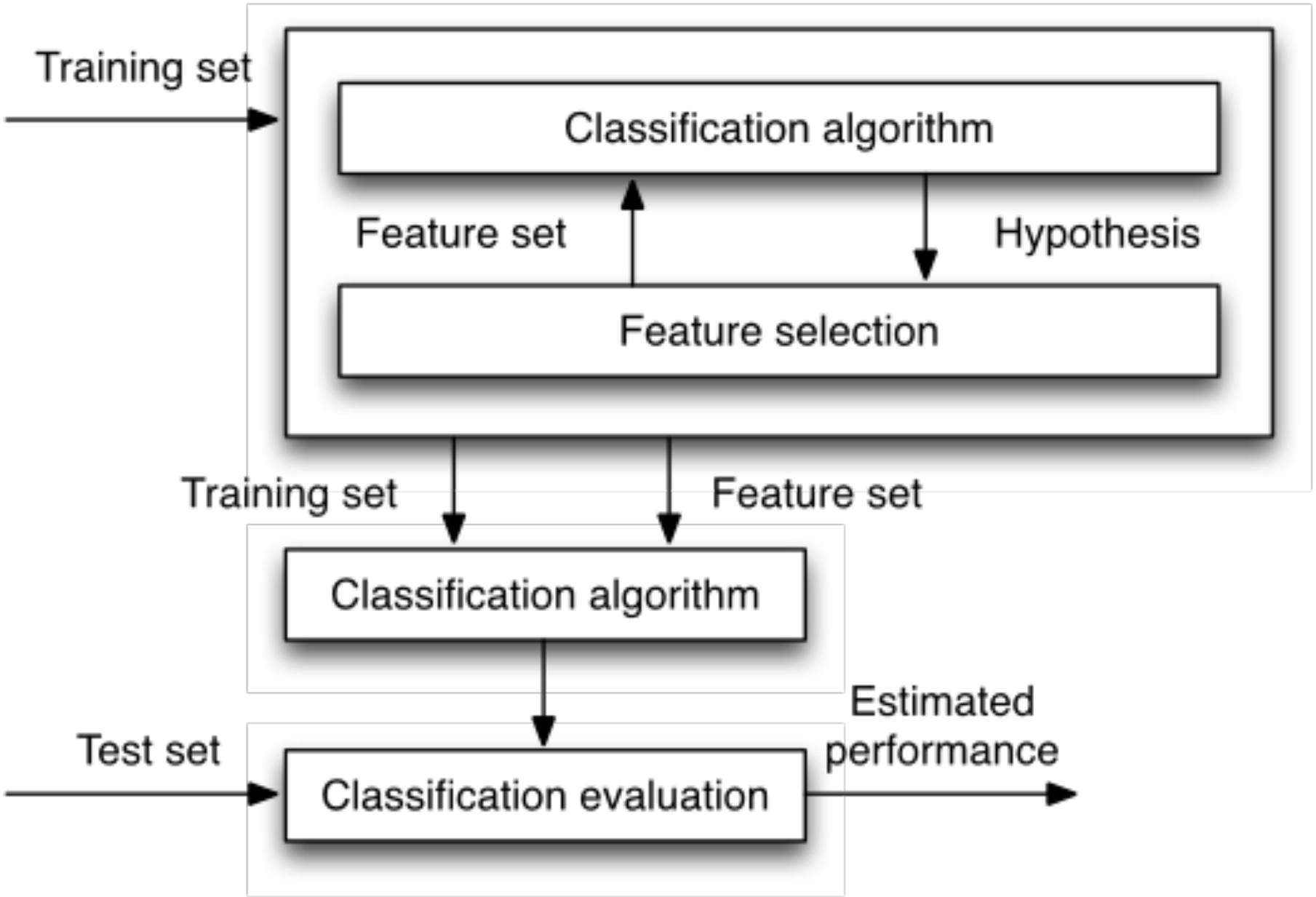
Simplicity of ***input space***

+

Abundance of ***tensor space***



Wrapper-based Feature Selection



Recursive Feature Elimination

- Feature ranking with ***Support Vector Machines***.

Train a SVM classifier

Weight vector $\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i$

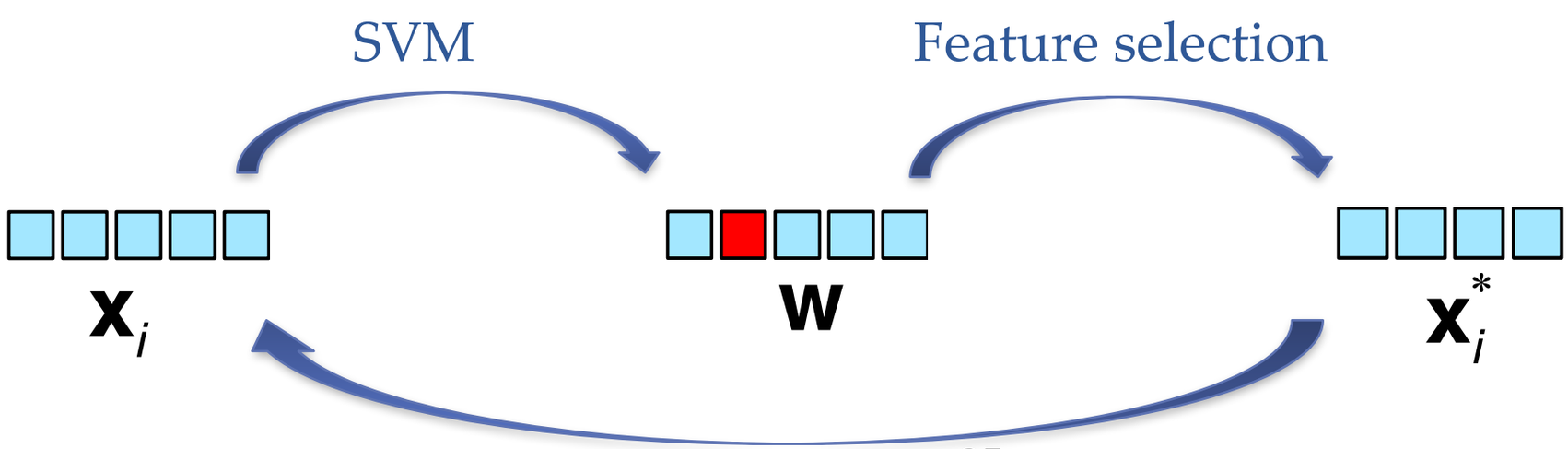
Ranking criteria $r_i = (w_i)^2$

Eliminate feature $\arg\min(\mathbf{r})$

Iteration

Guyon, Isabelle, et al. "Gene selection for cancer classification using support vector machines." Machine learning 46.1-3 (2002): 389-422.

SVM-RFE



Recursive Feature Elimination

- Feature ranking with ***Support Tensor Machines***.

Train a STM classifier $\mathcal{X} = \prod_{v=1}^{m} \circ \mathbf{x}^{(v)}$

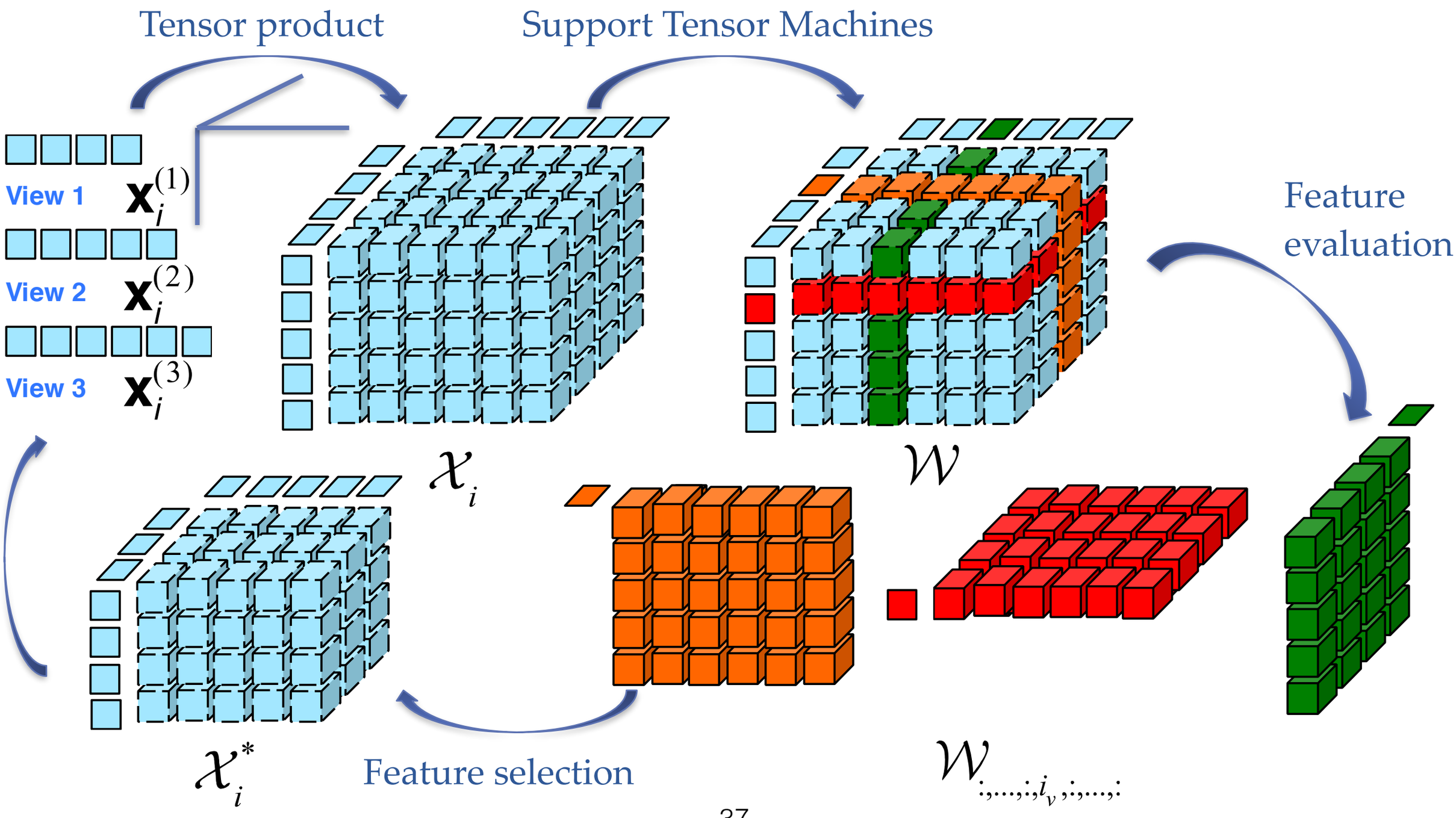
Weight tensor $\mathcal{W} = \sum_{i} \alpha_i y_i \mathcal{X}_i$

Ranking criteria $r_{i_v}^{(v)} = \sum_{i_1} \cdots \sum_{i_{v-1}} \sum_{i_{v+1}} \cdots \sum_{i_m} (w_{i_1,\cdots,i_m})^2$

Eliminate feature $\arg\min(\mathbf{r}^{(v)})$

Iteration

Tensor-based Multi-View Recursive Feature Elimination



Linear Kernel

Optimization objective

$$\min_{\mathbf{w}^{(v)},b,\xi} \frac{1}{2}\prod_{v=1}^{m}\left\|\mathbf{w}^{(v)}\right\|_F^2 + C\sum_{i=1}^{n}\xi_i$$

$$\text{s.t. } y_i\left(\prod_{v=1}^{m}\left\langle \mathbf{w}^{(v)}, \mathbf{x}_i^{(v)}\right\rangle + b\right) \geq 1-\xi_i$$

$$\xi_i \geq 0, \forall i = 1, \cdots, n.$$

$$f(\mathcal{X}) = \text{sign}\left(\prod_{v=1}^{m}\left\langle \mathbf{w}^{(v)}, \mathbf{x}^{(v)}\right\rangle + b\right)$$

$$\mathcal{W} = \prod_{v=1}^{m} \circ \mathbf{w}^{(v)}$$

Feature selection

$$\underset{i_v}{\text{argmin}}\left(r_{i_v}^{(v)}\right) \qquad r_{i_v}^{(v)} = \left(w_{i_v}^{(v)}\right)^2$$

$$r_{i_v}^{(v)} = \sum_{i_1}\cdots\sum_{i_{v-1}}\sum_{i_{v+1}}\cdots\sum_{i_m}\left(w_{i_1,\cdots,i_m}\right)^2$$

$$= \sum_{i_1}\cdots\sum_{i_{v-1}}\sum_{i_{v+1}}\cdots\sum_{i_m}\left(w_{i_1}^{(1)}\cdots w_{i_m}^{(m)}\right)^2$$

$$= \left(w_{i_v}^{(v)}\right)^2 \prod_{1\leq j\leq m}^{j\neq v}\left\|\mathbf{w}^{(j)}\right\|_F^2$$

$$= P^{(-v)}\left(w_{i_v}^{(v)}\right)^2$$

Extension to Nonlinear Kernels

Optimization objective

$$\min_{\alpha} \frac{1}{2}\alpha^{(v)T}\mathbf{H}\alpha^{(v)} - \alpha^{(v)T}\mathbf{1}$$

$$\text{s.t.} \sum_{i=1}^{n} \alpha_i^{(v)} y_i = 0$$

$$0 \le \alpha_i^{(v)} \le C, \forall i = 1, \cdots, n.$$

where $\mathbf{H}_{p,q} = y_p y_q \kappa(\mathbf{x}_p^{(v)'}, \mathbf{x}_q^{(v)'})$

$$P^{(-v)} = \prod_{1 \le j \le m}^{j \ne v} \|\mathbf{w}^{(j)}\|_F^2$$

$$Q_i^{(-v)} = \prod_{1 \le j \le m}^{j \ne v} \langle \mathbf{w}^{(j)}, \mathbf{x}_i^{(j)} \rangle$$

Feature selection

$$\underset{i_v}{\text{argmin}} \left(r_{i_v}^{(v)} \right)$$

$$r_{i_v}^{(v)} = \frac{1}{2}(\alpha^{(v)T}\mathbf{H}\alpha^{(v)} - \alpha^{(v)T}\mathbf{H}(-i_v)\alpha^{(v)})$$

where

$$\mathbf{H}_{p,q}(-i_v) = y_p y_q \kappa(\mathbf{x}_p^{(v)'}(-i_v), \mathbf{x}_q^{(v)'}(-i_v))$$

$$\mathbf{x}_i^{(v)'} = (Q_i^{(-v)}/\sqrt{P^{(-v)}})\mathbf{x}_i^{(v)}$$

Input:

