Learning Hierarchical Feature Influence for Recommendation by Recursive Regularisation

Jie Yang, Zhu Sun, Alessandro Bozzon,
Jie Zhang

TU Delft
Delft University of Technology

NANYANG TECHNOLOGICAL UNIVERSITY

Objective

Fully exploit feature hierarchies for recommendation by utilising the structured information, and historical data

Feature Hierarchy

Feature Hierarchies are everywhere

matters

Science and Engineering

- Biology
- Chemistry
- Machine learning
- Natural language processing
- ...

Feature Hierarchy

Feature Hierarchies are everywhere

matters

Science and Engineering

- Biology
- Chemistry
- Machine learning
- Natural language processing
- …

Recommender Systems

Feature Hierarchy

Science and Engineering

- Biology
- Chemistry
- Machine learning
- Natural language processing
- ...

matters

Feature Hierarchies are everywhere

Product Category

amazon

Recommender Systems

Feature Hierarchy

Feature Hierarchies are everywhere

matters

Science and Engineering

- Biology
- Chemistry
- Machine learning
- Natural language processing
- ...

Recommender Systems

Product Category

Article Topic

amazon

Wikipedia

Feature

Hierarchy

Feature Hierarchies are everywhere

matters

Science and Engineering

- Biology
- Chemistry
- Machine learning
- Natural language processing
- …

Recommender Systems

- Product Category
- Article Topic
- Music Genre

amazon

Wikipedia

YAHOO! MUSIC

Feature Hierarchy

Feature Hierarchies are everywhere

matters

Science and Engineering

- Biology
- Chemistry
- Machine learning
- Natural language processing
- ...

Recommender Systems

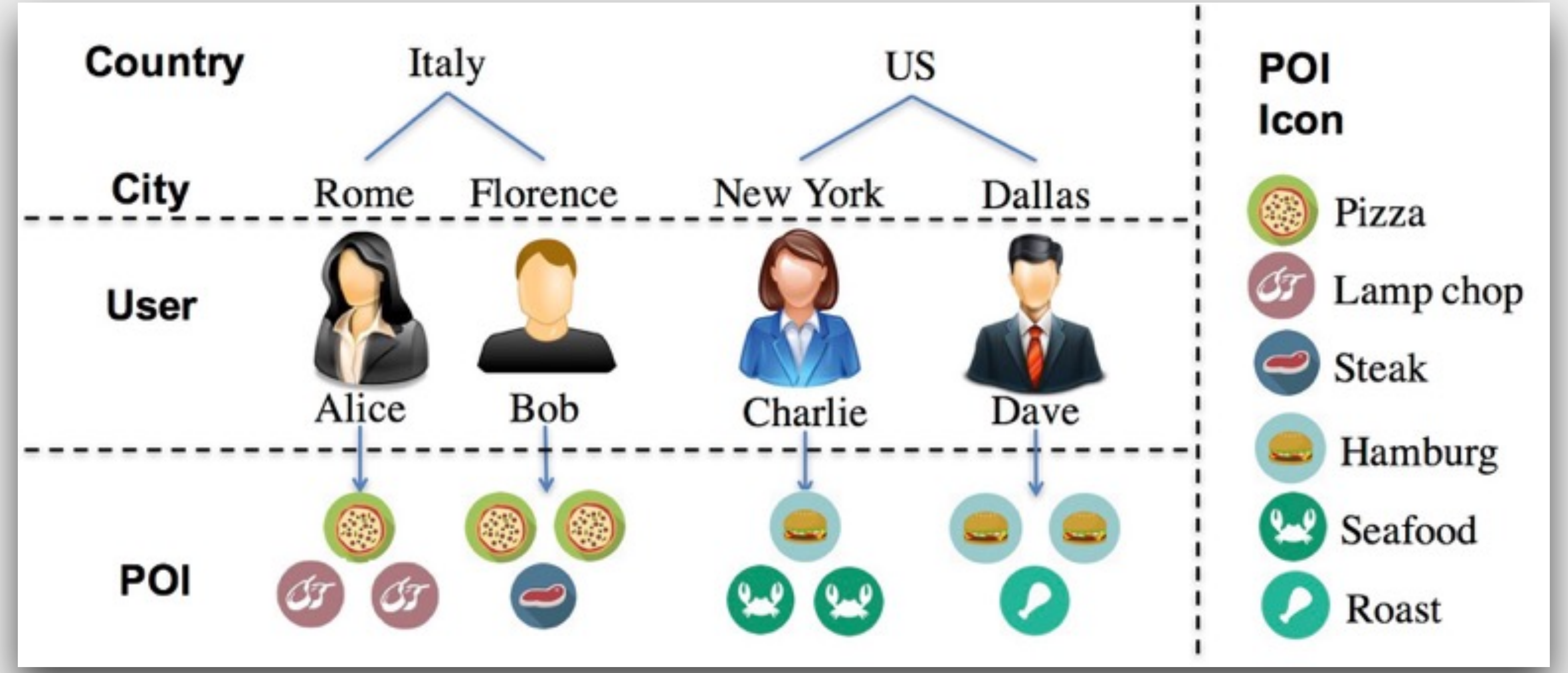
amazon
WIKIPEDIA
YAHOO! MUSIC
FOURSQUARE

- Product Category
- Article Topic
- Music Genre
- Venue Category

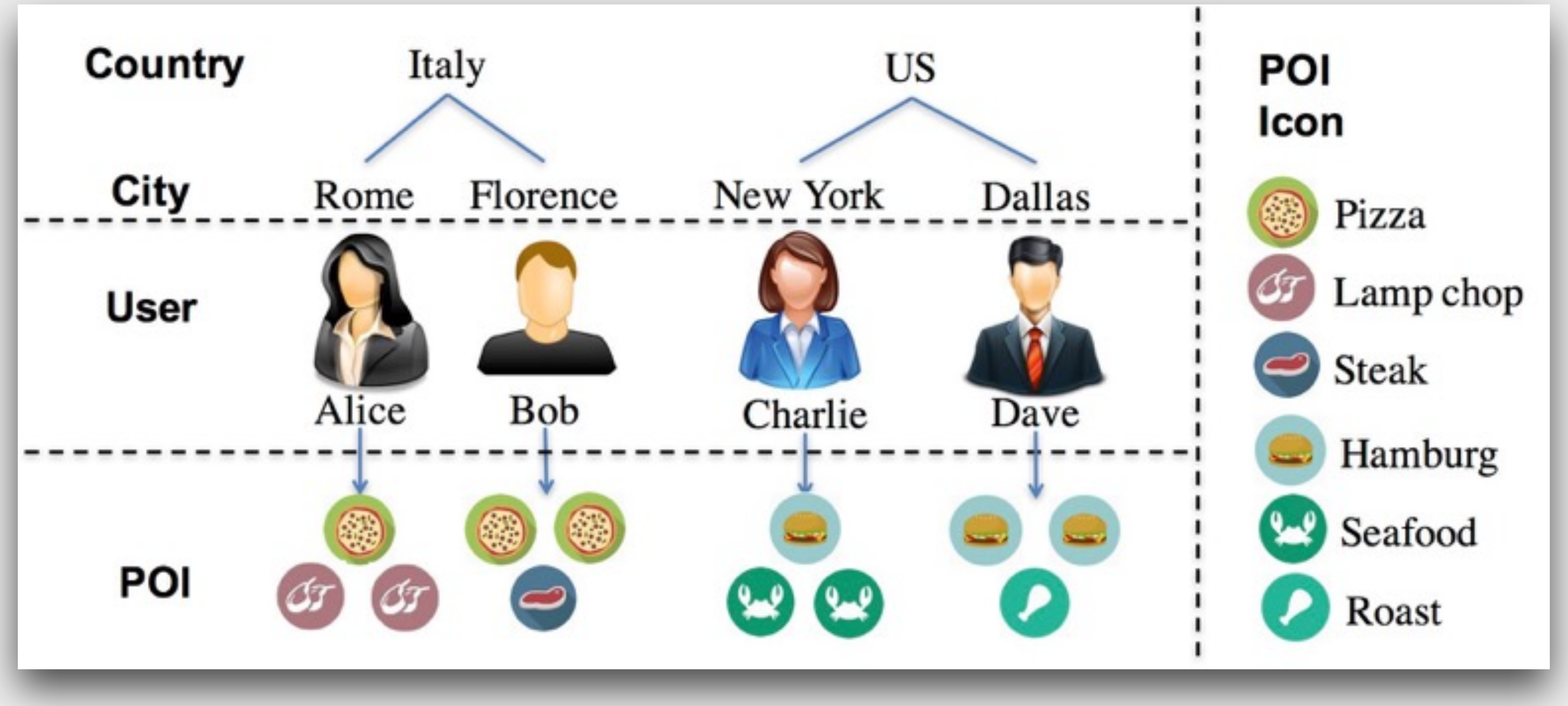
...

How can feature hierarchies be used?

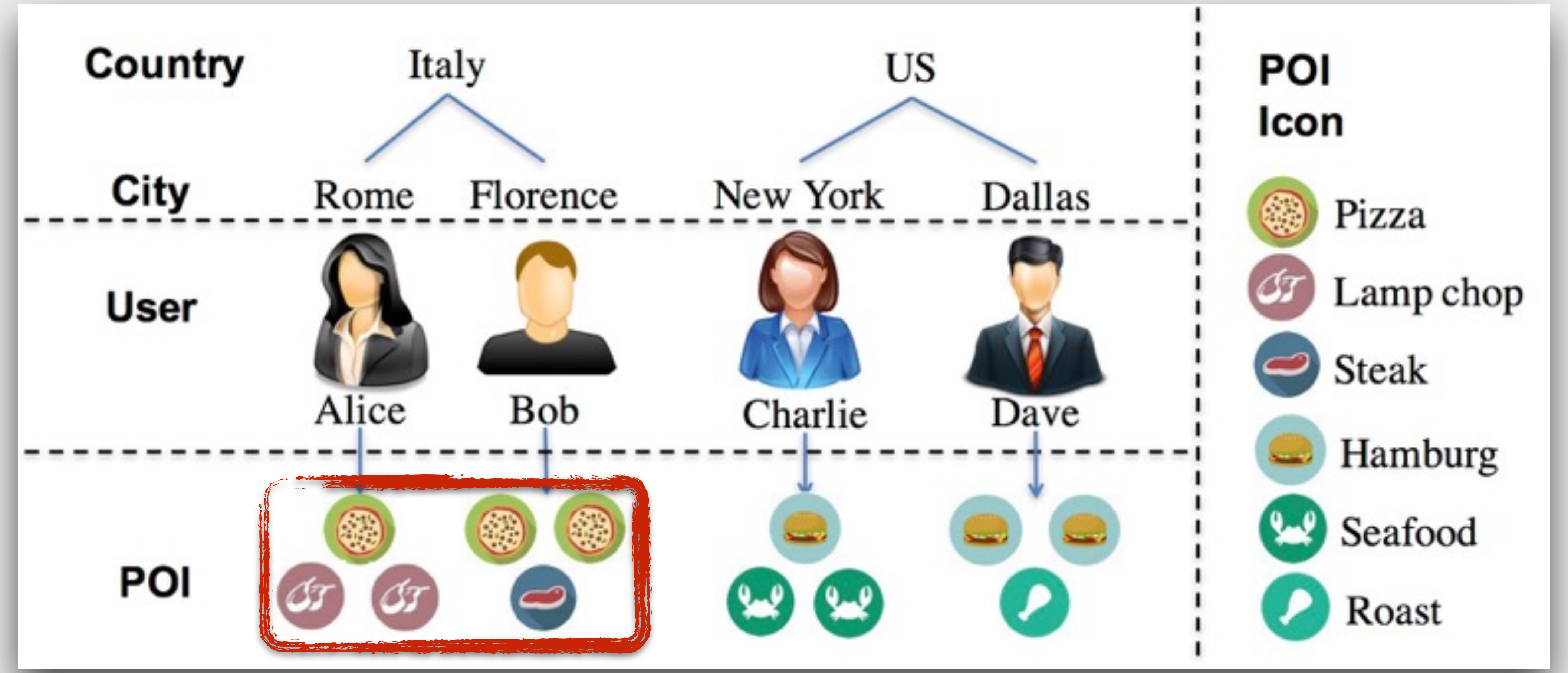
Example


Country
Italy
US
City
Rome
Florence
New York
Dallas
User
Alice
Bob
Charlie
Dave
POI
POI
Icon
Pizza
Lamp chop
Steak
Hamburg
Seafood
Roast