- Training examples in multiple views:
  $\mathfrak{X}^{(v)} = \{\mathbf{x}_1^{(v)}, \cdots, \mathbf{x}_n^{(v)}\}, v = 1, 2, \cdots, m$
- Class labels: $\mathfrak{Y} = \{y_1, \cdots, y_n\}$
- Number of features selected in each view: p_v

Initialize:

- Subset of surviving features: $\mathbf{s}^{(v)} = [1, 2, \cdots, I_v]$

Iterate through each view v**:**

Repeat until length($\mathbf{s}^{(v)}$) $\leq p_v$

- Restrict training examples to good feature indices:
  $\mathfrak{X}^{(v)*} = \mathfrak{X}^{(v)}(\mathbf{s}^{(v)}, :)$
- Train the classifier: $\alpha = \text{SVM-TRAIN}(\mathfrak{X}^{(v)*}, \mathfrak{Y})$
- Compute the weight vector $\mathbf{w}^{(v)}$ according to Eq. (26)
- Compute the ranking criteria $\mathbf{r}^{(v)}$ according to Eq. (18)
- Find the feature with smallest ranking criterion:
  $f = \arg\min(\mathbf{r}^{(v)})$
- Eliminate the feature with smallest ranking criterion:
  $\mathbf{s}^{(v)} = \mathbf{s}^{(v)}(1{:}f\text{-}1, f{+}1{:}\text{length}(\mathbf{s}^{(v)}))$

Output:

- Subset of selected features in each view:
  $\mathbf{s}^{(v)}, v = 1, 2, \cdots, m$

The Next

- To use other loss functions.
- To include lower-order feature interactions.
- To extend to higher rank weight decomposition.

$$\mathcal{W} = \prod_{v=1}^{m} \circ \mathbf{w}^{(v)} \rightarrow \mathcal{W} = \sum_{r=1}^{R} \prod_{v=1}^{m} \circ \mathbf{w}^{(v)}$$

$$\hat{y}(x) := w_0 + \sum_{i=1}^{n} w_i \, x_i + \sum_{l=2}^{d} \sum_{i_1=1}^{n} \cdots \sum_{i_l=i_{l-1}+1}^{n} \left(\prod_{j=1}^{l} x_{i_j} \right) \left(\sum_{f=1}^{k_l} \prod_{j=1}^{l} v_{i_j,f}^{(l)} \right)$$

FM?

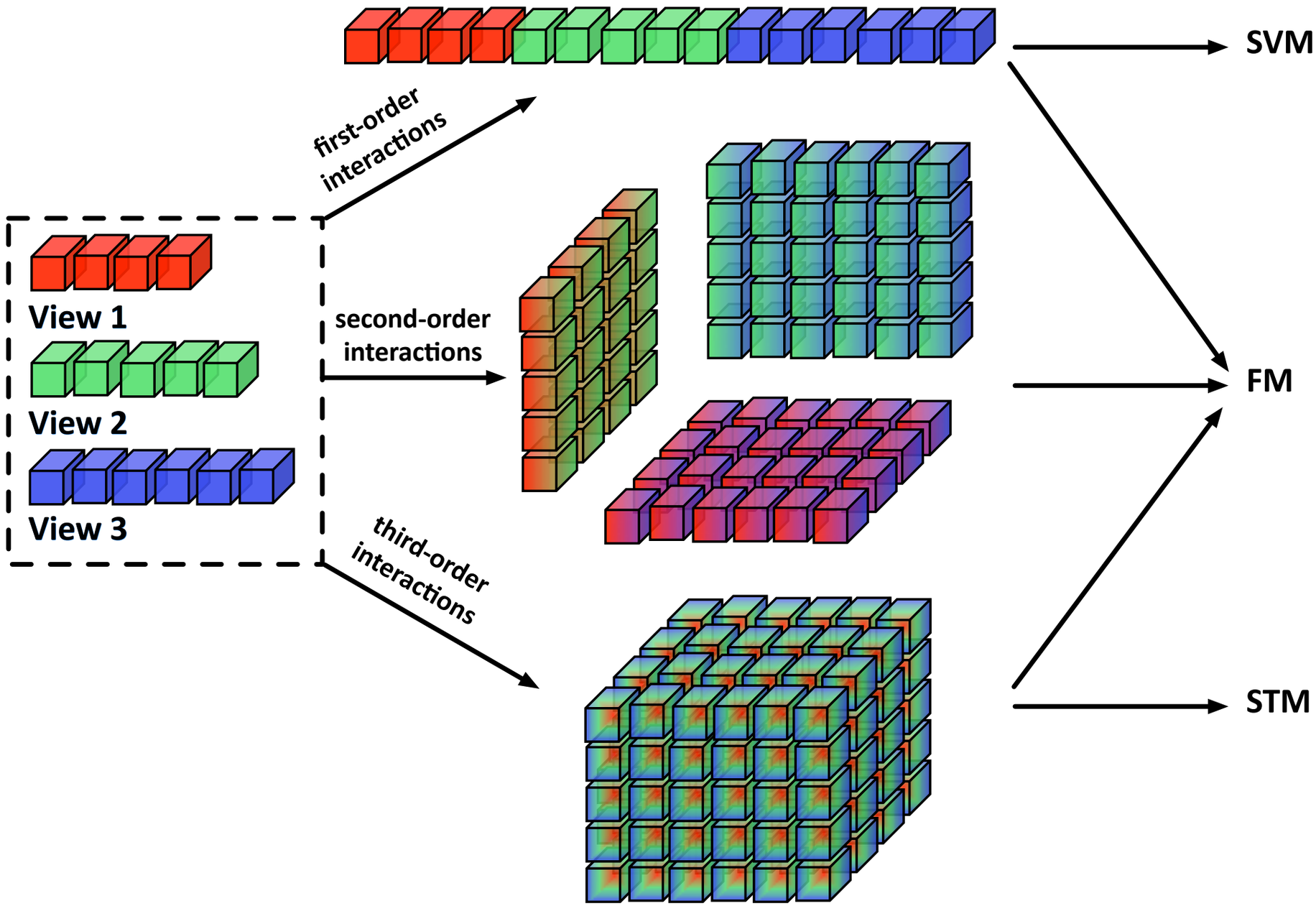
Problem Setting

- (P2) How can we predict the click through rate given a tuple (user, ad, query)?

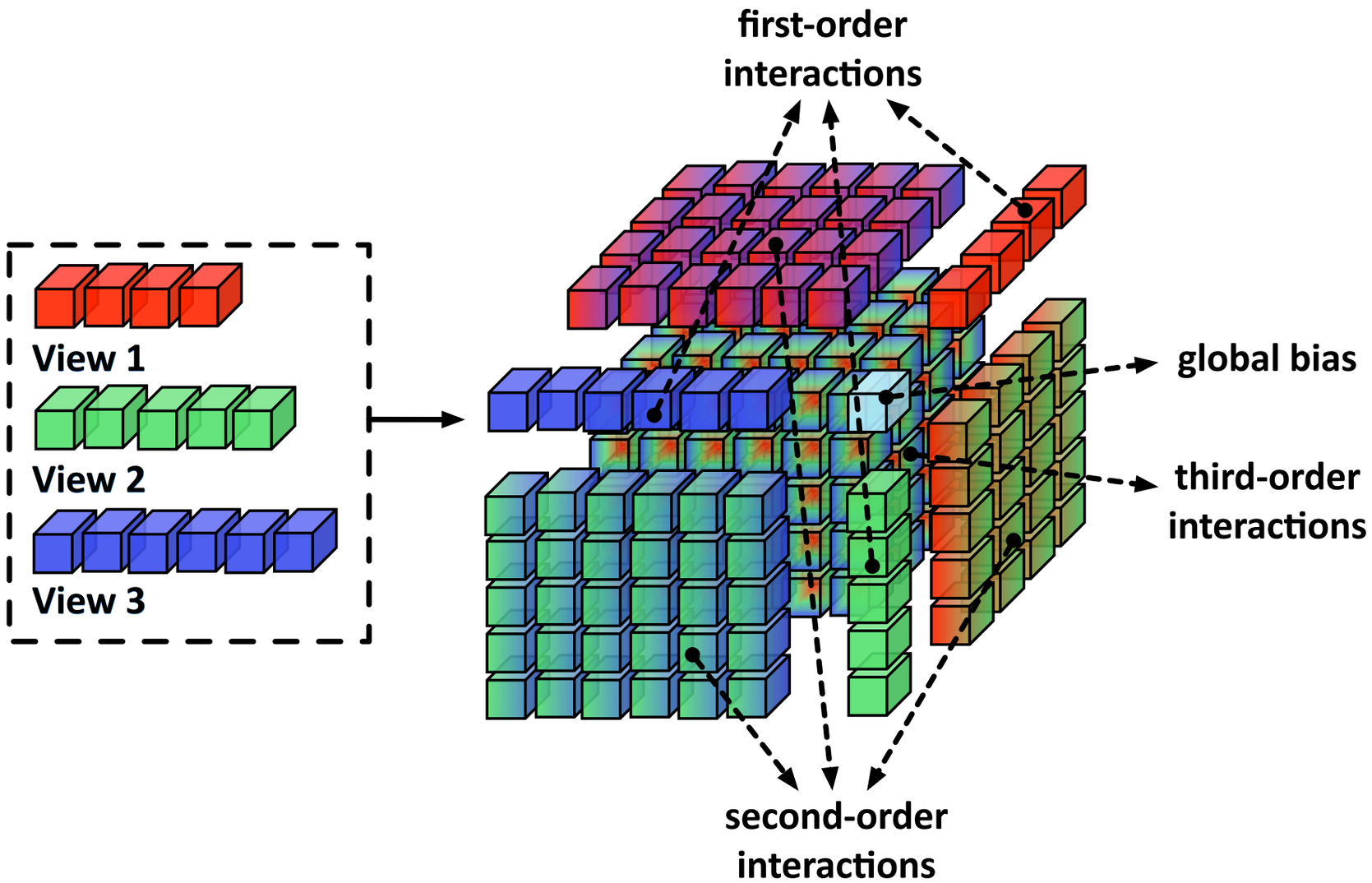
Table 1: An example showing the discrepancy between feature interactions with different orders. $\#1 = user + ad + query$, $\#2 = user \times ad + user \times query + ad \times query$, $\#3 = user \times ad \times query$.

User	Ad	Query	#1	#2	#3
1.20 (+)	1.80 (+)	0.50 (+)	3.50 (+)	3.66 (+)	1.08 (+)
1.20 (+)	1.80 (+)	-0.50 (-)	2.50 (+)	0.66 (+)	-1.08 (-)
1.20 (+)	-1.80 (-)	-0.50 (-)	-1.10 (-)	-1.86 (-)	1.08 (+)

Related Works



Multi-View Machines (MVM)



- Include the full-order interactions across multi-view features.
- Jointly factorize the interaction parameters in different orders.

$$\hat{y} = \underbrace{\beta_0}_{\text{global bias}} + \underbrace{\sum_{p=1}^{m} \sum_{i_p=1}^{I_p} \beta_{i_p}^{(p)} x_{i_p}^{(p)}}_{\text{first-order interactions}}$$

$$+ \underbrace{\sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \beta_{i_1,i_2}^{(1,2)} x_{i_i}^{(1)} x_{i_2}^{(2)} + \cdots + \sum_{i_{m-1}=1}^{I_{m-1}} \sum_{i_m=1}^{I_m} \beta_{i_{m-1},i_m}^{(m-1,m)} x_{i_{m-1}}^{(m-1)} x_{i_m}^{(m)}}_{\text{second-order interactions}}$$

$$+ \cdots + \underbrace{\sum_{i_1=1}^{I_1} \cdots \sum_{i_m=1}^{I_m} \beta_{i_1,\ldots,i_m} \left(\prod_{p=1}^{m} x_{i_p}^{(p)} \right)}_{m\text{th-order interactions}}$$

$$\hat{y} = \underbrace{\beta_0}_{\text{global bias}} + \underbrace{\sum_{p=1}^{m}\sum_{i_p=1}^{I_p}\beta_{i_p}^{(p)}x_{i_p}^{(p)}}_{\text{first-order interactions}}$$

$$+\underbrace{\sum_{i_1=1}^{I_1}\sum_{i_2=1}^{I_2}\beta_{i_1,i_2}^{(1,2)}x_{i_i}^{(1)}x_{i_2}^{(2)}+\cdots+\sum_{i_{m-1}=1}^{I_{m-1}}\sum_{i_m=1}^{I_m}\beta_{i_{m-1},i_m}^{(m-1,m)}x_{i_{m-1}}^{(m-1)}x_{i_m}^{(m)}}_{\text{second-order interactions}}$$

$$+\cdots+\underbrace{\sum_{i_1=1}^{I_1}\cdots\sum_{i_m=1}^{I_m}\beta_{i_1,\ldots,i_m}\left(\prod_{p=1}^{m}x_{i_p}^{(p)}\right)}_{m\text{th-order interactions}}$$

$$\mathbf{z}^{(p)^T} = (\mathbf{x}^{(p)^T}, 1) \in \mathbb{R}^{I_p+1}, \forall p = 1, ..., m$$

$$\hat{y} = \sum_{i_1=1}^{I_1+1}\cdots\sum_{i_m=1}^{I_m+1} w_{i_1,\ldots,i_m}\left(\prod_{p=1}^{m}z_{i_p}^{(p)}\right)$$