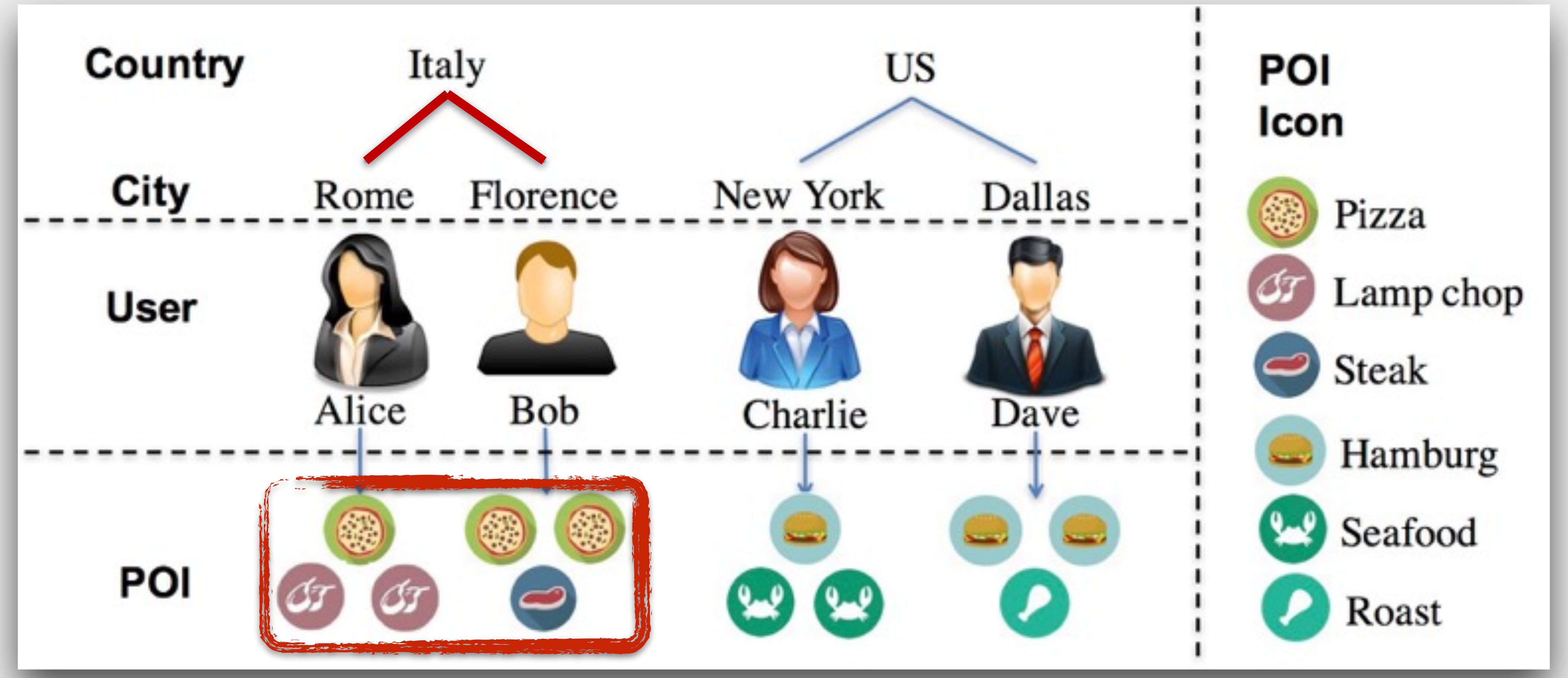
Structure to be considered



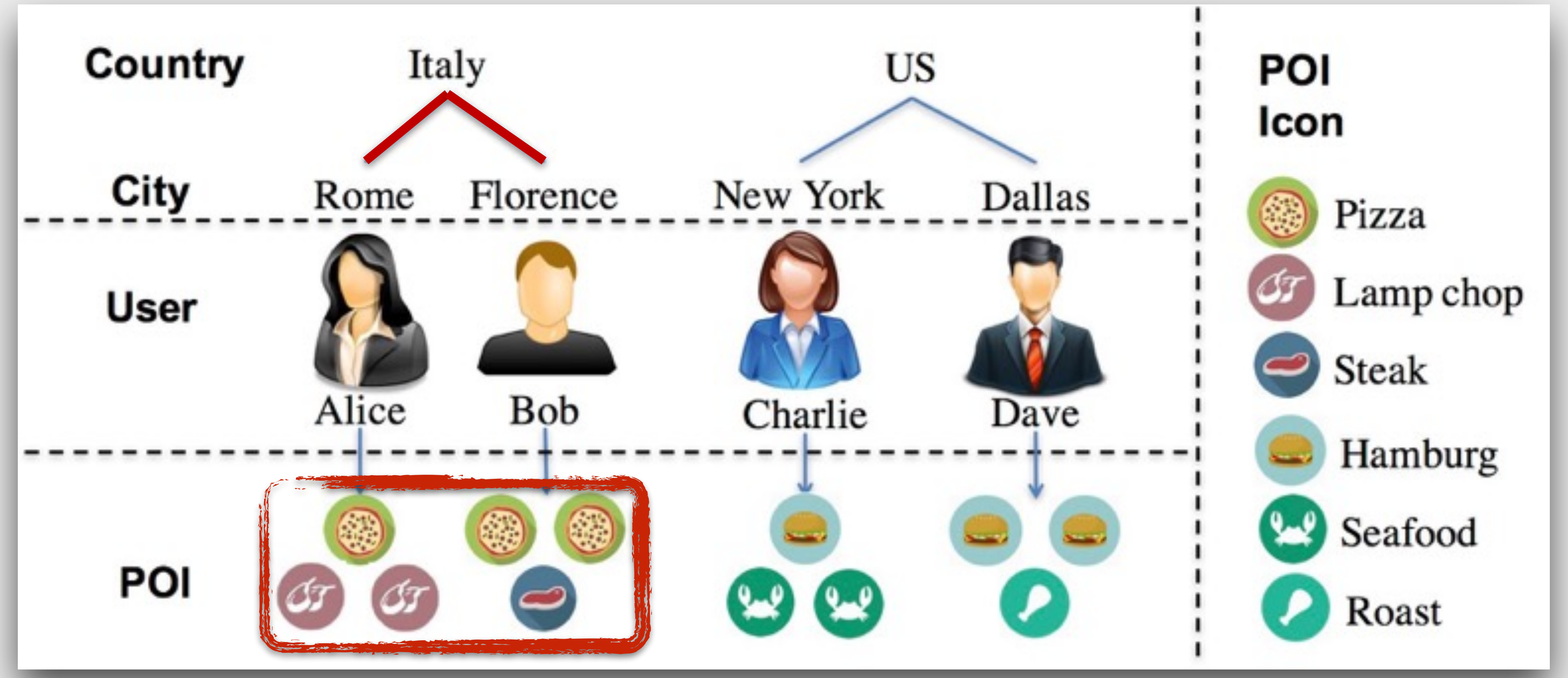
Structure to be considered



Structure to be considered

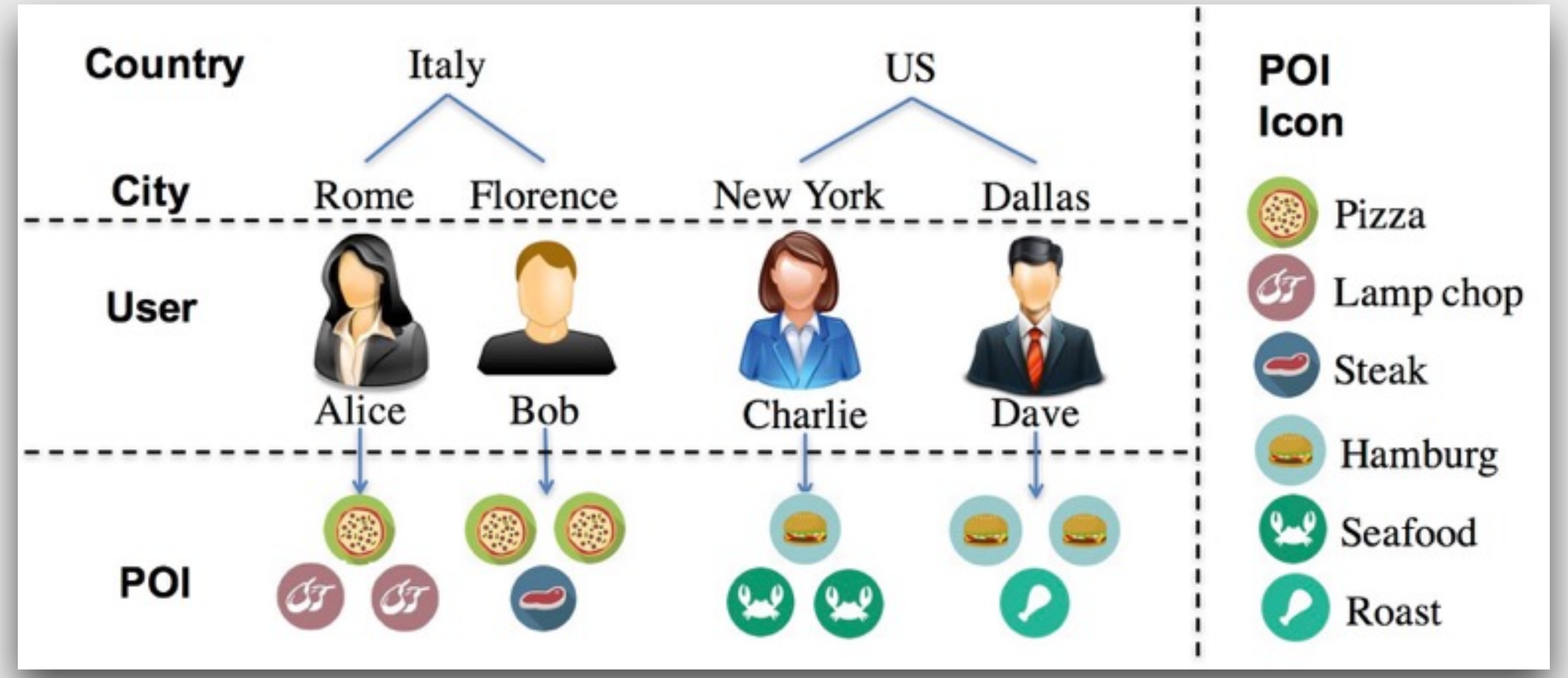


Structure to be considered

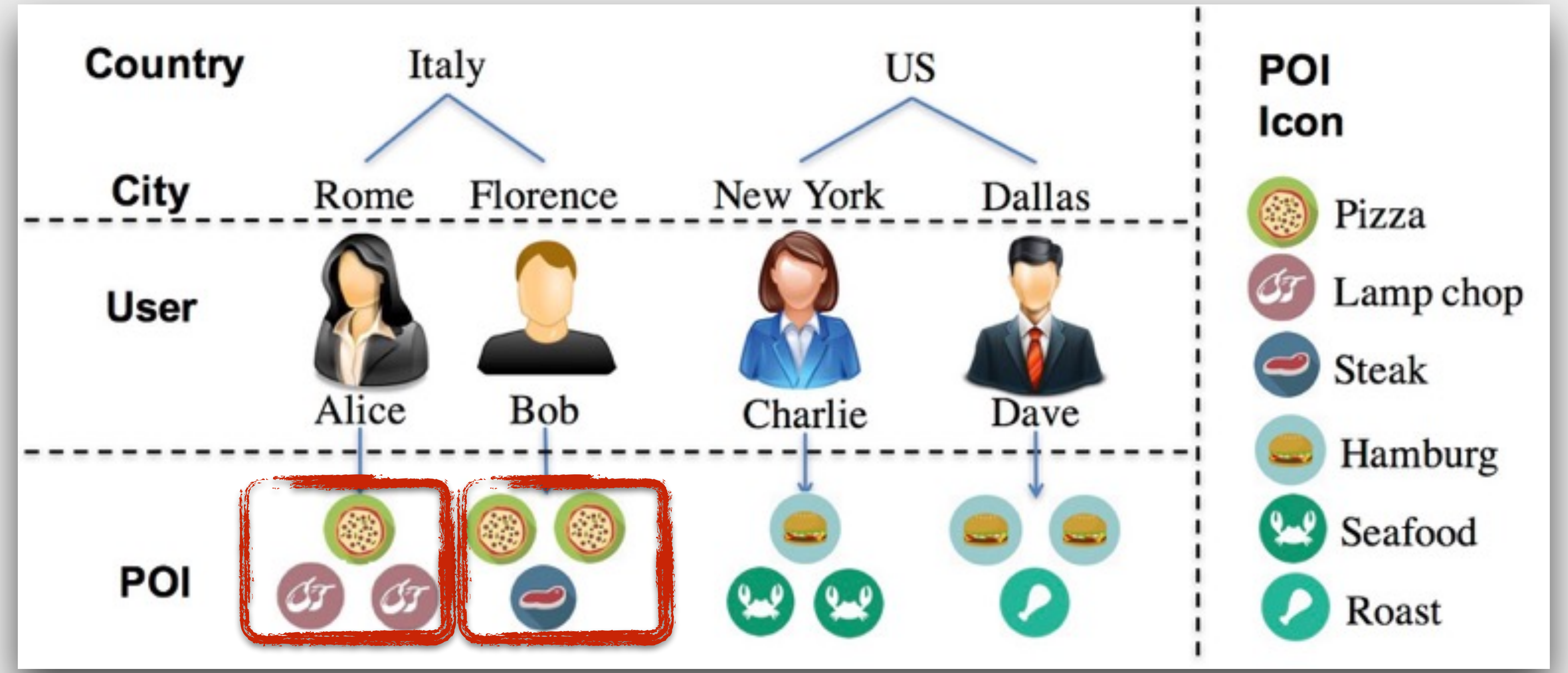


State-of-the-art methods reduce hierarchies to simpler structures (e.g. flat), which brings severe information loss

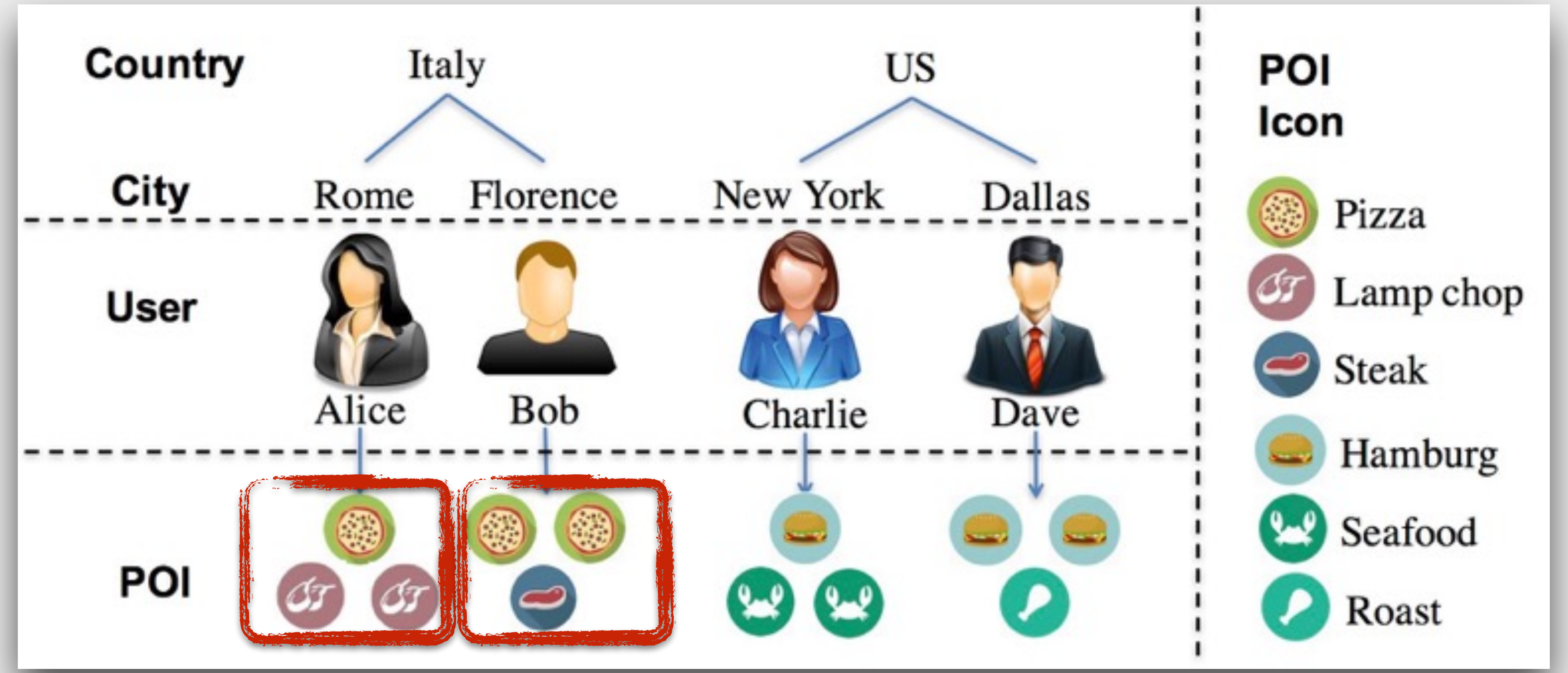
Data is also important



Data is also important



Data is also important



The different *influence* of structured features can only be learnt from historical user-item interaction data

Objective

Fully exploit feature hierarchies for recommendation by utilising the structured information, and historical data

Objective

Fully exploit feature hierarchies for recommendation by utilising the structured information, and historical data

Solution:

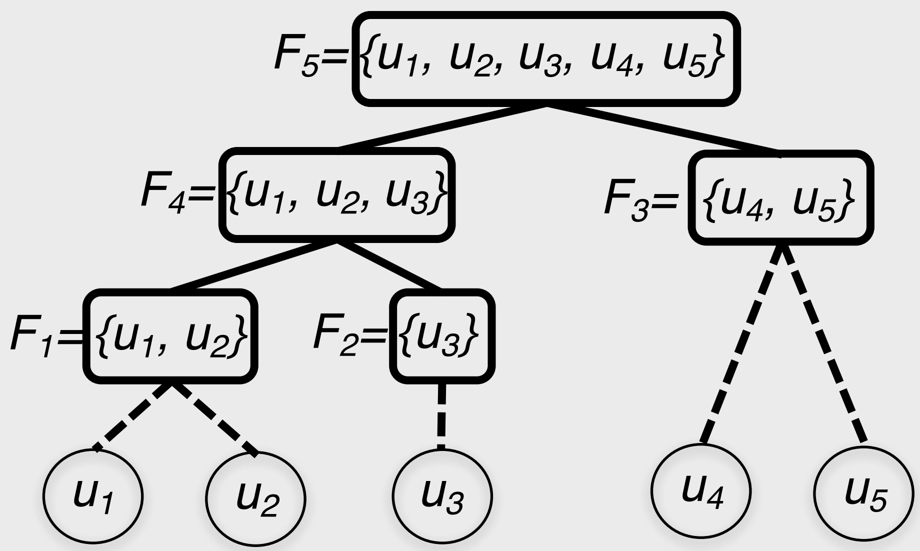
Recursive Regularisation

Basic Model

Latent factor model

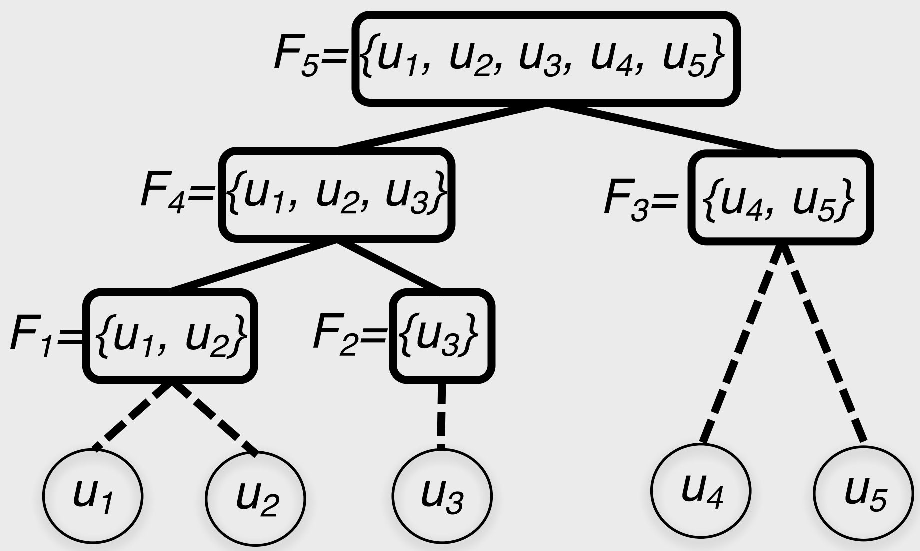
$$\min_{\mathbf{U},\mathbf{V}} \frac{1}{2} \sum_{i,j} \mathbf{O}_{ij} (\mathbf{R}_{ij} - \mathbf{U}_i \mathbf{V}_j^T)^2 + \frac{\lambda}{2} (\|\mathbf{U}\|_F^2 + \|\mathbf{V}\|_F^2)$$

Feature Influence (1)



Influence of an ***isolated*** feature:

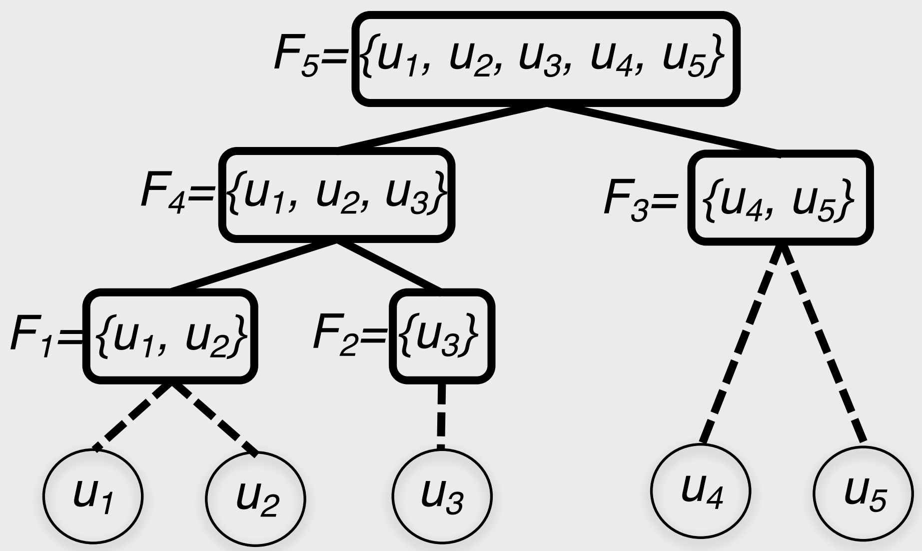
Feature Influence (1)



Influence of an ***isolated*** feature:

$$Dis(F_p) = \sum_{u_i, u_k \in F_p, i<k} \|\mathbf{U}_i - \mathbf{U}_k\|_F^2$$

Feature Influence (1)

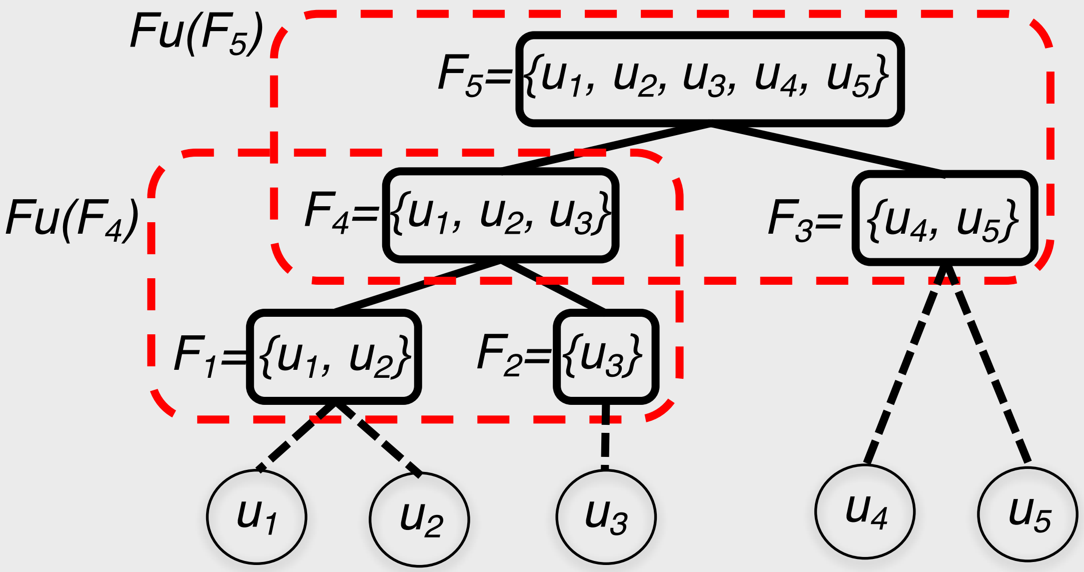


Influence of an ***isolated*** feature:

$$Dis(F_p) = \sum_{u_i, u_k \in F_p, i<k} \|\mathbf{U}_i - \mathbf{U}_k\|_F^2$$

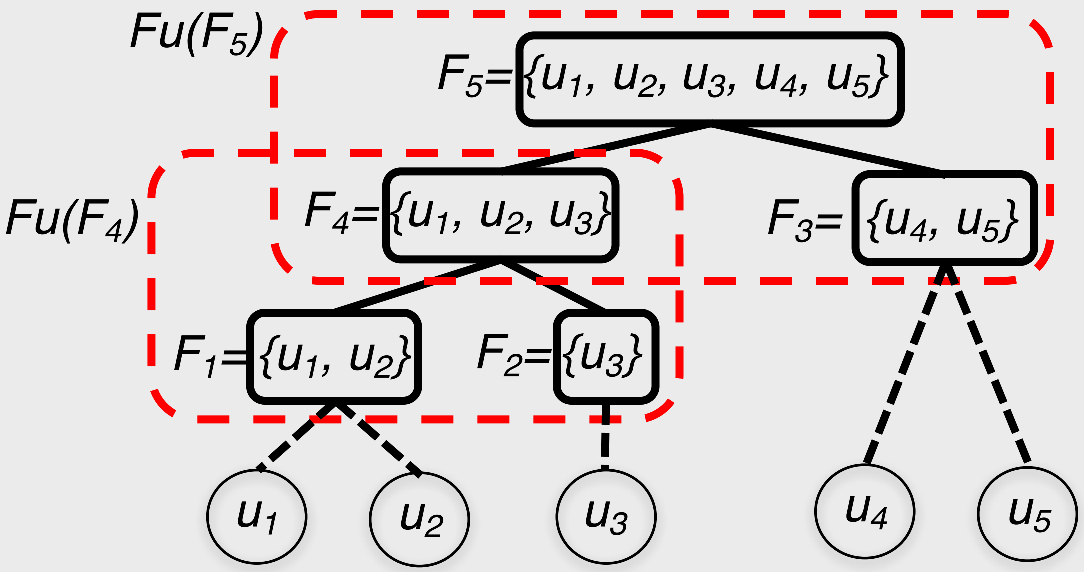
E.g. the influence of $\boldsymbol{F_4}$ is the dissimilarity of latent factors of all users characterised by it, i.e. $\boldsymbol{u_1}$, $\boldsymbol{u_2}$, $\boldsymbol{u_3}$.