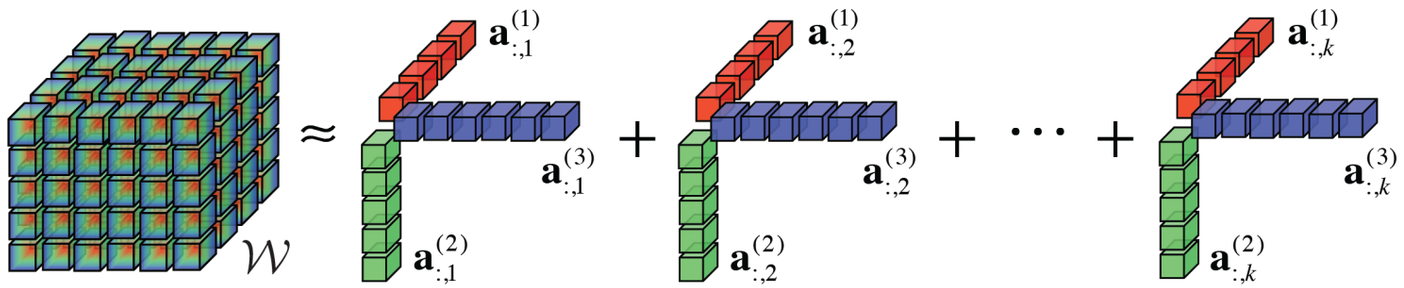
$$\hat{y} = \underbrace{\beta_0}_{\text{global bias}} + \underbrace{\sum_{p=1}^{m}\sum_{i_p=1}^{I_p}\beta_{i_p}^{(p)}x_{i_p}^{(p)}}_{\text{first-order interactions}}$$

$$+\underbrace{\sum_{i_1=1}^{I_1}\sum_{i_2=1}^{I_2}\beta_{i_1,i_2}^{(1,2)}x_{i_i}^{(1)}x_{i_2}^{(2)} + \cdots + \sum_{i_{m-1}=1}^{I_{m-1}}\sum_{i_m=1}^{I_m}\beta_{i_{m-1},i_m}^{(m-1,m)}x_{i_{m-1}}^{(m-1)}x_{i_m}^{(m)}}_{\text{second-order interactions}}$$

$$+\cdots+\underbrace{\sum_{i_1=1}^{I_1}\cdots\sum_{i_m=1}^{I_m}\beta_{i_1,\ldots,i_m}\left(\prod_{p=1}^{m}x_{i_p}^{(p)}\right)}_{m\text{th-order interactions}}$$

$$\mathbf{z}^{(p)^T} = (\mathbf{x}^{(p)^T}, 1) \in \mathbb{R}^{I_p+1}, \forall p = 1, ..., m$$

$$\hat{y} = \sum_{i_1=1}^{I_1+1}\cdots\sum_{i_m=1}^{I_m+1}w_{i_1,\ldots,i_m}\left(\prod_{p=1}^{m}z_{i_p}^{(p)}\right)$$



$$\mathcal{W} = \mathbf{C} \times_1 \mathbf{A}^{(1)} \times_2 \cdots \times_m \mathbf{A}^{(m)}$$

$$\hat{y} = \sum_{i_1=1}^{I_1+1}\cdots\sum_{i_m=1}^{I_m+1}\left(\prod_{p=1}^{m}z_{i_p}^{(p)}\right)\left(\sum_{f=1}^{k}\prod_{p=1}^{m}a_{i_p,f}^{(p)}\right)$$

Experimental Setup

- **Datasets**.
  - MovieLens: users (138,493), movies (27,278) and implicit feedback (27,278).
  - BingAds: queries (958,426), ads (1,935,510) and impressions (18).
- **Baselines**.
  - Linear regression/logistic regression (LR), tensor factorization (TF) and multi-view FMs (mvFM, mvFM-3d, mvFM-reg).
- **Implementation**.
  - AdaGrad, GraphX in Spark with iterative forward and backward steps.
  - Code available at https://goo.gl/Mi4aPx.

Experimental Result

Table 5: Prediction accuracy. ↓ indicates the smaller the value the better the performance; ↑ indicates the larger the value the better the performance.

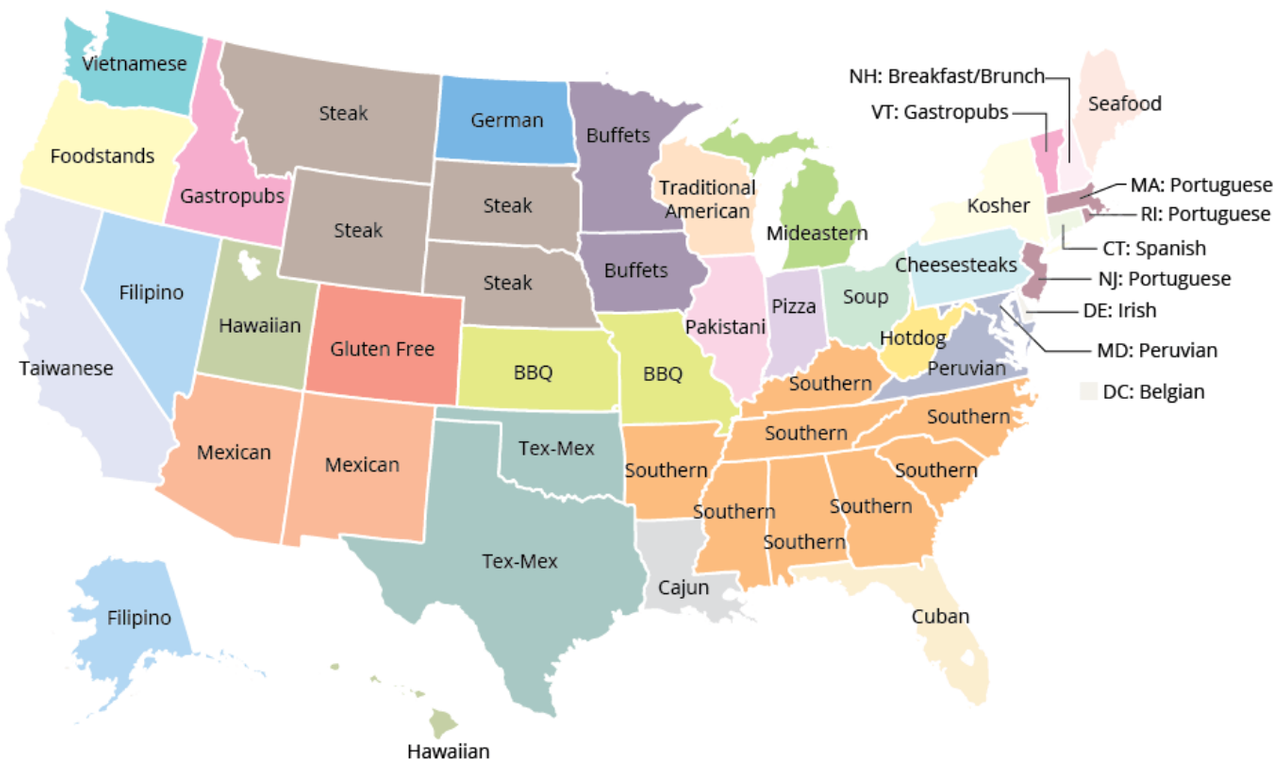
Dataset	MovieLens (RMSE) ↓	BingAds (AUC) ↑
MVM	0.8376	0.7917
FM	0.8681	0.7872
mvFM	0.8447	0.7729
mvFM-3d	0.9060	0.7201
mvFM-reg	0.9807	0.6947
TF	0.8572	0.6645
LR	1.0017	0.7450

The Next

- To explore shared feature interactions for multi-task learning.

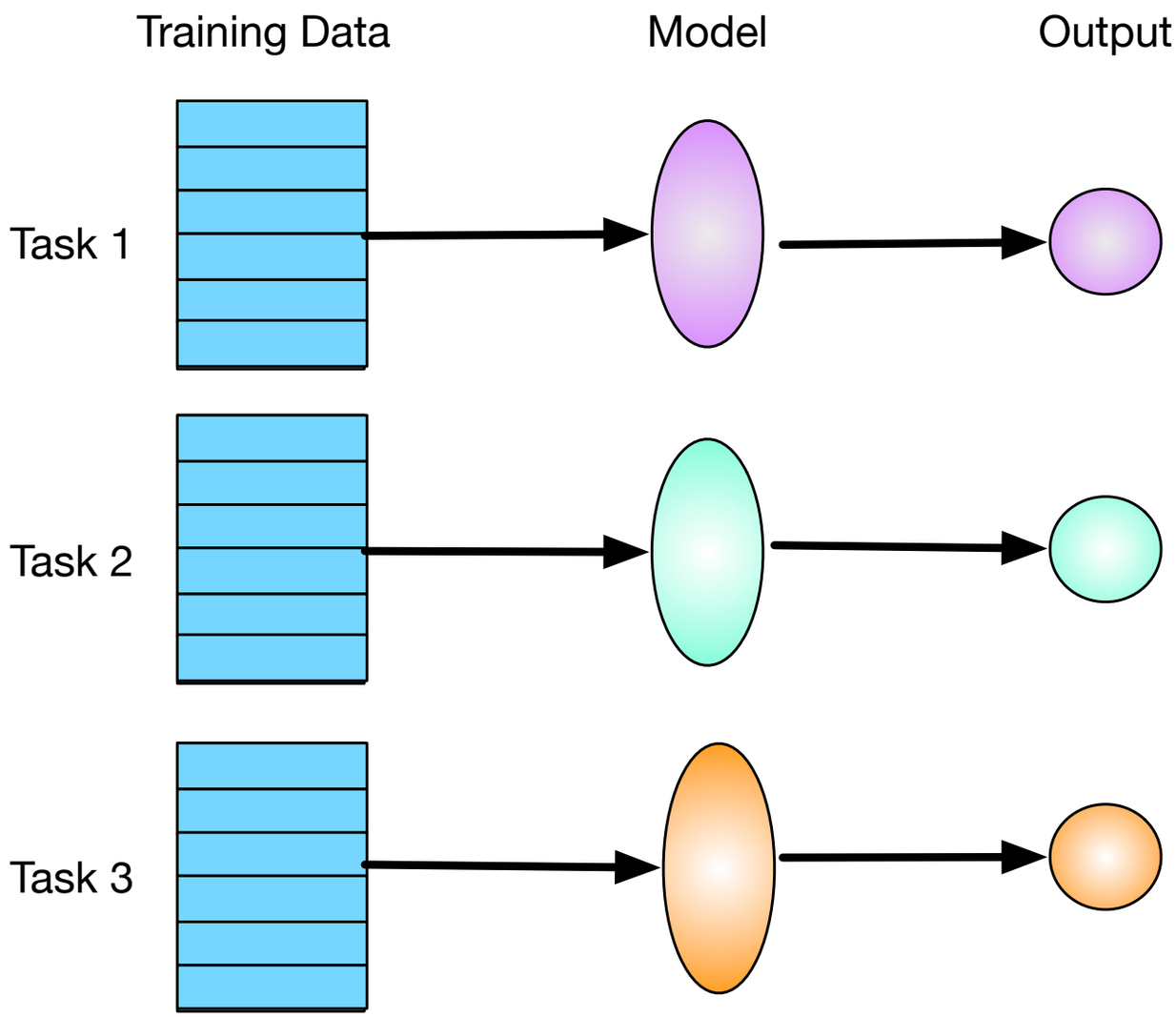
Problem Setting

- (P3) How can we predict ratings for multiple genres of movies on IMDb, or for multiple categories of items on Amazon, etc.?



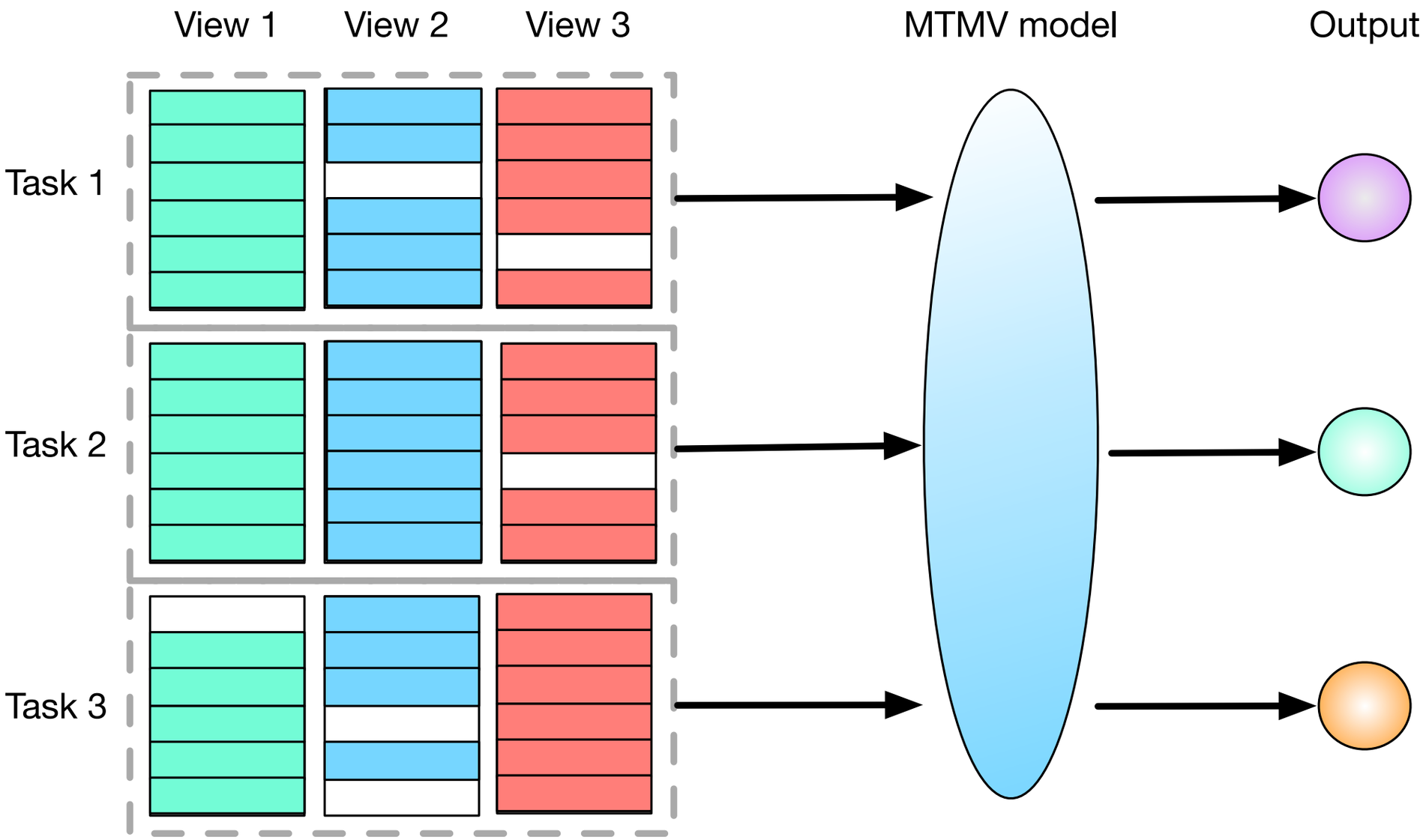
Imagine source: The Huffington Post

Single-Task Learning



- Limited data available for each task.
- Unable to learn from related tasks.