Feature Influence (2)



Influence of an ***isolated*** feature ***unit*** *(a parent with its children)*:

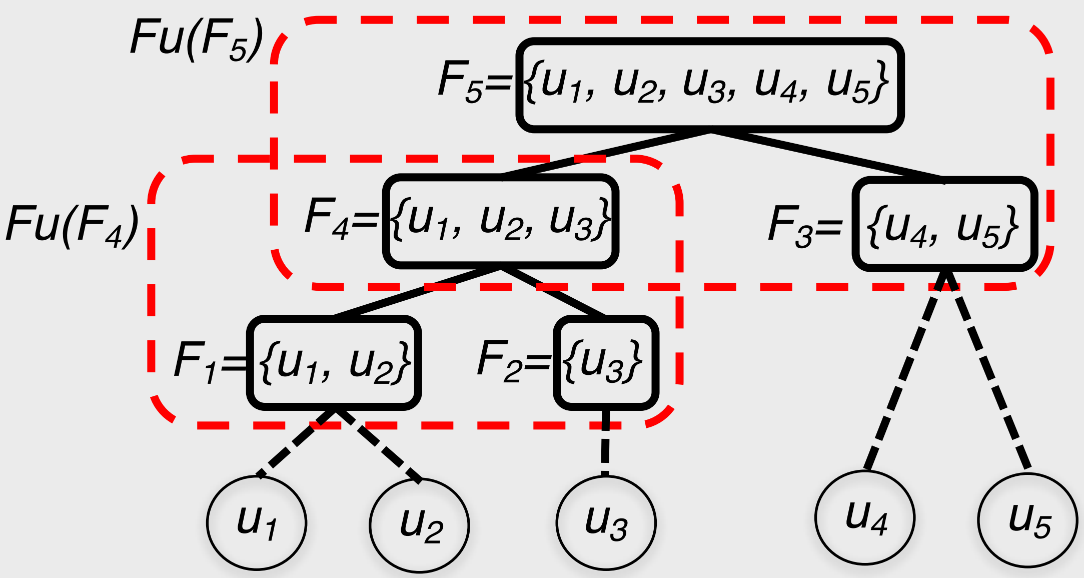
Feature Influence (2)



Influence of an ***isolated*** feature ***unit*** *(a parent with its children)*:

$$\mathbf{I}'(F_p) = g_p Dis(F_p) + s_p(\sum_{\forall F_c \in children(F_p)} Dis(F_c))$$

Feature Influence (2)

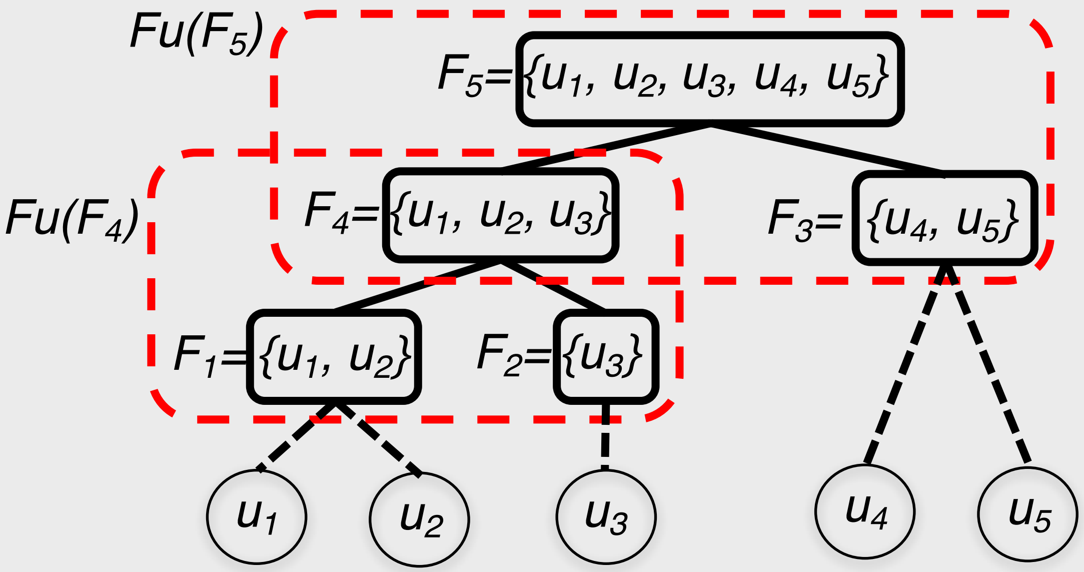


Influence of an ***isolated*** feature ***unit*** *(a parent with its children)*:

$$\mathbf{I}'(F_p) = g_p Dis(F_p) + s_p \sum_{\forall F_c \in children(F_p)} Dis(F_c))$$

$$g_p + s_p = 1$$

Feature Influence (2)



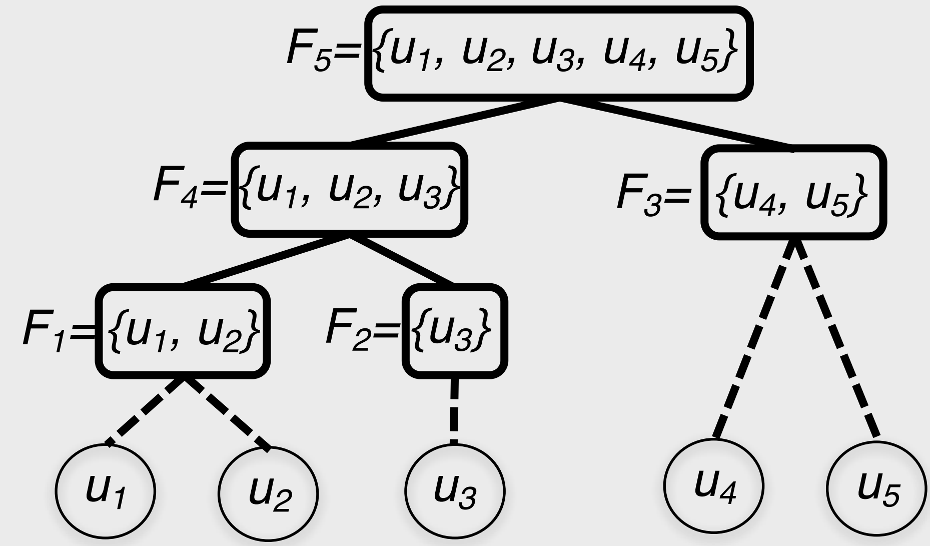
Influence of an ***isolated*** feature ***unit*** *(a parent with its children)*:

$$\mathbf{I}'(F_p) = g_p Dis(F_p) + s_p \sum_{\forall F_c \in children(F_p)} Dis(F_c)) \qquad g_p + s_p = 1$$

E.g. the influence of feature unit *I'(F5)* is

$$\mathbf{I}'(F_5) = g_5 Dis(F_5) + s_5(Dis(F_4) + Dis(F_3))$$

Recursive Regularisation



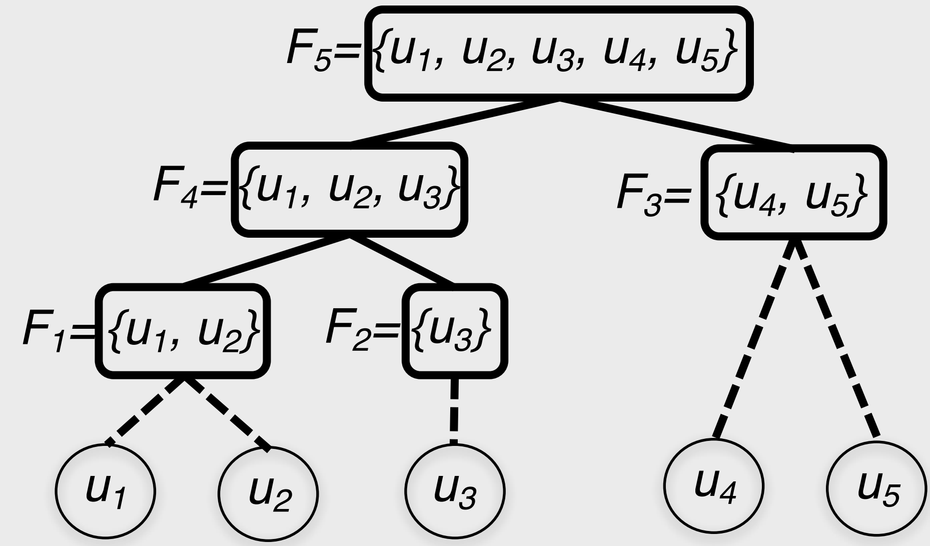
Influence of a feature hierarchy:

DEFINITION 1 (**RECURSIVE REGULARIZATION**).

$$\mathbf{I}(F_p) = \begin{cases} g_p Dis(F_p) \quad +s_p(\sum_{\forall F_c \in children(F_p)} \mathbf{I}(F_c)), \\ \qquad \textit{if } F_p \textit{ is an internal feature;} \\ Dis(F_p), \quad \textit{if } F_p \textit{ is a leaf feature and } |F_p| > 1; \\ 0, \quad \textit{otherwise,} \end{cases}$$

where $|F_p|$ *is the number of users characterized by feature* F_p.

Recursive Regularisation



Influence of a feature hierarchy:

DEFINITION 1 (**RECURSIVE REGULARIZATION**).

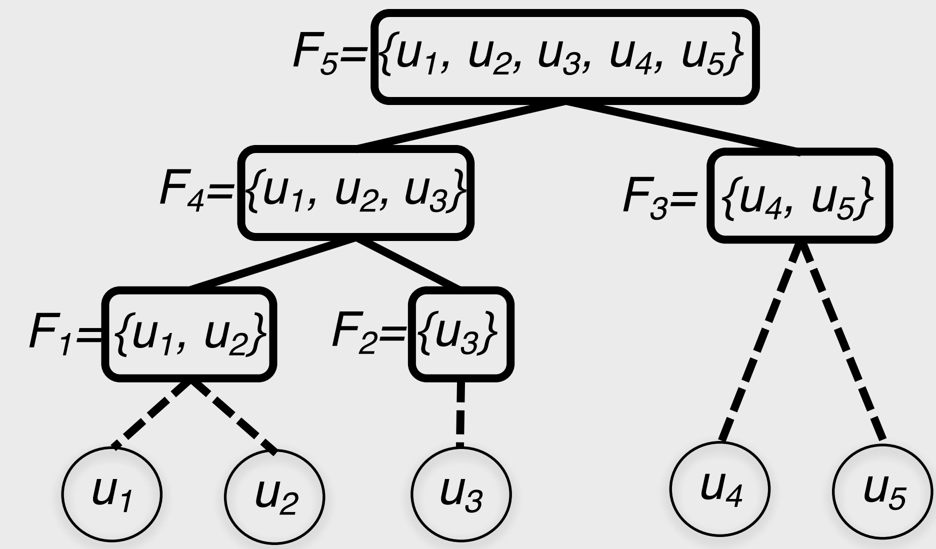
$$\mathbf{I}(F_p) = \begin{cases} g_p Dis(F_p) \quad +s_p(\sum_{\forall F_c \in children(F_p)} \mathbf{I}(F_c)), \\ \qquad \text{if } F_p \text{ is an internal feature;} \\ Dis(F_p), \quad \text{if } F_p \text{ is a leaf feature and } |F_p| > 1; \\ 0, \quad \text{otherwise,} \end{cases}$$

where $|F_p|$ is the number of users characterized by feature F_p.

E.g. the influence the above hierarchy is:

$$\begin{aligned} &\mathbf{I}(\mathcal{F}) = \mathbf{I}(F_5) \\ &= g_5 Dis(F_5) + s_5(\mathbf{I}(F_4) + \mathbf{I}(F_3)) \\ &= g_5 Dis(F_5) + s_5(g_4 Dis(F_4) + s_4(\mathbf{I}(F_1) + \mathbf{I}(F_2)) + Dis(F_3)) \\ &= g_5 Dis(F_5) + s_5(g_4 Dis(F_4) + s_4 Dis(F_1) + Dis(F_3)). \end{aligned}$$

Recursive Regularisation



Influence of a feature hierarchy:

Definition 1 (Recursive Regularization).

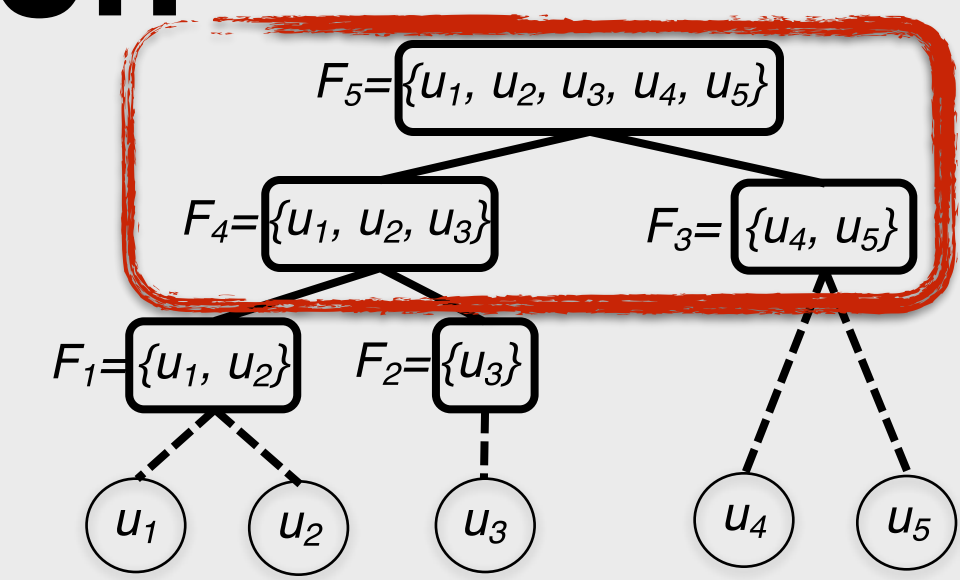
$$\mathbf{I}(F_p) = \begin{cases} g_p Dis(F_p) \quad + s_p\left(\sum_{\forall F_c \in children(F_p)} \mathbf{I}(F_c)\right), & \text{if } F_p \text{ is an internal feature;} \\ Dis(F_p), & \text{if } F_p \text{ is a leaf feature and } |F_p| > 1; \\ 0, & \text{otherwise,} \end{cases}$$

where $|F_p|$ is the number of users characterized by feature F_p.