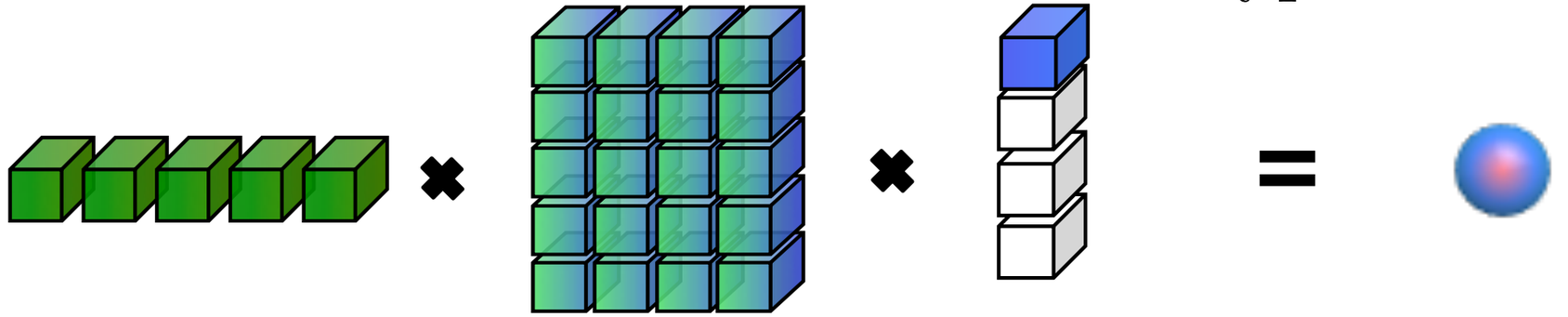
Multi-Task Multi-View Learning



Bilinear Predictive Models

- MTL w/ single view is to learn a bilinear map.

$$\mathbf{e}_t = [\underbrace{0, \cdots, 0}_{\text{t-1}}, 1, 0, \cdots, 0]^{\mathrm{T}}$$



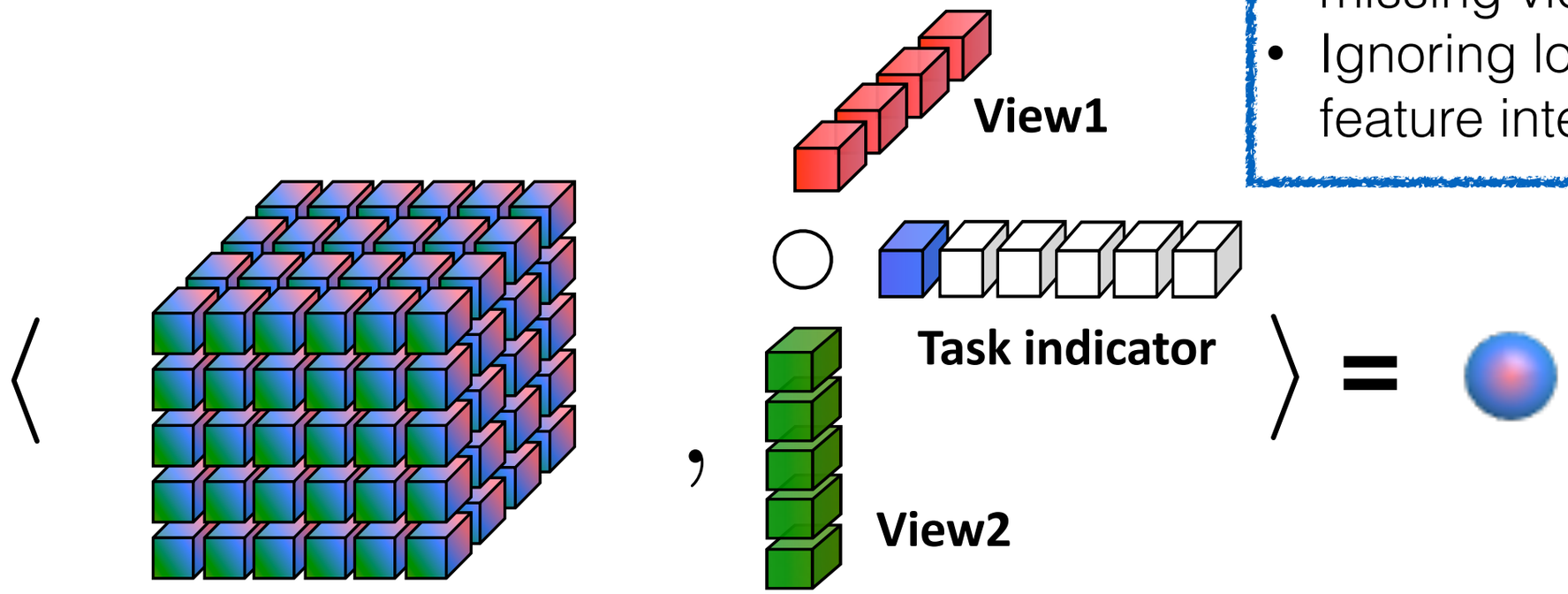
Feature vector **Weight matrix** **Task indicator** **Predicted value**

$$f_t(\mathbf{x}) = \mathbf{x}^{\mathrm{T}}\mathbf{w}_t = \mathbf{x}^{\mathrm{T}}\mathbf{W}\mathbf{e}_t = \langle \mathbf{W}, \mathbf{x} \circ \mathbf{e}_t \rangle = f(\{\mathbf{x}, \mathbf{e}_t\})$$

Multilinear Predictive Models

- MTMVL is to learn a multilinear map.

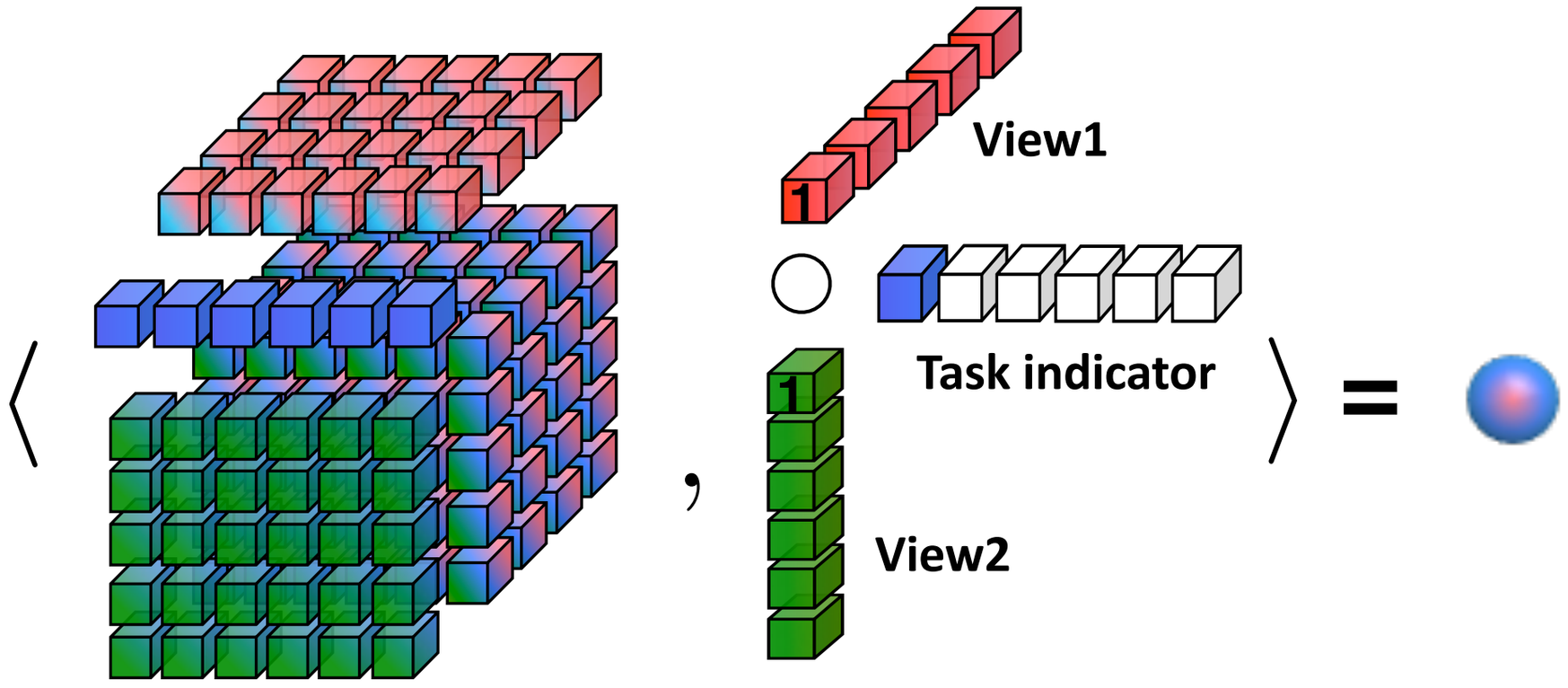
- Unable to handle missing views.
- Ignoring lower-order feature interactions.



$$f_t(\{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}\}) = \mathbf{x}^{(1)^{\mathrm{T}}} \mathbf{W}_t \mathbf{x}^{(2)} = \left\langle \mathcal{W}, \mathbf{x}^{(1)} \circ \mathbf{x}^{(2)} \circ \mathbf{e}_t \right\rangle$$

Multilinear Predictive Models

- Nest full-order feature interactions.

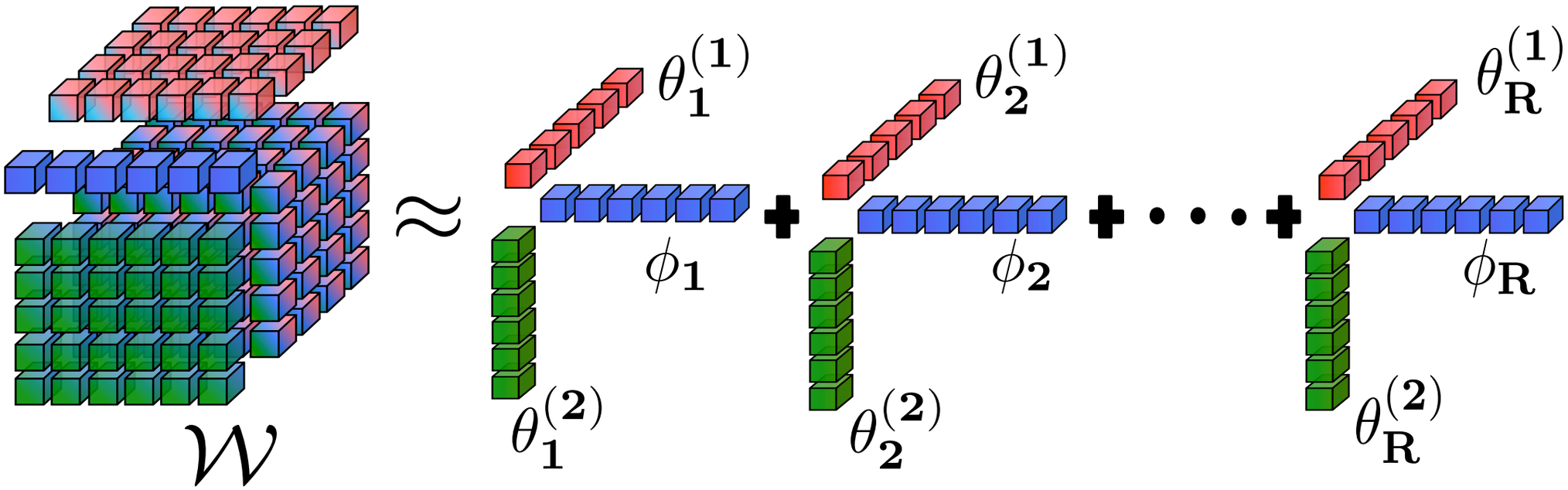


$$f_t(\{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}\}) = w_t + \sum_{v=1}^{2} \mathbf{x}^{(v)^{\mathrm{T}}} \mathbf{w}_t^{(v)} + \mathbf{x}^{(1)^{\mathrm{T}}} \mathbf{W}_t \mathbf{x}^{(2)} = \left\langle \mathcal{W}, [1; \mathbf{x}^{(1)}] \circ [1; \mathbf{x}^{(2)}] \circ \mathbf{e}_t \right\rangle$$

$$= \left\langle \mathcal{W}, \mathbf{z}^{(1)} \circ \mathbf{z}^{(2)} \circ \mathbf{e}_t \right\rangle = \langle \mathcal{W}, \mathcal{Z}_t \rangle$$

Multilinear Factorization Machines (MFM)

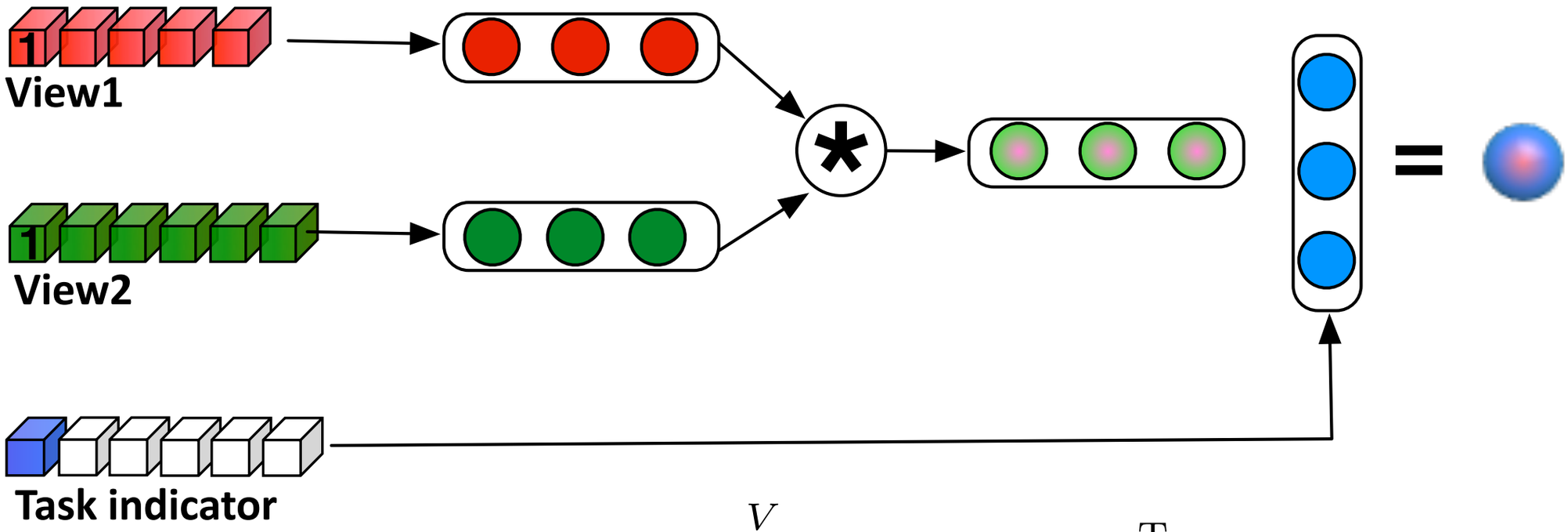
- CP factorization on the weight tensor.



$$\mathcal{W} = \sum_{r=1}^{R} \phi_{\mathbf{r}} \circ \theta_{\mathbf{r}}^{(\mathbf{1})} \circ \theta_{\mathbf{r}}^{(\mathbf{2})}$$

Multilinear Factorization Machines (MFM)

- Computational graph perspective.