E.g. the influence the above hierarchy is:

$$\begin{aligned} &\mathbf{I}(\mathcal{F}) = \mathbf{I}(F_5) \\ =&g_5 Dis(F_5) + s_5(\mathbf{I}(F_4) + \mathbf{I}(F_3)) \\ =&g_5 Dis(F_5) + s_5(g_4 Dis(F_4) + s_4(\mathbf{I}(F_1) + \mathbf{I}(F_2)) + Dis(F_3)) \\ =&g_5 Dis(F_5) + s_5(g_4 Dis(F_4) + s_4 Dis(F_1) + Dis(F_3)). \end{aligned}$$

Recursive Regularisation



Influence of a feature hierarchy:

Definition 1 (**Recursive Regularization**).

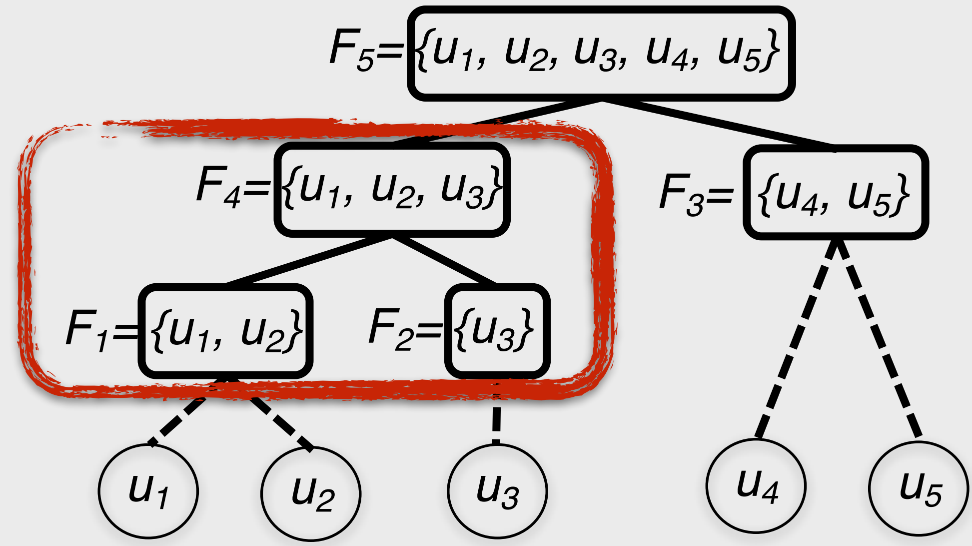
$$\mathbf{I}(F_p) = \begin{cases} g_p Dis(F_p) \quad + s_p\left(\sum_{\forall F_c \in children(F_p)} \mathbf{I}(F_c)\right), & \\ \qquad\qquad \text{if } F_p \text{ is an internal feature;} & \\ Dis(F_p), \quad \text{if } F_p \text{ is a leaf feature and } |F_p| > 1; & \\ 0, \quad \text{otherwise,} & \end{cases}$$

where $|F_p|$ is the number of users characterized by feature F_p.

E.g. the influence the above hierarchy is:

$$\mathbf{I}(\mathcal{F}) = \mathbf{I}(F_5)$$
$$= g_5 Dis(F_5) + s_5(\mathbf{I}(F_4) + \mathbf{I}(F_3))$$
$$= g_5 Dis(F_5) + s_5(g_4 Dis(F_4) + s_4(\mathbf{I}(F_1) + \mathbf{I}(F_2)) + Dis(F_3))$$
$$= g_5 Dis(F_5) + s_5(g_4 Dis(F_4) + s_4 Dis(F_1) + Dis(F_3)).$$

Recursive Regularisation



Influence of a feature hierarchy:

DEFINITION 1 (**RECURSIVE REGULARIZATION**).

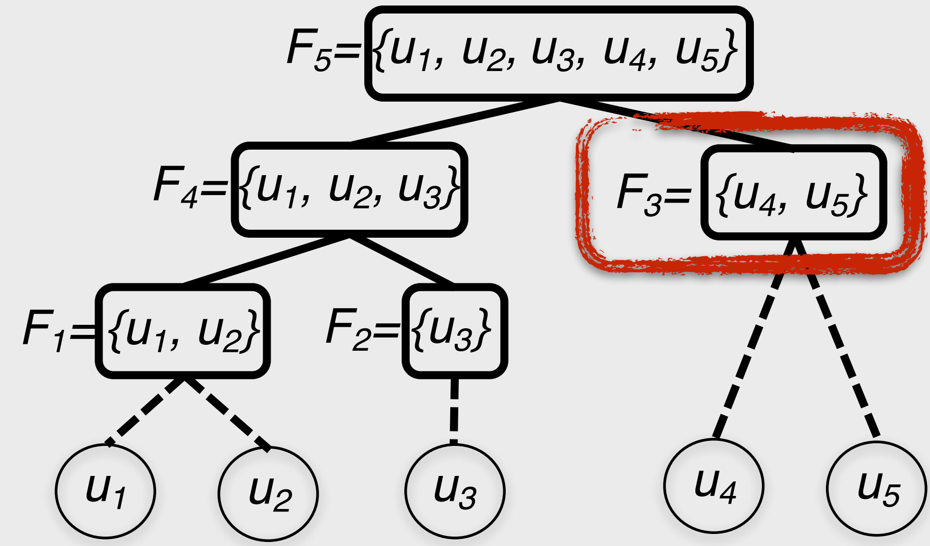
$$\mathbf{I}(F_p) = \begin{cases} g_p Dis(F_p) \quad + s_p\left(\sum_{\forall F_c \in children(F_p)} \mathbf{I}(F_c)\right), \\ \qquad\qquad \text{if } F_p \text{ is an internal feature;} \\ Dis(F_p), \quad \text{if } F_p \text{ is a leaf feature and } |F_p| > 1; \\ 0, \quad \text{otherwise,} \end{cases}$$

where $|F_p|$ *is the number of users characterized by feature* F_p.

E.g. the influence the above hierarchy is:

$$\begin{aligned} &\mathbf{I}(\mathcal{F}) = \mathbf{I}(F_5) \\ =&g_5 Dis(F_5) + s_5(\mathbf{I}(F_4) + \mathbf{I}(F_3)) \\ =&g_5 Dis(F_5) + s_5(g_4 Dis(F_4) + s_4(\mathbf{I}(F_1) + \mathbf{I}(F_2)) + Dis(F_3)) \\ =&g_5 Dis(F_5) + s_5(g_4 Dis(F_4) + s_4 Dis(F_1) + Dis(F_3)). \end{aligned}$$

Recursive Regularisation



Influence of a feature hierarchy:

DEFINITION 1 (**RECURSIVE REGULARIZATION**).

$$\mathbf{I}(F_p) = \begin{cases} g_p Dis(F_p) + s_p\left(\sum_{\forall F_c \in children(F_p)} \mathbf{I}(F_c)\right), & \text{if } F_p \text{ is an internal feature;} \\ Dis(F_p), & \text{if } F_p \text{ is a leaf feature and } |F_p| > 1; \\ 0, & \text{otherwise,} \end{cases}$$

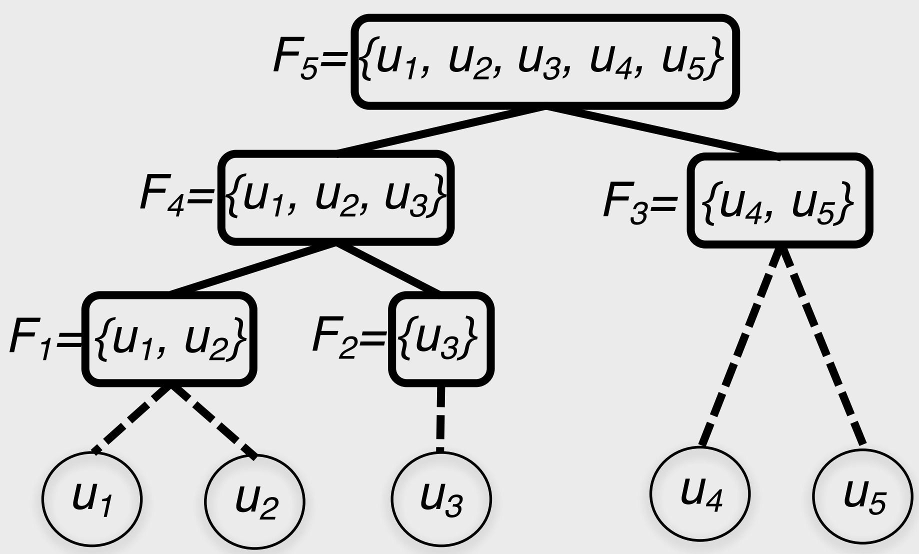
where $|F_p|$ is the number of users characterized by feature F_p.

E.g. the influence the above hierarchy is:

$$\begin{aligned} &\mathbf{I}(\mathcal{F}) = \mathbf{I}(F_5) \\ =& g_5 Dis(F_5) + s_5(\mathbf{I}(F_4) + \mathbf{I}(F_3)) \\ =& g_5 Dis(F_5) + s_5(g_4 Dis(F_4) + s_4(\mathbf{I}(F_1) + \mathbf{I}(F_2)) + Dis(F_3)) \\ =& g_5 Dis(F_5) + s_5(g_4 Dis(F_4) + s_4 Dis(F_1) + Dis(F_3)). \end{aligned}$$

Recursive Regularisation

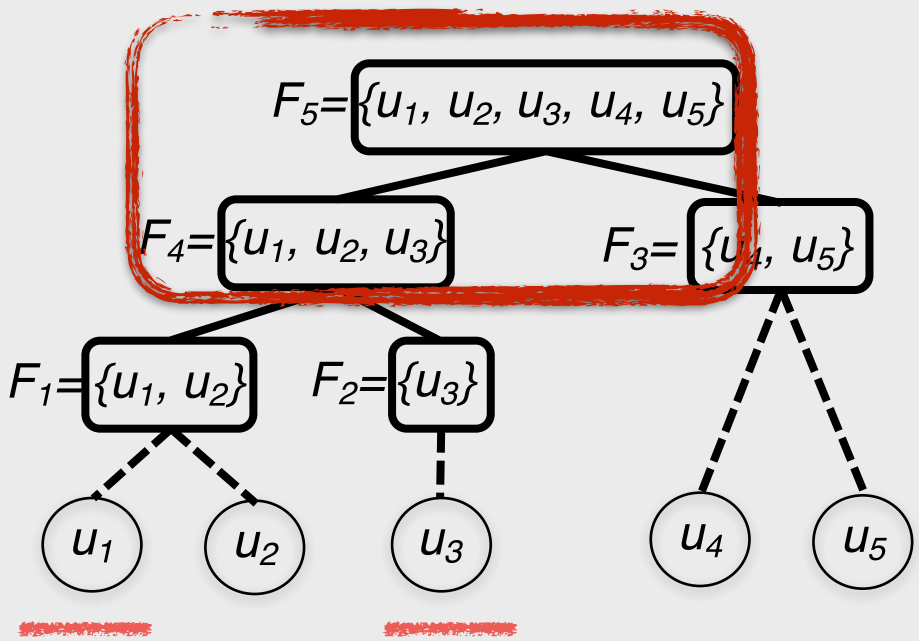
With recursive regularisation, we can express the influence of features on users as a parameterised function of g, and s



	u_1	u_2	u_3	u_4	u_5
u_1	-	1	$g_5+s_5g_4$	g_5	g_5
u_2	1	-	$g_5+s_5g_4$	g_5	g_5
u_3	$g_5+s_5g_4$	$g_5+s_5g_4$	-	g_5	g_5
u_4	g_5	g_5	g_5	-	1
u_5	g_5	g_5	g_5	1	-