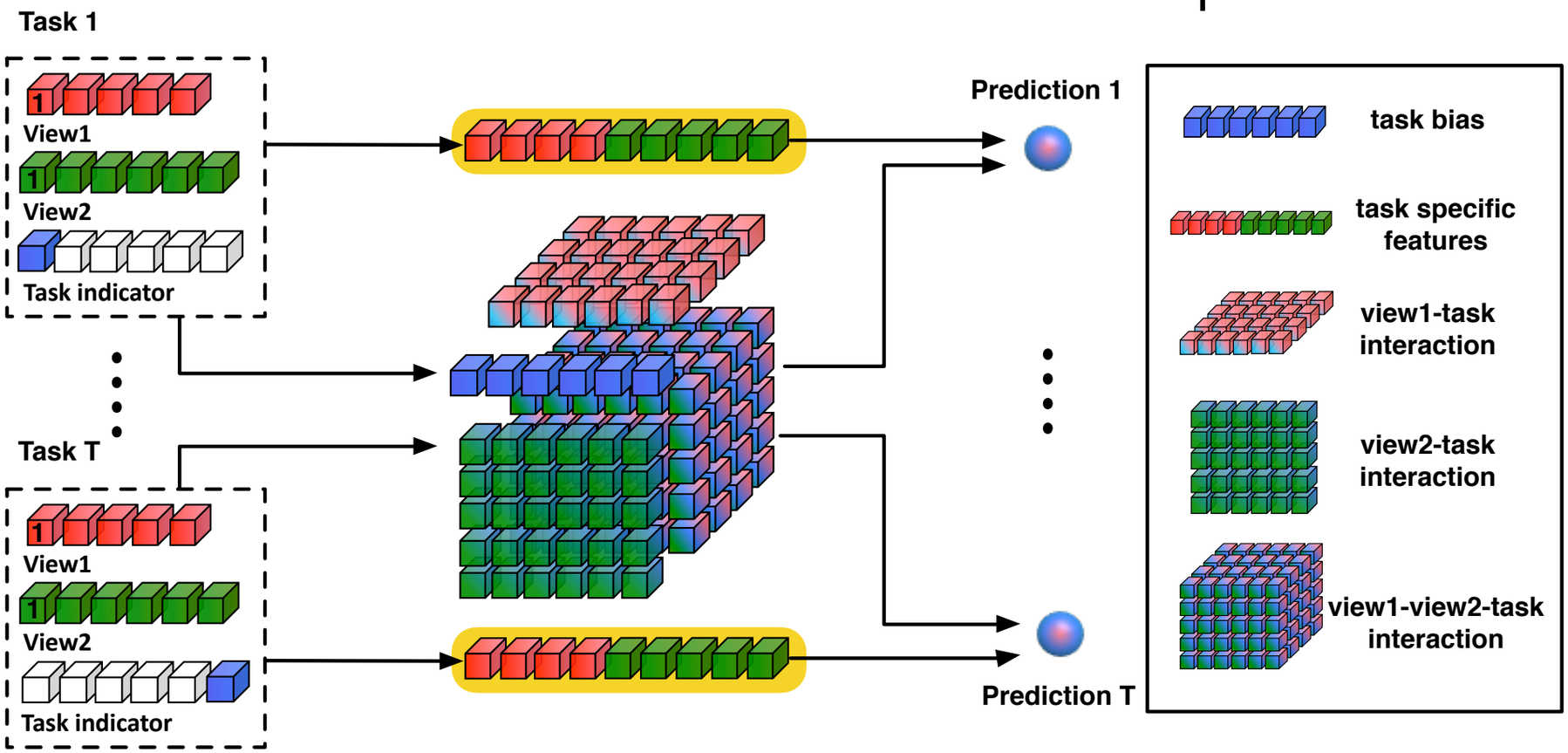


$$\langle \mathcal{W}, \mathcal{Z}_t \rangle = \boldsymbol{\phi}^t \prod_{v=1}^{V} * \left(\mathbf{z}^{(v)^{\mathrm{T}}} \boldsymbol{\Theta}^{(v)} \right)^{\mathrm{T}}$$

- Too restrictive to assume all tasks share the same subspace.

Multilinear Factorization Machines (MFM)

- Learn from both the task-specific linear space and task-view shared multilinear feature space.



$$f_t(\{\mathbf{x}^{(v)}\}) = \mathbf{x}^{\mathrm{T}}\mathbf{u}_t + \phi^t \prod_{v=1}^{V} * \left(\mathbf{z}^{(v)^{\mathrm{T}}}\mathbf{\Theta}^{(v)}\right)^{\mathrm{T}}$$

Data Collection

- FOX: multi-class w/ image and text features.
- DBLP: multi-label w/ textual and linkage features.
- MovieLens: regression w/ users, items and tags.
- Amazon: large-scale regression w/ users, items and text.

Classification	#Feature	T	N_p	N_n
FOX	image(996), text(2,711)	4	178∼635	888∼1,345
DBLP	linkage(4,638), text(687)	6	635∼1,950	2,688∼3,985
Regression	**#Feature**	T	N	**Density**
MovieLens	users(943), movies(1,599), tags(1,065)	10	758∼39,895	6.3%
Amazon	users(1,805,364), items(192,978), text(83,143)	5	349,038∼1,015,189	0.001%

- Report the average results of 10 times of random sampling: n% labeled instances as training set (n=10,20,30), 10% as validation set and 40% as testing set.

Compared Methods

- **rMTFL**: robust multi-task feature learning algorithm.
- **IteM^2**: transductive MTMV classification algorithm.
- **CSL-MTMV**: state-of-the-art inductive MTMV learning algorithm.
- **Factorization Machine (FM)**: state-of-the-art factorization model.
- **Tensor Factorization (TF)**: factorizing the highest-order weight tensor.
- **Multilinear Tensor Factorization (MFM)**:
  - **MFM-T**: removing the task-specific linear space.
  - **MFM-F**: using F-norm regularizers for all parameters.
  - **MFM-F-S**: using L2,1-norm on U for joint feature selection, F-norm on the rest of parameters.

Classification on FOX Dataset

Training Ratio	Measure	rMTFL	FM	TF	ItеM^2
10%	ACC	0.8816±0.011	0.7883±0.011	0.8460±0.035	0.4052±0.076
	F1	0.6911±0.035	0.2930±0.046	0.6362±0.044	0.3598±0.030
	AUC	0.9109±0.013	0.7764±0.018	0.8681±0.038	0.5326±0.036
20%	ACC	0.9039±0.013	0.8087±0.011	0.8546±0.025	0.5091±0.078
	F1	0.7654±0.026	0.3764±0.050	0.6632±0.051	0.3306±0.068
	AUC	0.9353±0.016	0.8260±0.012	0.8751±0.029	0.4954±0.043
30%	ACC	0.9314±0.005	0.8255±0.007	0.8767±0.082	0.4289±0.134
	F1	0.8051±0.015	0.4448±0.026	0.7302±0.132	0.3314±0.056
	AUC	0.9709±0.005	0.8393±0.012	0.9010±0.091	0.5365±0.039

Training Ratio	Measure	CSL-MTMV	MFM-T	MFM-F	MFM-F-S
10%	ACC	0.8986±0.011	0.9259±0.019	**0.9343±0.012**	**0.9364±0.011**
	F1	0.7335±0.029	0.7799±0.053	**0.8076±0.038**	**0.8119±0.027**
	AUC	0.9342±0.011	0.9678±0.015	**0.9763±0.008**	**0.9777±0.009**
20%	ACC	0.9264±0.005	0.9551±0.005	**0.9569±0.010**	**0.9612±0.005**
	F1	0.8004±0.012	0.8721±0.012	**0.8769±0.027**	**0.8882±0.014**
	AUC	0.9705±0.003	0.9883±0.003	**0.9885±0.006**	**0.9922±0.002**
30%	ACC	0.9390±0.004	0.9641±0.007	**0.9709±0.003**	**0.9697±0.004**
	F1	0.8341±0.012	0.9000±0.018	**0.9185±0.010**	**0.9149±0.010**
	AUC	0.9812±0.003	0.9916±0.003	**0.9949±0.001**	**0.9949±0.001**

Classification on DBLP Dataset

Training Ratio	Measure	rMTFL	FM	TF	IteM^2
10%	ACC	0.8057±0.004	0.7264±0.004	0.7471±0.011	0.6223±0.004
	F1	0.5395±0.015	0.0732±0.019	0.5606±0.011	0.3176±0.007
	AUC	0.7888±0.007	0.6264±0.023	0.7723±0.009	0.5310±0.007
20%	ACC	0.8319±0.004	0.7628±0.007	0.7878±0.007	0.6309±0.003
	F1	0.6447±0.008	0.2680±0.038	0.6247±0.014	0.3494±0.006
	AUC	0.8374±0.005	0.7548±0.022	0.8200±0.010	0.5550±0.006
30%	ACC	0.8412±0.004	0.7978±0.005	0.8191±0.008	0.6256±0.003
	F1	0.6796±0.010	0.4312±0.021	0.6670±0.021	0.3569±0.009
	AUC	0.8590±0.005	0.8351±0.010	0.8498±0.009	0.5563±0.006

Training Ratio	Measure	CSL-MTMV	MFM-T	MFM-F	MFM-F-S
10%	ACC	0.7290±0.005	0.8008±0.004	**0.8058±0.004**	**0.8062±0.005**
	F1	0.4402±0.004	0.5278±0.018	**0.5469±0.014**	**0.5471±0.015**
	AUC	0.6890±0.006	0.8039±0.010	**0.8113±0.010**	**0.8120±0.009**
20%	ACC	0.7760±0.002	0.8346±0.004	**0.8374±0.004**	**0.8371±0.004**
	F1	0.5295±0.007	0.6274±0.013	**0.6499±0.012**	**0.6508±0.012**
	AUC	0.7655±0.005	0.8531±0.006	**0.8658±0.005**	**0.8632±0.005**
30%	ACC	0.8037±0.003	0.8501±0.004	**0.8527±0.004**	**0.8535±0.004**
	F1	0.5869±0.007	0.6800±0.013	**0.6891±0.012**	**0.6892±0.009**
	AUC	0.8083±0.006	0.8757±0.005	**0.8866±0.006**	**0.8866±0.006**

Regression on MovieLens Dataset

Training Ratio	Measure	rMTFL	FM	TF	CSL-MTMV
10%	RMSE	1.1861±0.008	1.0251±0.003	1.5679±0.099	1.05013±0.005
	MAE	0.8516±0.004	0.8422±0.004	1.2497±0.088	0.8516±0.004
20%	RMSE	1.0631±0.005	0.9898±0.003	1.2519±0.069	1.0214±0.004
	MAE	0.8539±0.005	0.7997±0.004	0.9801±0.053	0.8294±0.004
30%	RMSE	0.9917±0.003	**0.9765±0.003**	1.2066±0.061	1.0082±0.003
	MAE	0.8159±0.003	**0.7815±0.003**	0.9380±0.045	0.8189±0.003

Training Ratio	Measure	MFM-T	MFM-F	MFM-F-S
10%	RMSE	1.0078±0.005	**1.0069±0.005**	**0.9976±0.004**
	MAE	0.8142±0.005	**0.8082±0.005**	**0.8022±0.004**
20%	RMSE	**0.9877±0.003**	0.9977±0.003	**0.9857±0.003**
	MAE	0.**7987±0.003**	0.8023±0.003	**0.7927±0.004**
30%	RMSE	0.9795±0.003	0.9887±0.004	**0.9785±0.003**
	MAE	0.7885±0.002	0.7823±0.004	**0.7789±0.004**

Regression on Amazon Dataset

Training Ratio	Measure	FM	TF	MFM-T	MFM-F	MFM-F-S
10%	RMSE	0.9834±0.001	3.6044±0.003	**0.9775±0.001**	0.9857±0.001	**0.9825±0.002**
	MAE	0.7420±0.001	3.4574±0.005	0.7249±0.001	**0.7158±0.002**	**0.7129±0.001**
20%	RMSE	0.9814±0.001	3.5611±0.018	**0.9764±0.001**	0.9845±0.001	**0.9775±0.001**
	MAE	0.7343±0.002	3.3965±0.030	0.7255±0.001	**0.7112±0.001**	**0.7086±0.001**
30%	RMSE	0.9782±0.002	3.4962±0.018	**0.9705±0.002**	0.9841±0.001	**0.9733±0.001**
	MAE	0.7257±0.002	3.2945±0.034	**0.7001±0.001**	0.7115±0.001	**0.7078±0.001**

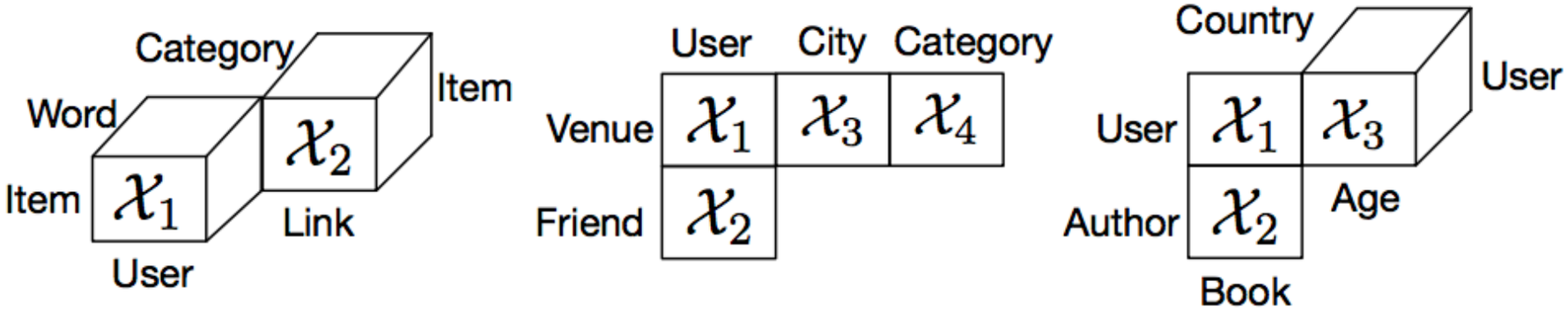
- Due to memory overhead, rMTFL and CSL-MTMV are not compared.