Recursive Regularisation

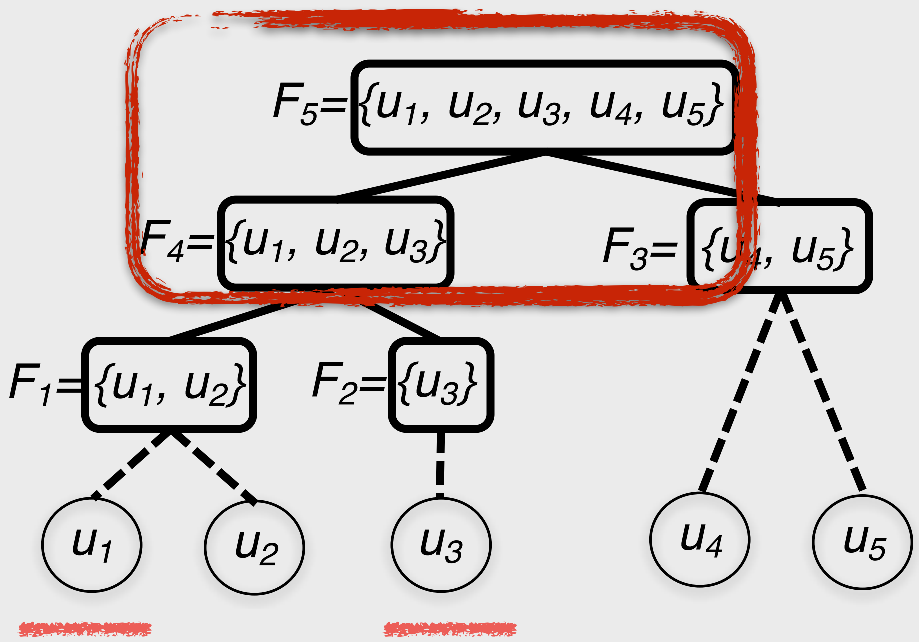
With recursive regularisation, we can express the influence of features on users as a parameterised function of g, and s



	u_1	u_2	u_3	u_4	u_5
u_1	-	1	$g_5+s_5g_4$	g_5	g_5
u_2	1	-	$g_5+s_5g_4$	g_5	g_5
u_3	$g_5+s_5g_4$	$g_5+s_5g_4$	-	g_5	g_5
u_4	g_5	g_5	g_5	-	1
u_5	g_5	g_5	g_5	1	-

Recursive Regularisation

With recursive regularisation, we can express the influence of features on users as a parameterised function of g, and s



	u_1	u_2	u_3	u_4	u_5
u_1	-	1	$g_5+s_5g_4$	g_5	g_5
u_2	1	-	$g_5+s_5g_4$	g_5	g_5
u_3	$g_5+s_5g_4$	$g_5+s_5g_4$	-	g_5	g_5
u_4	g_5	g_5	g_5	-	1
u_5	g_5	g_5	g_5	1	-

ReMF Framework

DEFINITION 2 (**THE REMF FRAMEWORK**).

$$\min_{\substack{\mathbf{U},\mathbf{V},\\ g_p, s_p \forall F_p \in \mathcal{F}}} \mathcal{J} = \frac{1}{2}\sum_{i,j} \mathbf{O}_{ij}(\mathbf{R}_{ij} - \mathbf{U}_i \mathbf{V}_j^T)^2 + \frac{\alpha}{2}\mathbf{I}(\mathcal{F}) + \frac{\lambda}{2}(\|\mathbf{U}\|_F^2 + \|\mathbf{V}\|_F^2)$$

where α is a regularization parameter that controls the impact of recursive regularization, i.e. $\mathbf{I}(\mathcal{F})$.

ReMF Framework

DEFINITION 2 (**THE REMF FRAMEWORK**).

$$\min_{\substack{\mathbf{U},\mathbf{V},\\ g_p, s_p \forall F_p \in \mathcal{F}}} \mathcal{J} = \frac{1}{2}\sum_{i,j}\mathbf{O}_{ij}(\mathbf{R}_{ij} - \mathbf{U}_i\mathbf{V}_j^T)^2 + \frac{\alpha}{2}\mathbf{I}(\mathcal{F}) + \frac{\lambda}{2}(\|\mathbf{U}\|_F^2 + \|\mathbf{V}\|_F^2)$$

where α is a regularization parameter that controls the impact of recursive regularization, i.e. $\mathbf{I}(\mathcal{F})$.

ReMF Framework

DEFINITION 2 (THE REMF FRAMEWORK).

$$\min_{\substack{\mathbf{U},\mathbf{V},\\ g_p,s_p\forall F_p\in\mathcal{F}}} \mathcal{J} = \frac{1}{2}\sum_{i,j}\mathbf{O}_{ij}(\mathbf{R}_{ij}-\mathbf{U}_i\mathbf{V}_j^T)^2+\frac{\alpha}{2}\mathbf{I}(\mathcal{F})+\frac{\lambda}{2}(\|\mathbf{U}\|_F^2+\|\mathbf{V}\|_F^2)$$

where α is a regularization parameter that controls the impact of recursive regularization, i.e. $\mathbf{I}(\mathcal{F})$.

$$\frac{\partial\mathcal{J}}{\partial\mathbf{U}_i} = -\sum_j\mathbf{O}_{ij}(\mathbf{R}_{ij}-\mathbf{U}_i\mathbf{V}_j^T)\mathbf{V}_j+\lambda\mathbf{U}_i+\alpha\sum_{u_i,u_k\in\mathcal{U},i<k}\mathbf{C}_{ik}(\mathbf{U}_i-\mathbf{U}_k),$$

$$\frac{\partial\mathcal{J}}{\partial\mathbf{V}_j} = -\sum_i\mathbf{O}_{ij}(\mathbf{R}_{ij}-\mathbf{U}_i\mathbf{V}_j^T)\mathbf{U}_i+\lambda\mathbf{V}_j.$$

$$\frac{\partial\mathcal{J}}{\partial g_p} = \begin{cases} Dis(F_p), & \text{if } F_p \text{ is root,}\\ \prod_{\forall a:F_a\in ancestors(F_p)} s_a Dis(F_p), & \text{otherwise;}\end{cases}$$

$$\frac{\partial\mathcal{J}}{\partial s_p} = \begin{cases} \sum_{\forall F_c\in children(F_p)}\mathbf{I}(F_c), & \text{if } F_p \text{ is root,}\\ \prod_{\forall a:F_a\in ancestors(F_p)} s_a(\sum_{\forall F_c\in children(F_p)}\mathbf{I}(F_c)), & \text{otherwise.}\end{cases}$$

ReMF can automatically learn feature influence from historical user-item interaction data

ReMF Framework

DEFINITION 2 (THE REMF FRAMEWORK).

$$\min_{\substack{\mathbf{U},\mathbf{V},\\ g_p, s_p \forall F_p \in \mathcal{F}}} \mathcal{J} = \frac{1}{2}\sum_{i,j}\mathbf{O}_{ij}(\mathbf{R}_{ij} - \mathbf{U}_i\mathbf{V}_j^T)^2 + \frac{\alpha}{2}\mathbf{I}(\mathcal{F}) + \frac{\lambda}{2}(\|\mathbf{U}\|_F^2 + \|\mathbf{V}\|_F^2)$$

where α is a regularization parameter that controls the impact of recursive regularization, i.e. $\mathbf{I}(\mathcal{F})$.

$$\frac{\partial \mathcal{J}}{\partial \mathbf{U}_i} = -\sum_j \mathbf{O}_{ij}(\mathbf{R}_{ij} - \mathbf{U}_i\mathbf{V}_j^T)\mathbf{V}_j + \lambda\mathbf{U}_i + \alpha\sum_{u_i, u_k \in \mathcal{U}, i<k}\mathbf{C}_{ik}(\mathbf{U}_i - \mathbf{U}_k),$$

$$\frac{\partial \mathcal{J}}{\partial \mathbf{V}_j} = -\sum_i \mathbf{O}_{ij}(\mathbf{R}_{ij} - \mathbf{U}_i\mathbf{V}_j^T)\mathbf{U}_i + \lambda\mathbf{V}_j.$$

$$\frac{\partial \mathcal{J}}{\partial g_p} = \begin{cases} Dis(F_p), & \text{if } F_p \text{ is root,} \\ \prod_{\forall a: F_a \in ancestors(F_p)} s_a Dis(F_p), & \text{otherwise;} \end{cases}$$

$$\frac{\partial \mathcal{J}}{\partial s_p} = \begin{cases} \sum_{\forall F_c \in children(F_p)} \mathbf{I}(F_c), & \text{if } F_p \text{ is root,} \\ \prod_{\forall a: F_a \in ancestors(F_p)} s_a (\sum_{\forall F_c \in children(F_p)} \mathbf{I}(F_c)), & \text{otherwise.} \end{cases}$$

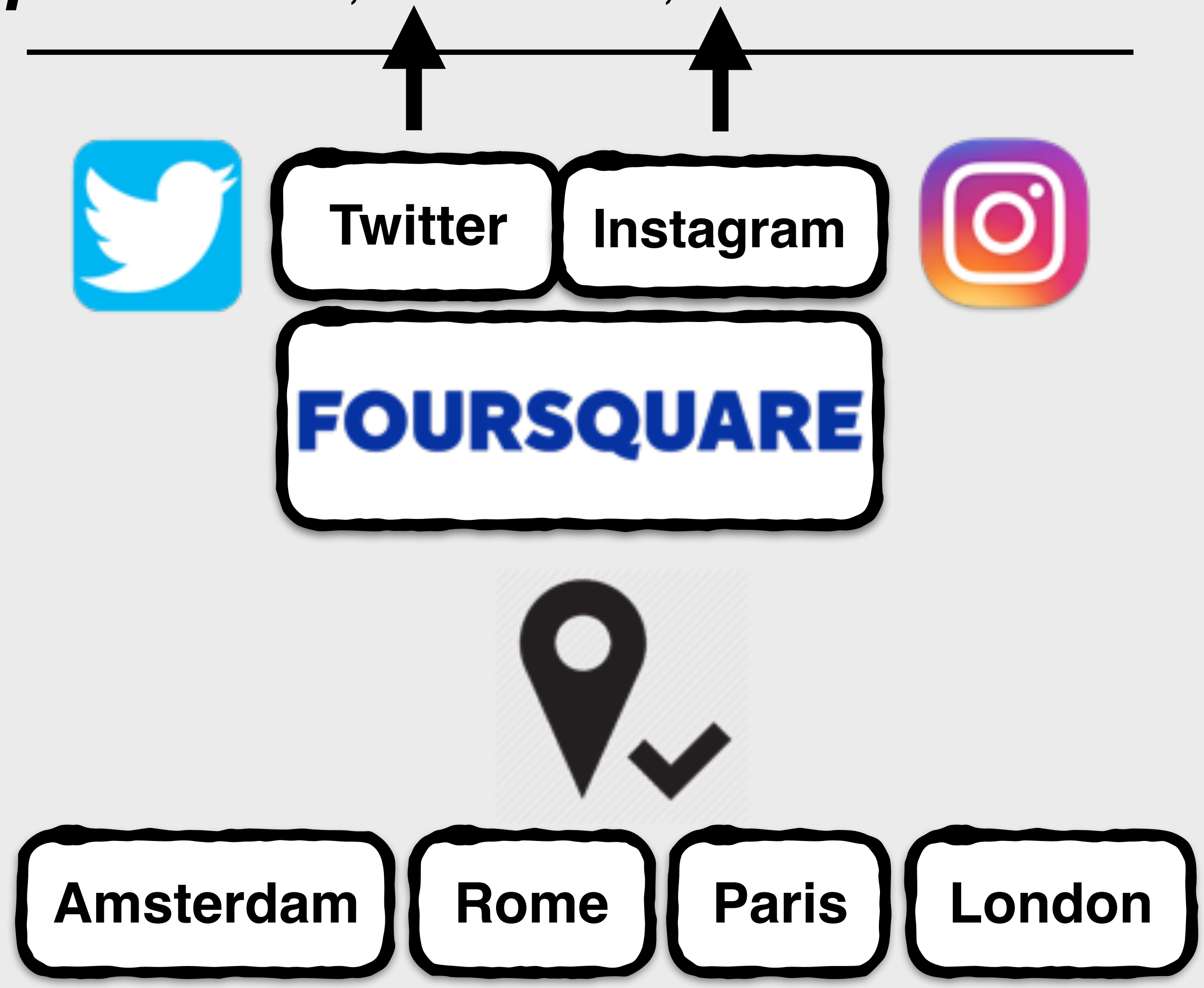
Remark: ReMF can work with feature hierarchies of *both* users and items; can work with *imbalanced* feature hierarchies; and it is *scalable* to large dataset

Validation

Setup

Datasets

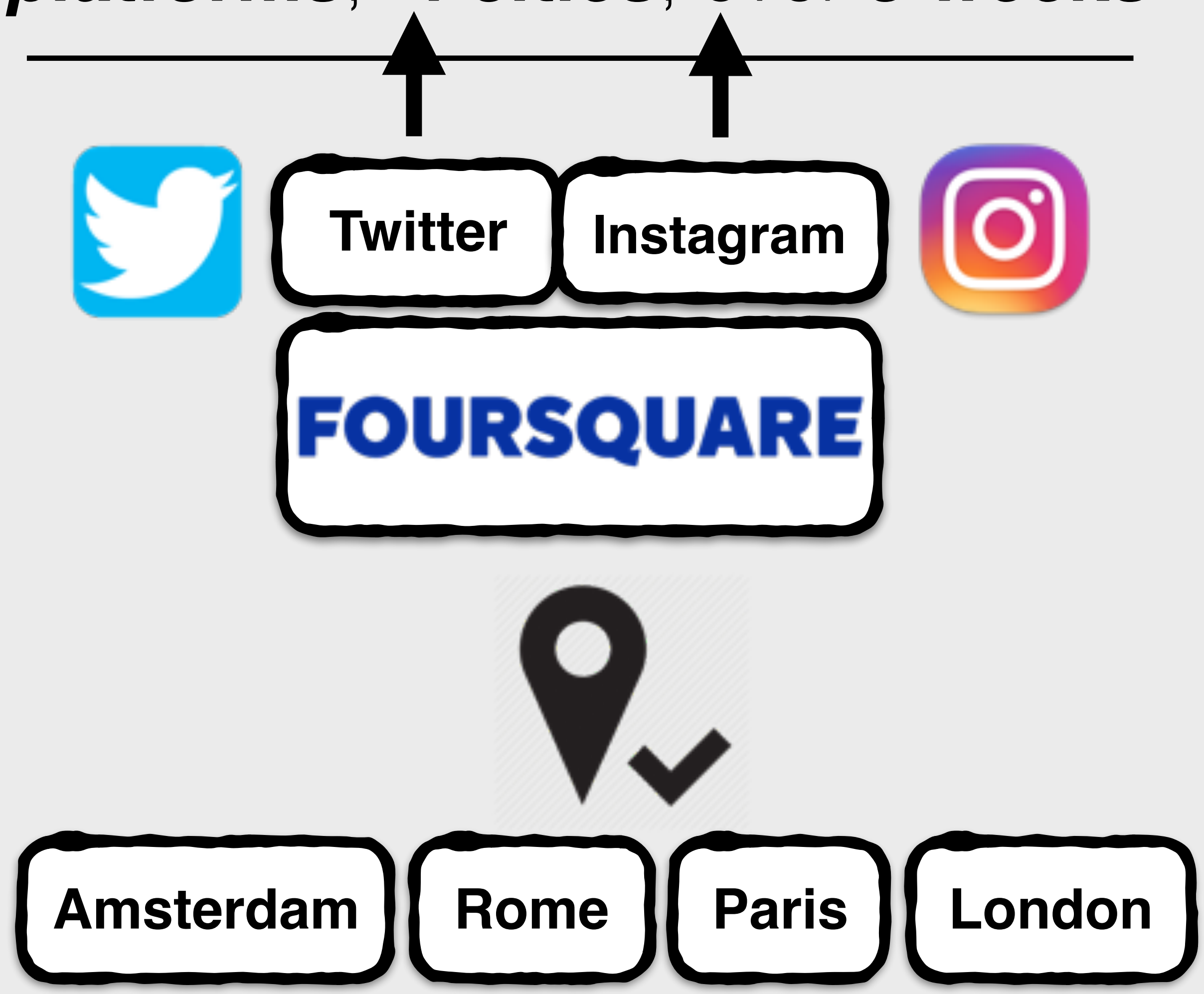
Users' foursquare check-ins in ***2 platforms***, ***4 cities***, over ***3 weeks***



Setup

Datasets

Users' foursquare check-ins in ***2 platforms***, ***4 cities***, over ***3 weeks***



Evaluation

MAE, RMSE, AUC
5-fold cross-validation

Setup

Datasets

Users' foursquare check-ins in ***2 platforms***, ***4 cities***, over ***3 weeks***

Twitter
Instagram

FOURSQUARE

Amsterdam
Rome
Paris
London

Evaluation

MAE, RMSE, AUC
5-fold cross-validation

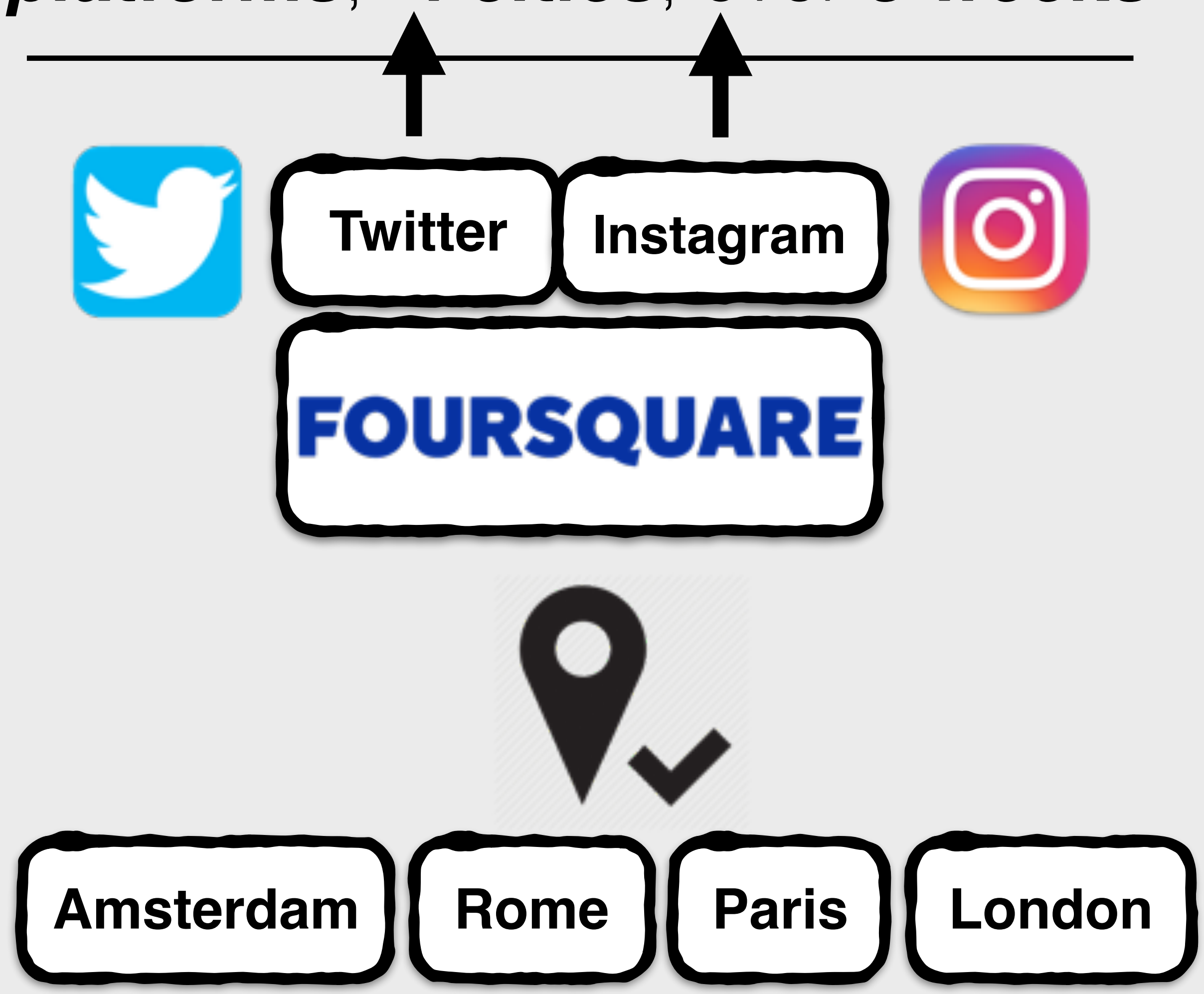
Parameter setting

Grid Search

Setup

Datasets

Users' foursquare check-ins in ***2 platforms***, ***4 cities***, over ***3 weeks***



Evaluation

MAE, RMSE, AUC
5-fold cross-validation

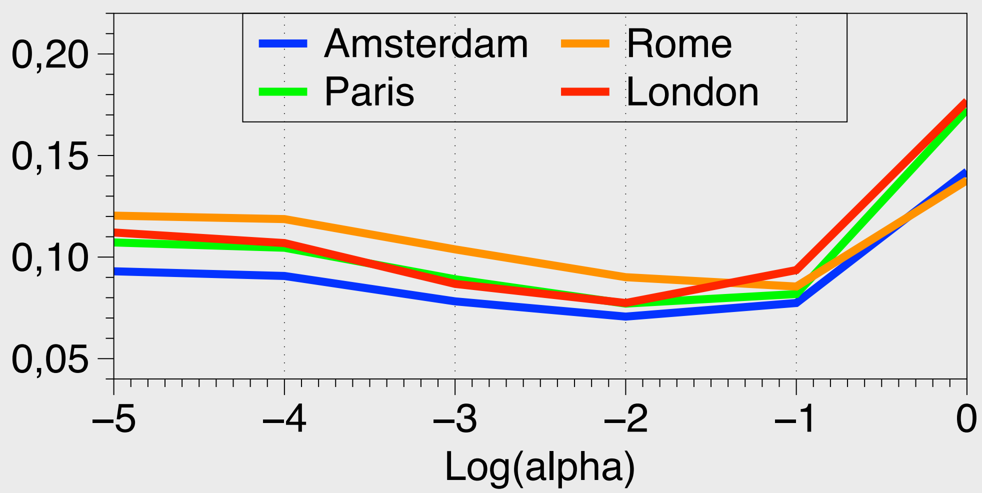
Parameter setting

Grid Search

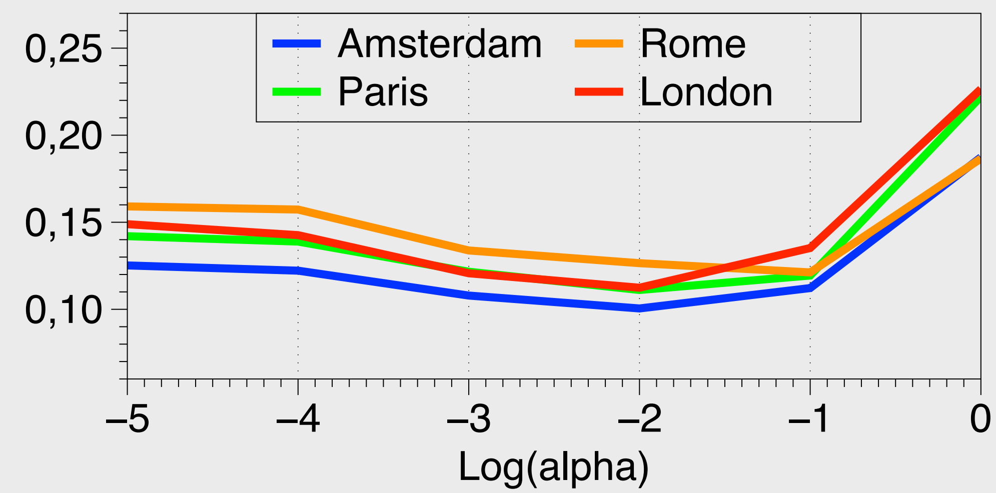
Comparison methods

MF
CMF, FM
(generic feature based methods)
TaxMF, ***HieFM***
(w. hierarchical features)