The Next

- To extend to overlapping groups of views.

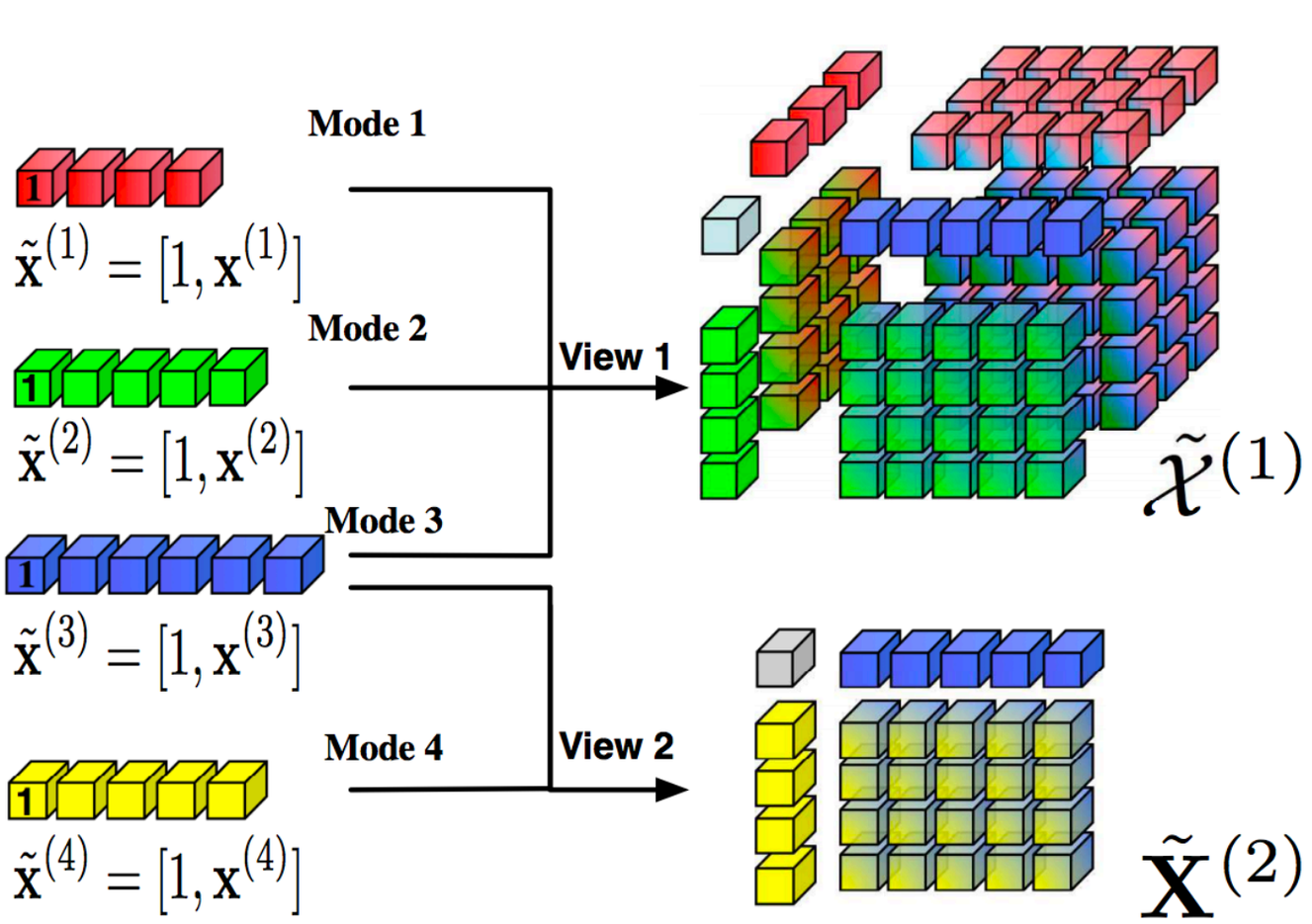
Problem Setting

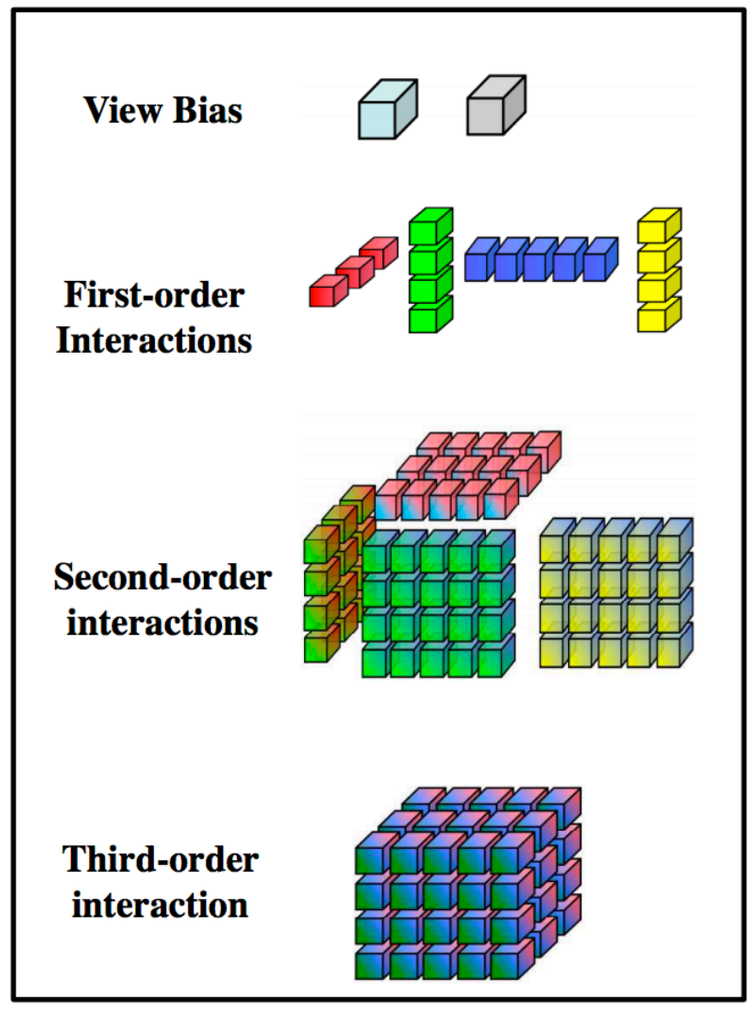
- (P4) How can we predict ratings between entities with rich meta information?



(a) Amazon (b) Yelp (c) BookCrossing

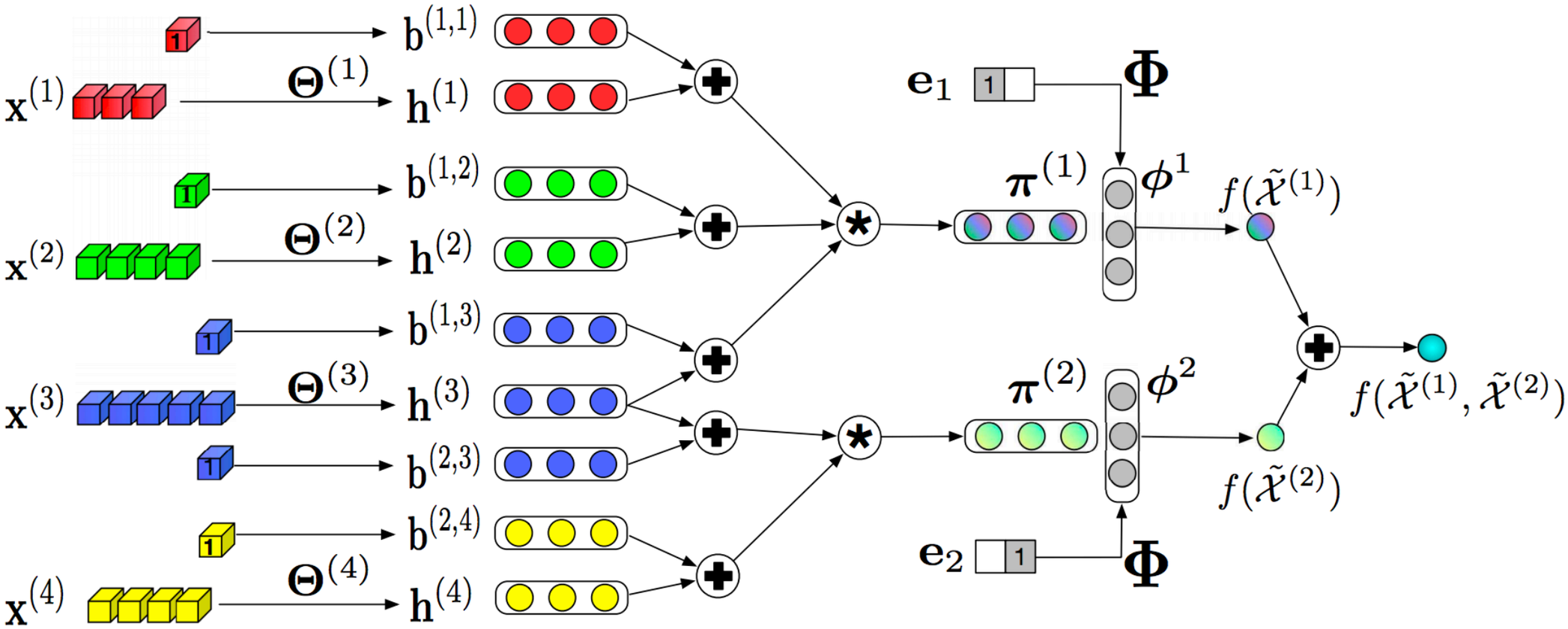
Structural Factorization Machines (SFM)





Structural Factorization Machines (SFM)

- Computational graph perspective.



Experiments

Dataset	#Samples	Mode					Density	$N_z(X)$	$N_z(\mathcal{B})$
Amazon		#Users	#Items	#Words	#Categories	#Links			
Game	231,780	24,303	10,672	7,500	193	17,974	0.089%	32.9M	15.2M
Cloth	278,677	39,387	23,033	3,493	1,175	107,139	0.031%	25.6M	7.3M
Sport	296,337	35,598	18,357	5,202	1,432	73,040	0.045%	34.2M	10.2M
Health	346,355	38,609	18,534	5,889	849	80,379	0.048%	33.6M	12.1M
Home	551,682	66,569	28,237	6,455	970	99,090	0.029%	46.8M	19.4M
Elec	1,689,188	192,403	63,001	12,805	967	89,259	0.014%	161.5M	69M
		#Users	#Venues	#Friends	#Categories	#Cities			
Yelp	1,319,870	88,009	40,520	88,009	892	412	0.037%	70.5M	1.4M
		#Users	#Books	#Countries	#Ages	#Authors			
BX	244,848	24,325	45,074	57	8	17,178	0.022%	1.2M	163K

Dataset	(a) **MF**	(b) **MVM**	(c) **FM-2**	(d) **FM-3**	(e) **PolyNet-2**	(f) **PolyNet-3**	(g) **SFM**	Improvement of SFM verus b	min(c,d)	min(e,f)
Game	1.569 ± 0.005	0.753 ± 0.007	0.764 ± 0.006	0.749 ± 0.007	0.749 ± 0.004	0.748 ± 0.006	**0.723 ± 0.006**	4.06%	3.52%	3.35%
Cloth	1.624 ± 0.009	0.725 ± 0.046	0.678 ± 0.004	0.679 ± 0.004	0.678 ± 0.007	0.680 ± 0.005	**0.659 ± 0.013**	9.03%	2.82%	2.84%
Sport	1.290 ± 0.004	0.646 ± 0.019	0.638 ± 0.003	0.632 ± 0.007	0.631 ± 0.005	0.632 ± 0.005	**0.614 ± 0.011**	5.00%	2.91%	2.79%
Health	1.568 ± 0.007	0.807 ± 0.012	0.779 ± 0.004	0.778 ± 0.004	0.779 ± 0.005	0.776 ± 0.005	**0.763 ± 0.019**	5.47%	2.02%	1.77%
Home	1.591 ± 0.004	0.729 ± 0.067	0.714 ± 0.002	0.714 ± 0.004	0.690 ± 0.003	0.692 ± 0.005	**0.678 ± 0.008**	6.93%	5.00%	1.72%
Elec	1.756 ± 0.002	0.792 ± 0.042	0.776 ± 0.006	0.749 ± 0.007	0.760 ± 0.004	0.757 ± 0.001	**0.747 ± 0.006**	5.69%	0.27%	1.33%
Yelp	1.713 ± 0.003	1.2575 ± 0.013	1.277 ± 0.002	1.277 ± 0.002	1.272 ± 0.002	1.272 ± 0.002	**1.256 ± 0.010**	0.09%	1.58%	1.19%
BX	4.094 ± 0.025	2.844 ± 0.024	2.766 ± 0.012	2.767 ± 0.014	2.654 ± 0.013	2.658 ± 0.013	**2.541 ± 0.025**	10.66%	8.16%	4.27%
Average on all datasets								5.87%	3.29%	2.41%

The Next

- To handle structured input.
- To explicitly model feature interactions in a deep framework.

Problem Setting

- (P5) How can we monitor the mental health using time series data collected from multiple sensors?

Accelerometer

Keyboard

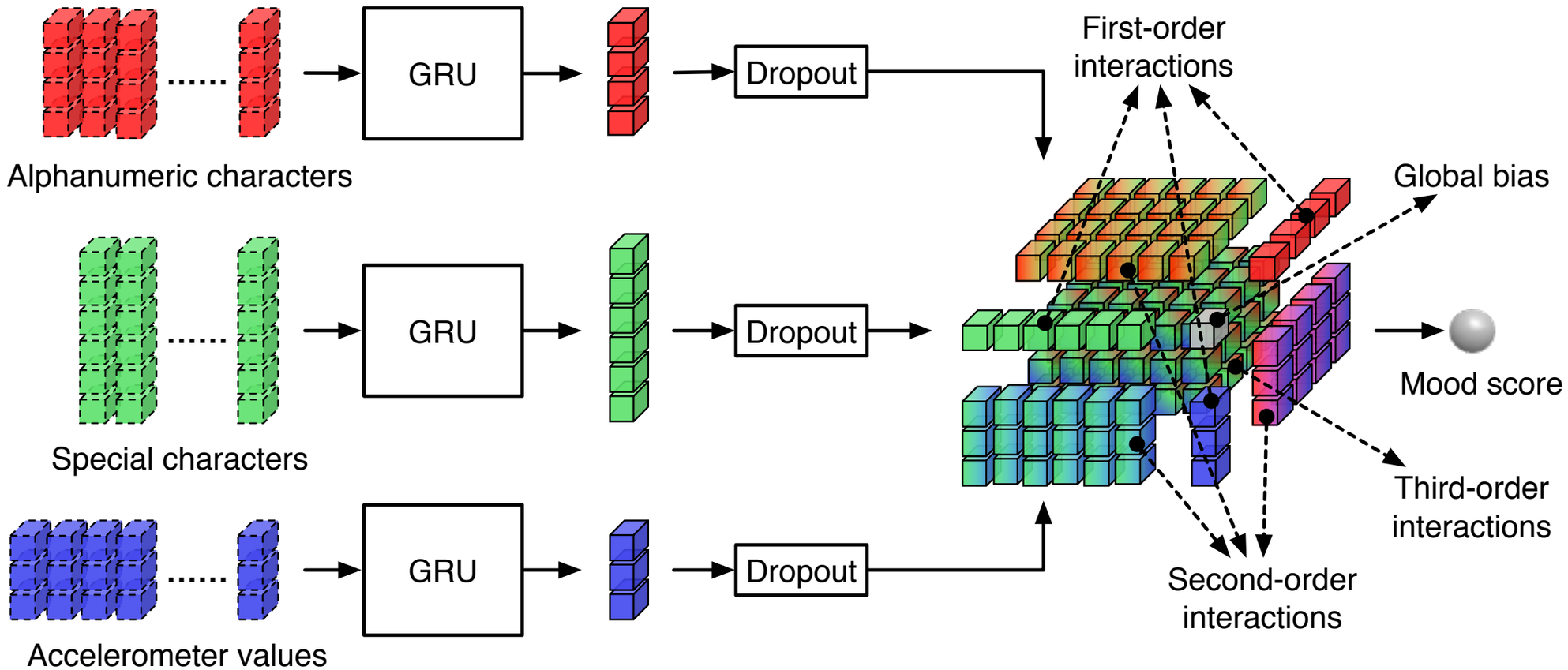
😀 Mood detection 😭

Gyroscope

GPS

Microphone

DeepMood


Alphanumeric characters
GRU
Dropout
Special characters
GRU
Dropout
Accelerometer values
GRU
Dropout
First-order interactions
Global bias
Mood score
Third-order interactions
Second-order interactions

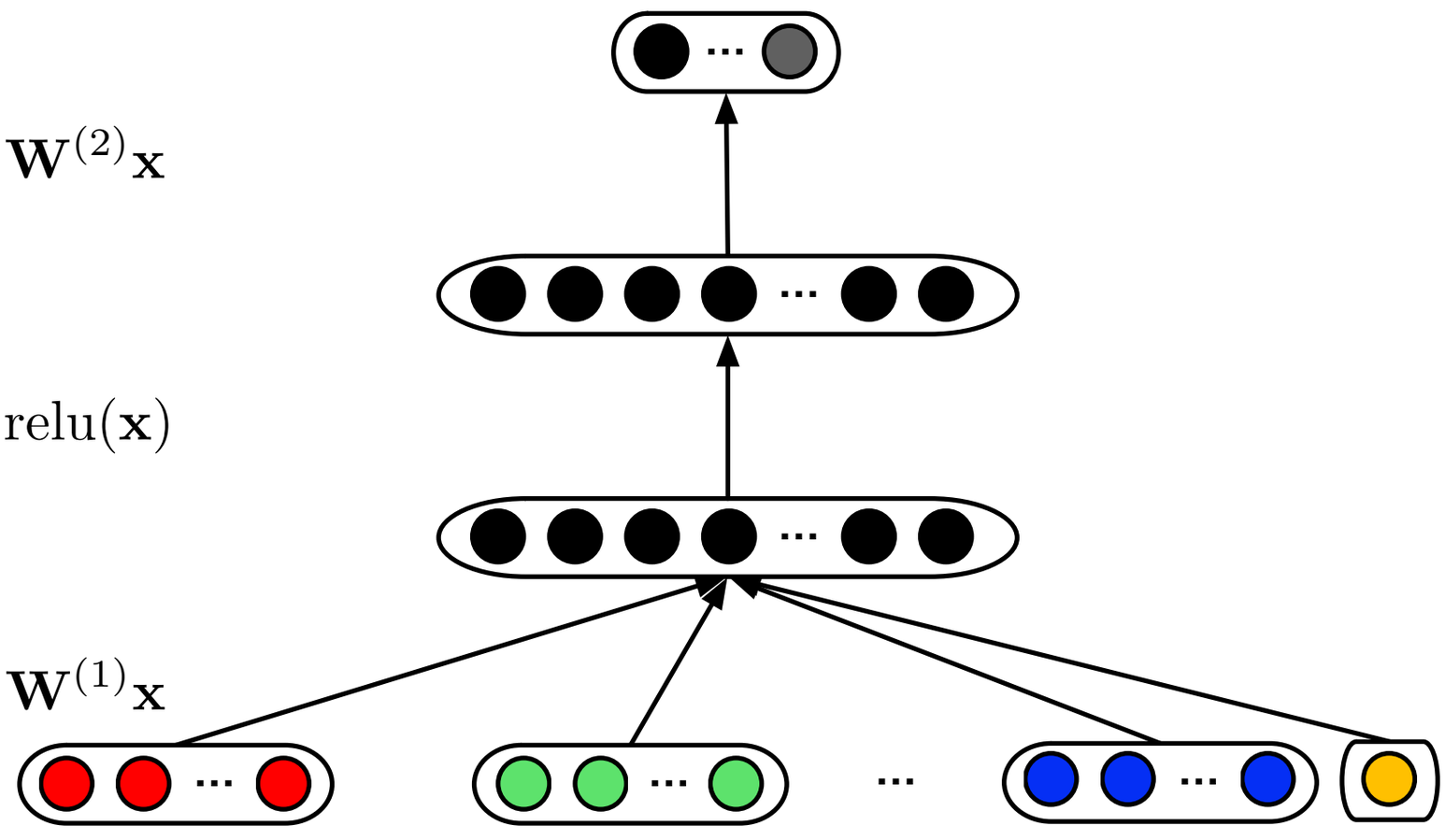
Fully Connected Layer

- **DNN**: DeepMood framework with a fully connected layer for data fusion.

$f(\mathbf{x}) = \mathbf{W}^{(2)}\mathbf{x}$

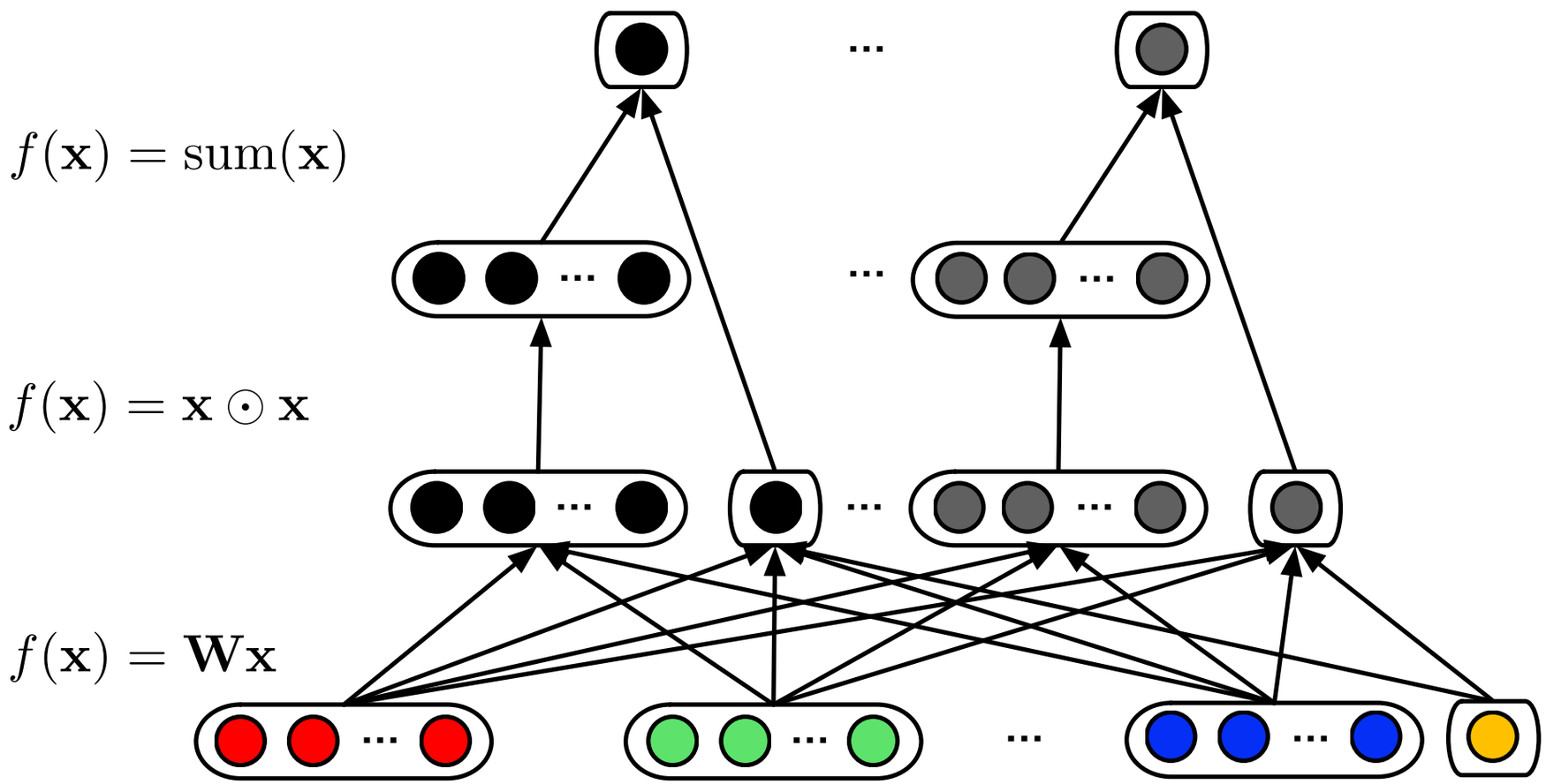
$f(\mathbf{x}) = \text{relu}(\mathbf{x})$

$f(\mathbf{x}) = \mathbf{W}^{(1)}\mathbf{x}$



Factorization Machine Layer

- **DFM**: DeepMood framework with a Factorization Machine layer for data fusion.



Factorization Machine Layer

$$\mathbf{q}_a = \mathbf{U}_a \mathbf{h}$$

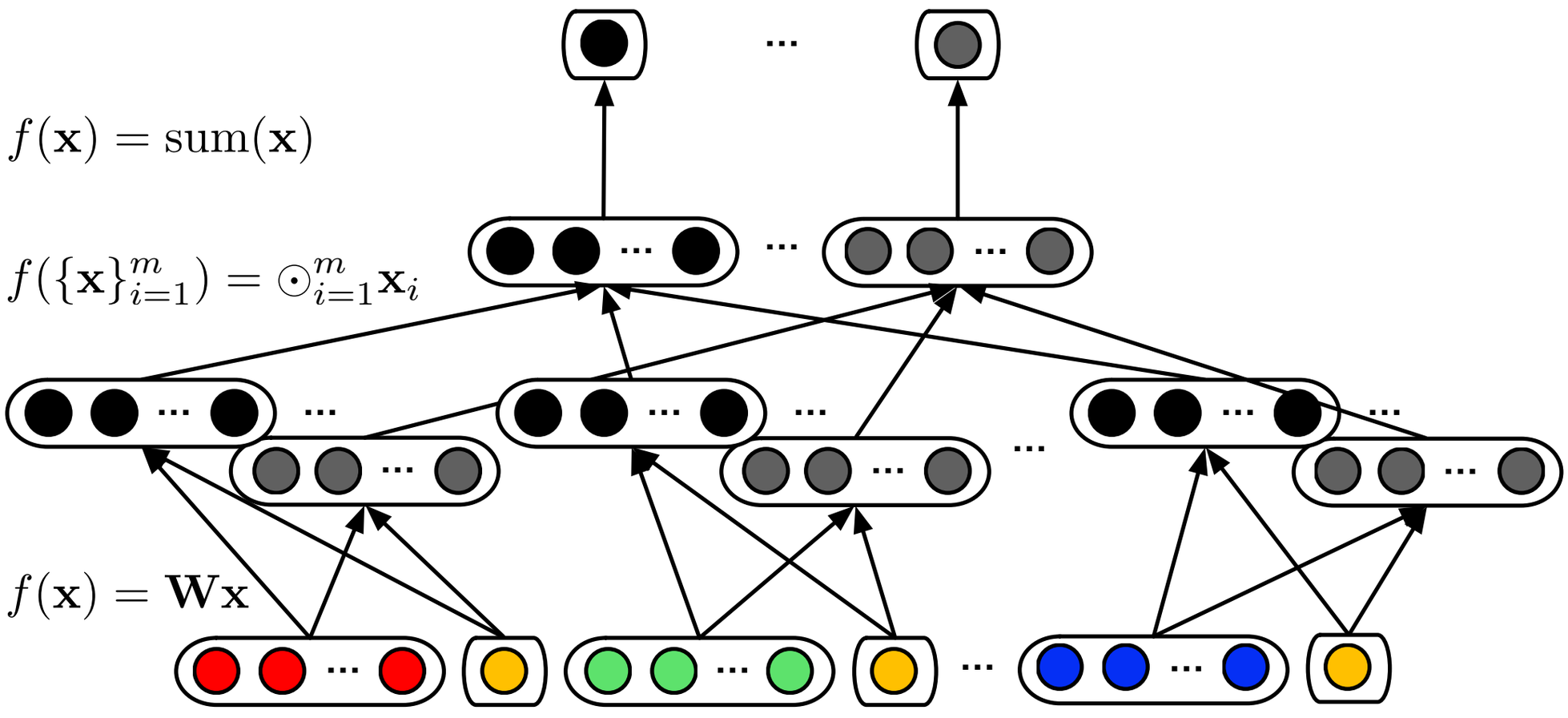
$$b_a = \mathbf{w}_a^T [\mathbf{h}; 1]$$

$$\hat{y}_a = \text{sum}([\mathbf{q}_a \odot \mathbf{q}_a; b_a])$$

$$\begin{aligned}
\hat{y}_a &= \sum_{f=1}^{k} \left(\sum_{i=1}^{d} \mathbf{U}_a(f, i)\mathbf{h}(i) \right)^2 + \sum_{i=1}^{d+1} \mathbf{w}_a(i)\bar{\mathbf{h}}(i) \\
&= \sum_{f=1}^{k} \left(\sum_{i=1}^{d} \mathbf{U}_a(f, i)\mathbf{h}(i) \right) \left(\sum_{j=1}^{d} \mathbf{U}_a(f, j)\mathbf{h}(j) \right) + \sum_{i=1}^{d+1} \mathbf{w}_a(i)\bar{\mathbf{h}}(i) \\
&= \sum_{f=1}^{k} \sum_{i=1}^{d} \sum_{j=1}^{d} \mathbf{U}_a(f, i)\mathbf{U}_a(f, j)\mathbf{h}(i)\mathbf{h}(j) + \sum_{i=1}^{d+1} \mathbf{w}_a(i)\bar{\mathbf{h}}(i) \\
&= \sum_{i=1}^{d} \sum_{j=1}^{d} \langle \mathbf{U}_a(:, i), \mathbf{U}_a(:, j) \rangle \mathbf{h}(i)\mathbf{h}(j) + \sum_{i=1}^{d} \mathbf{w}_a(i)\mathbf{h}(i) + \mathbf{w}_a(d+1)
\end{aligned}$$

Multi-View Machine Layer

- **DMVM**: DeepMood framework with a Multi-View Machine layer for data fusion.



Multi-View Machine Layer

$$\mathbf{q}_a^{(p)} = \mathbf{U}_a^{(p)}[\mathbf{h}^{(p)}; 1]$$

$$\hat{y}_a = \text{sum}([\mathbf{q}_a^{(1)} \odot \cdots \odot \mathbf{q}_a^{(m)}])$$

$$\hat{y}_a = \sum_{f=1}^{k} \prod_{p=1}^{m} \left(\sum_{i_p=1}^{d_h+1} \mathbf{U}_a^{(p)}(f, i_p) \bar{\mathbf{h}}^{(p)}(i_p) \right)$$

$$= \sum_{f=1}^{k} \sum_{i_1=1}^{d_h+1} \cdots \sum_{i_m=1}^{d_h+1} \left(\prod_{p=1}^{m} \mathbf{U}_a^{(p)}(f, i_p) \bar{\mathbf{h}}^{(p)}(i_p) \right)$$

$$= \sum_{i_1=1}^{d_h+1} \cdots \sum_{i_m=1}^{d_h+1} \left(\sum_{f=1}^{k} \prod_{p=1}^{m} \mathbf{U}_a^{(p)}(f, i_p) \right) \left(\prod_{p=1}^{m} \bar{\mathbf{h}}^{(p)}(i_p) \right)$$

Data Collection

Statistics of the BiAffect dataset.

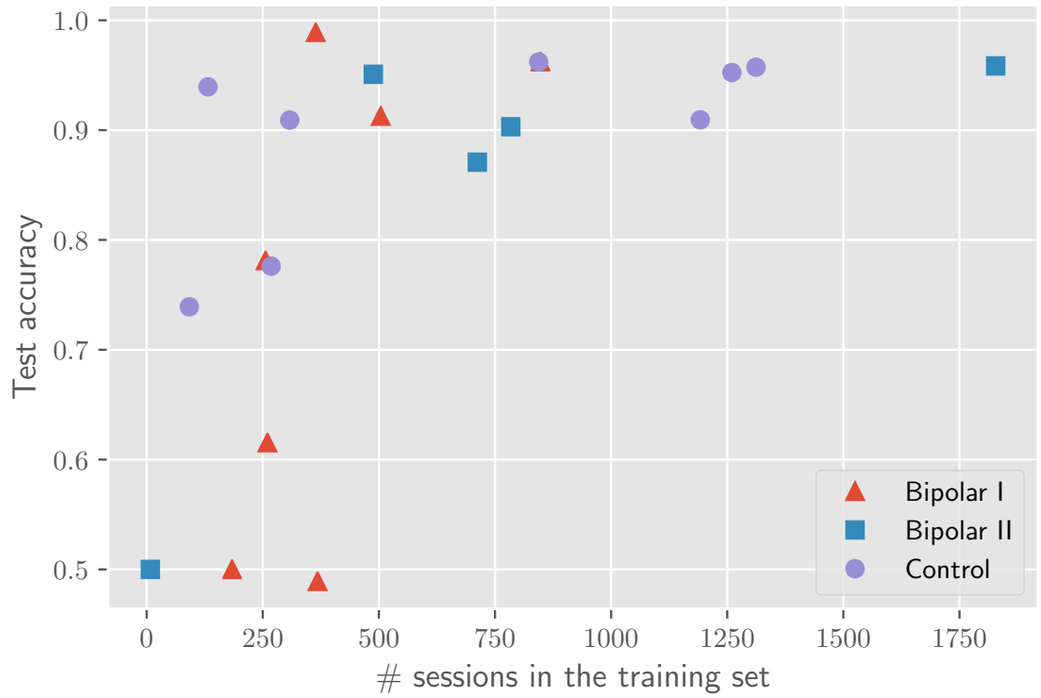
Statistics	ALPH.	SPEC.	ACCEL.
# data points	836,027	538,520	14,237,503
# sessions	34,993	33,385	37,647
Mean length	24	16	378
Median length	14	9	259
Maximum length	538	437	90,193

- Participants: 7 bipolar I, 5 bipolar II, 8 controls.
- Features: keypress metadata and accelerometer movement.
- Labels: Hamilton Depression Rating Scale (HDRS) and Young Mania Rating Scale (YMRS).

Prediction Performance

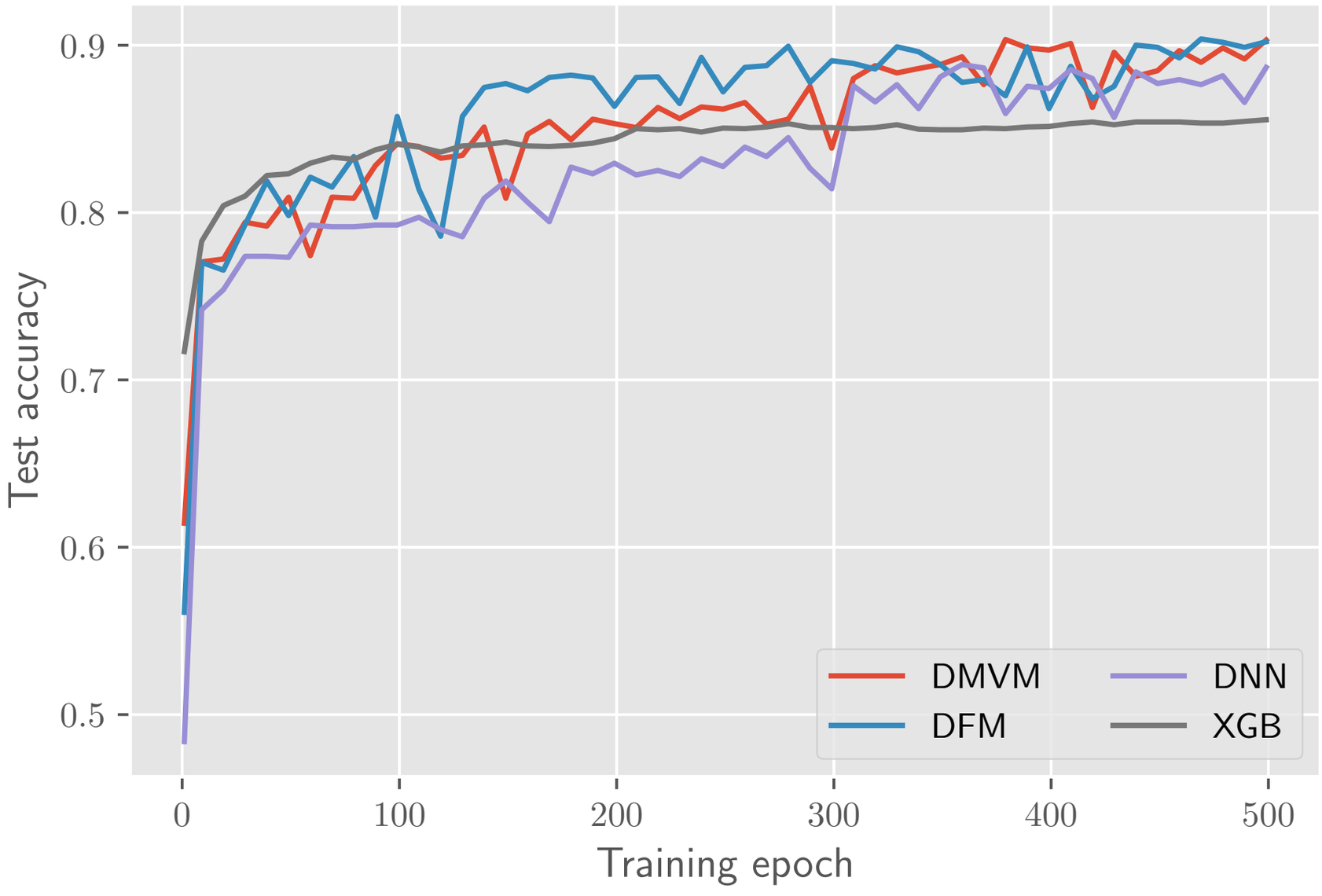
Prediction performance for mood detection.

Methods	Classification		Regression
	Accuracy	F-score	RMSE
DMVM	0.9031	0.9070	3.5664
DFM	0.9021	0.9029	3.6767
DNN	0.8868	0.8929	3.7874
XGB	0.8555	0.8562	3.9634
SVM	0.7323	0.7237	4.1257
LR	0.7293	0.7172	4.1822



- DMVM: DeepMood with a Multi-view Machine layer for data fusion.
- DFM: DeepMood with a Factorization Machine layer for data fusion.
- DNN: DeepMood with a conventional fully connected layer for data fusion.
- XGB: Tree boosting system XGBoost.
- SVM: Linear Support Vector Classification/Regression from scikit-learn.
- LR: Logistic/Ridge Regression from scikit-learn.

Convergence Efficiency



Summary

- Although NN models are capable of approximating many complicated functions in theory, there are practical difficulties in the learning process (e.g., local minima).
- It is critical to design NN modules that are suitable for different applications: convolution layers for computer vision, recurrent nets for sequence data, and some interaction operators for multi-view learning in the many other domains.

References

- Steffen Rendle. Factorization Machines. ICDM 2010.
- Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero and Larry Heck. Learning Deep Structured Semantic Models for Web Search using Clickthrough Data. CIKM 2013.
- Bokai Cao, Lifang He, Xiangnan Kong, Philip S. Yu, Zhifeng Hao and Ann B. Ragin. Tensor-based Multi-View Feature Selection with Applications to Brain Diseases. ICDM 2014.
- Bokai Cao, Hucheng Zhou, Guoqiang Li and Philip S. Yu. Multi-View Machines. WSDM 2016.
- Weinan Zhang, Tianming Du and Jun Wang. Deep Learning over Multi-Field Categorical Data: A Case Study on User Response Prediction. ECIR 2016.
- Ying Shan, T. Ryan Hoens, Jian Jiao, Haijing Wang, Dong Yu and JC Mao. Deep Crossing: Web-Scale Modeling without Manually Crafted Combinatorial Features. KDD 2016.
- Yuchin Juan, Yong Zhuang, Wei-Sheng Chin and Chih-Jen Lin. Field-aware Factorization Machines for CTR Prediction. RecSys 2016.
- Paul Covington, Jay Adams and Emre Sargin. Deep Neural Networks for YouTube Recommendations. RecSys 2016.
- Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, Rohan Anil, Zakaria Haque, Lichan Hong, Vihan Jain, Xiaobing Liu and Hemal Shah. Wide & Deep Learning for Recommender Systems. DLRS 2016.
- Yanru Qu, Han Cai, Kan Ren, Weinan Zhang, Yong Yu, Ying Wen and Jun Wang. Product-based Neural Networks for User Response Prediction. ICDM 2016.
- Chun-Ta Lu, Lifang He, Weixiang Shao, Bokai Cao and Philip S. Yu. Multilinear Factorization Machines for Multi-Task Multi-View Learning. WSDM 2017.
- Alexander Novikov, Mikhail Trofimov and Ivan Oseledets. Exponential Machines. ICLR 2017.
- Xiangnan He and Tat-Seng Chua. Neural Factorization Machines for Sparse Predictive Analytics. SIGIR 2017.
- Bokai Cao, Lei Zheng, Chenwei Zhang, Philip S. Yu, Andrea Piscitello, John Zulueta, Olu Ajilore, Kelly Ryan and Alex D. Leow. DeepMood: Modeling Mobile Phone Typing Dynamics for Mood Detection. KDD 2017.
- Jie Zhu, Ying Shan, JC Mao, Dong Yu, Holakou Rahmanian and Yi Zhang. Deep Embedding Forest: Forest-based Serving with Deep Embedding Features. KDD 2017.
- Jun Xiao, Hao Ye, Xiangnan He, Hanwang Zhang, Fei Wu and Tat-Seng Chua. Attentional Factorization Machines: Learning the Weight of Feature Interactions via Attention Networks. IJCAI 2017.
- Guorui Zhou, Chengru Song, Xiaoqiang Zhu, Xiao Ma, Yanghui Yan, Xingya Dai Han Zhu, Junqi Jin, Han Li and Kun Gai. Deep Interest Network for Click-Through Rate Prediction. arXiv 2017.
- Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li and Xiuqiang He. DeepFM: A Factorization-Machine based Neural Network for CTR Prediction. arXiv 2017.
- Ruoxi Wang, Bin Fu, Gang Fu and Mingliang Wang. Deep & Cross Network for Ad Click Predictions. arXiv 2017.
- Chun-Ta Lu, Lifang He, Hao Ding, Bokai Cao and Philip S. Yu. Learning from Multi-View Structural Data via Structural Factorization Machines. WWW 2018.