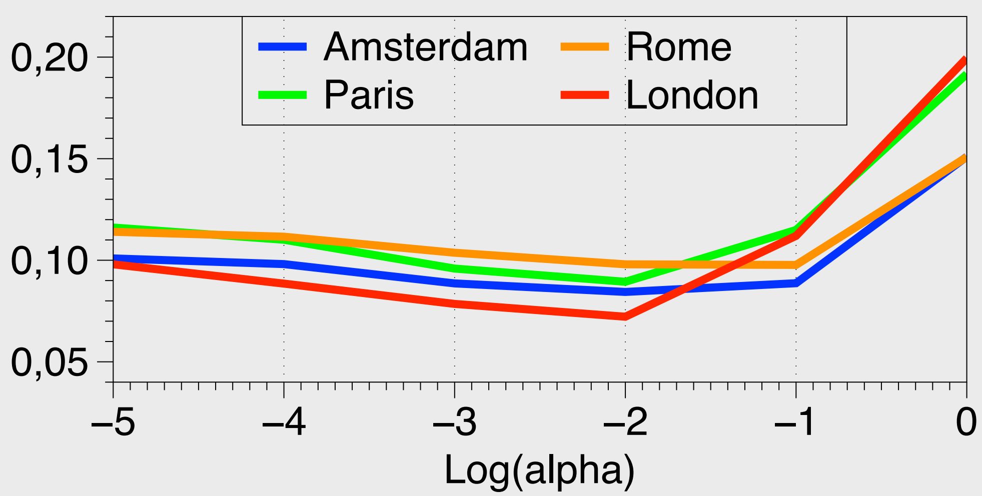
Results of ReMF



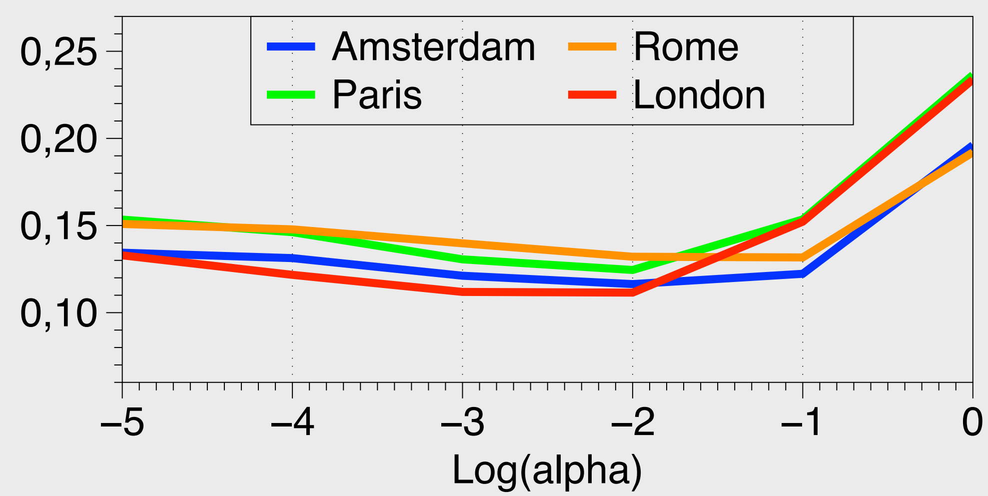
MAE - Instagram



RMSE - Instagram

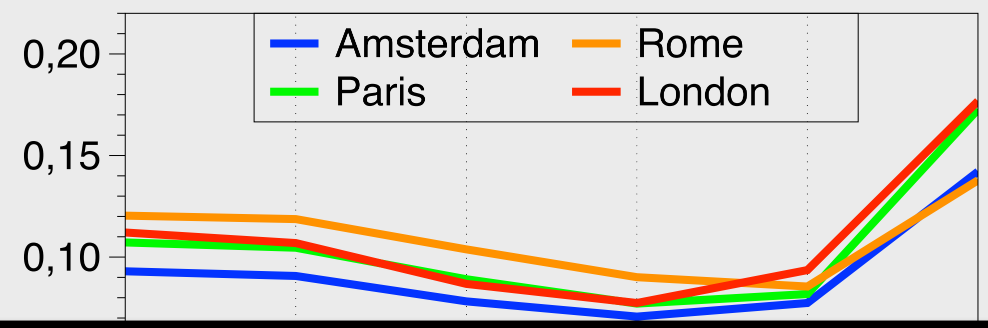


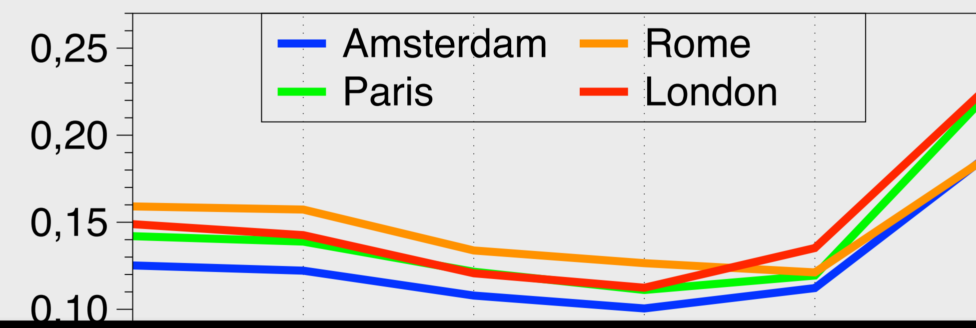
MAE - Twitter



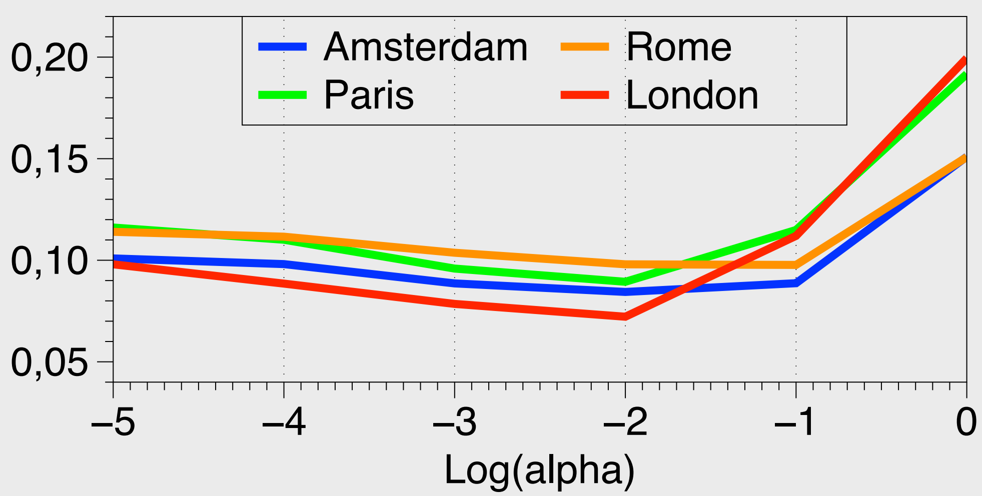
RMSE - Twitter

Results of ReMF

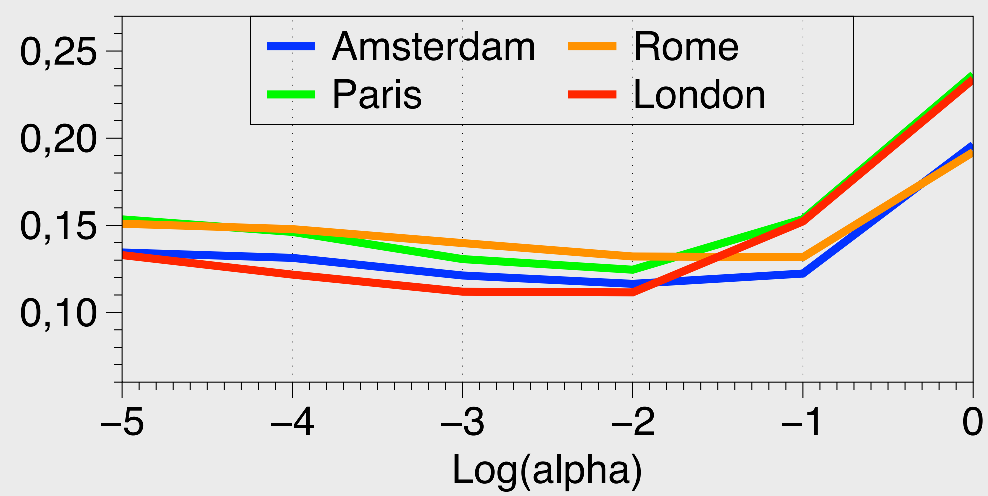




ReMF is robust to parameter setting.



MAE - Twitter



RMSE - Twitter

Comparative Results

MAE (RMSE in paper)

two views of each dataset

Views	Dataset		MF	CMF	TaxMF	FM	HieFM	ReMF
All	Inst.	Amsterdam	.1957	.1564	.1426	.0876	.0822	**.0707**
		Paris	.1539	.1550	.1416	.0790	.0830	**.0772**
		Rome	.2549	.1584	.1474	.0912	.0885	**.0855**
		London	.1799	.1559	.1369	.0834	.0851	**.0774**
	Tw.	Amsterdam	.2264	.1606	.1345	.0989	.0942	**.0844**
		Paris	.2014	.1714	.1552	.0956	.0935	**.0894**
		Rome	.2681	.1713	.1591	.1023	.0996	**.0977**
		London	.2176	.1659	.1545	.0931	.0898	**.0772**
Cold Start	Inst.	Amsterdam	.2938	.1552	.1457	.0924	.0885	**.0712**
		Paris	.1939	.1541	.1476	.0849	.0848	**.0799**
		Rome	.3840	.1614	.1518	.0952	.0938	**.0808**
		London	.3032	.1544	.1415	.0893	.0901	**.0791**
	Tw.	Amsterdam	.3261	.1604	.1426	.1006	.0956	**.0849**
		Paris	.2439	.1706	.1640	.1012	.0945	**.0873**
		Rome	.3922	.1718	.1681	.1073	.1070	**.0988**
		London	.3301	.1642	.1587	.0967	.0924	**.0756**

Comparative Results

MAE (RMSE in paper)

two views of each dataset

7.2% (all)

12.02% (cold)

Views	Dataset		MF	CMF	TaxMF	FM	HieFM	ReMF
All	Inst.	Amsterdam	.1957	.1564	.1426	.0876	.0822	**.0707**
		Paris	.1539	.1550	.1416	.0790	.0830	**.0772**
		Rome	.2549	.1584	.1474	.0912	.0885	**.0855**
		London	.1799	.1559	.1369	.0834	.0851	**.0774**
	Tw.	Amsterdam	.2264	.1606	.1345	.0989	.0942	**.0844**
		Paris	.2014	.1714	.1552	.0956	.0935	**.0894**
		Rome	.2681	.1713	.1591	.1023	.0996	**.0977**
		London	.2176	.1659	.1545	.0931	.0898	**.0772**
Cold Start	Inst.	Amsterdam	.2938	.1552	.1457	.0924	.0885	**.0712**
		Paris	.1939	.1541	.1476	.0849	.0848	**.0799**
		Rome	.3840	.1614	.1518	.0952	.0938	**.0808**
		London	.3032	.1544	.1415	.0893	.0901	**.0791**
	Tw.	Amsterdam	.3261	.1604	.1426	.1006	.0956	**.0849**
		Paris	.2439	.1706	.1640	.1012	.0945	**.0873**
		Rome	.3922	.1718	.1681	.1073	.1070	**.0988**
		London	.3301	.1642	.1587	.0967	.0924	**.0756**

Comparative Results

MAE (RMSE in paper)

two views of each dataset

↑ 7.2% (all)

12.02% (cold)

Views	Dataset		MF	CMF	TaxMF	FM	HieFM	ReMF
All	Inst.	Amsterdam	.1957	.1564	.1426	.0876	.0822	**.0707**
		Paris	.1539	.1550	.1416	.0790	.0830	**.0772**
		Rome	.2549	.1584	.1474	.0912	.0885	**.0855**
		London	.1799	.1559	.1369	.0834	.0851	**.0774**
	Tw.	Amsterdam	.2264	.1606	.1345	.0989	.0942	**.0844**
		Paris	.2014	.1714	.1552	.0956	.0935	**.0894**
		Rome	.2681	.1713	.1591	.1023	.0996	**.0977**
		London	.2176	.1659	.1545	.0931	.0898	**.0772**
	Inst.	Amsterdam	.2938	.1552	.1457	.0924	.0885	**.0712**
		Paris	.1939	.1541	.1476	.0849	.0848	**.0799**
	Tw.	Paris	.2439	.1706	.1640	.1012	.0945	**.0873**
		Rome	.3922	.1718	.1681	.1073	.1070	**.0988**
		London	.3301	.1642	.1587	.0967	.0924	**.0756**

ReMF consistently outperform the comparative methods.

Comparative Results

MAE (RMSE in paper)

two views of each dataset

7.2% (all)

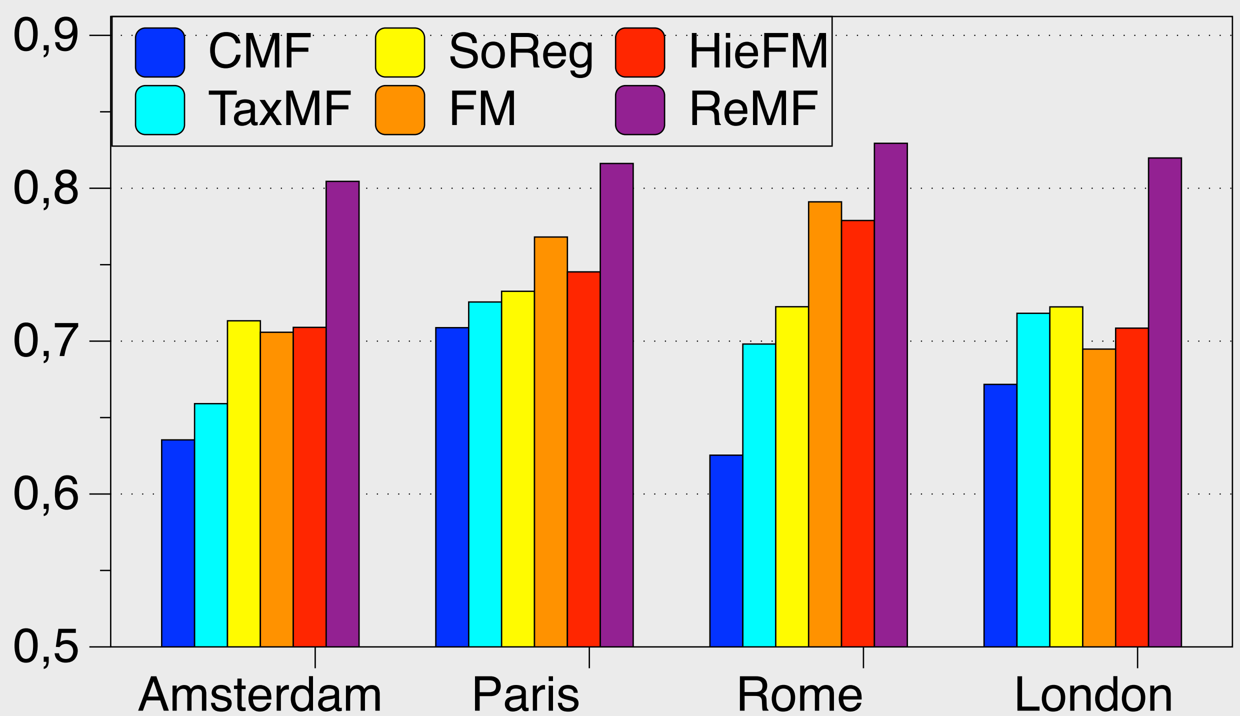
12.02% (cold)

Views	Dataset		MF	CMF	TaxMF	FM	HieFM	ReMF
All	Inst.	Amsterdam	.1957	.1564	.1426	.0876	.0822	**.0707**
		Paris	.1539	.1550	.1416	.0790	.0830	**.0772**
		Rome	.2549	.1584	.1474	.0912	.0885	**.0855**
		London	.1799	.1559	.1369	.0834	.0851	**.0774**
	Tw.	Amsterdam	.2264	.1606	.1345	.0989	.0942	**.0844**
		Paris	.2014	.1714	.1552	.0956	.0935	**.0894**
		Rome	.2681	.1713	.1591	.1023	.0996	**.0977**
		London	.2176	.1659	.1545	.0931	.0898	**.0772**
	Inst.	Amsterdam	.2938	.1552	.1457	.0924	.0885	**.0712**
		Paris	.1939	.1541	.1476	.0849	.0848	**.0799**
	Tw.	Rome	.3922	.1718	.1681	.1073	.1070	**.0988**
		London	.3301	.1642	.1587	.0967	.0924	**.0756**

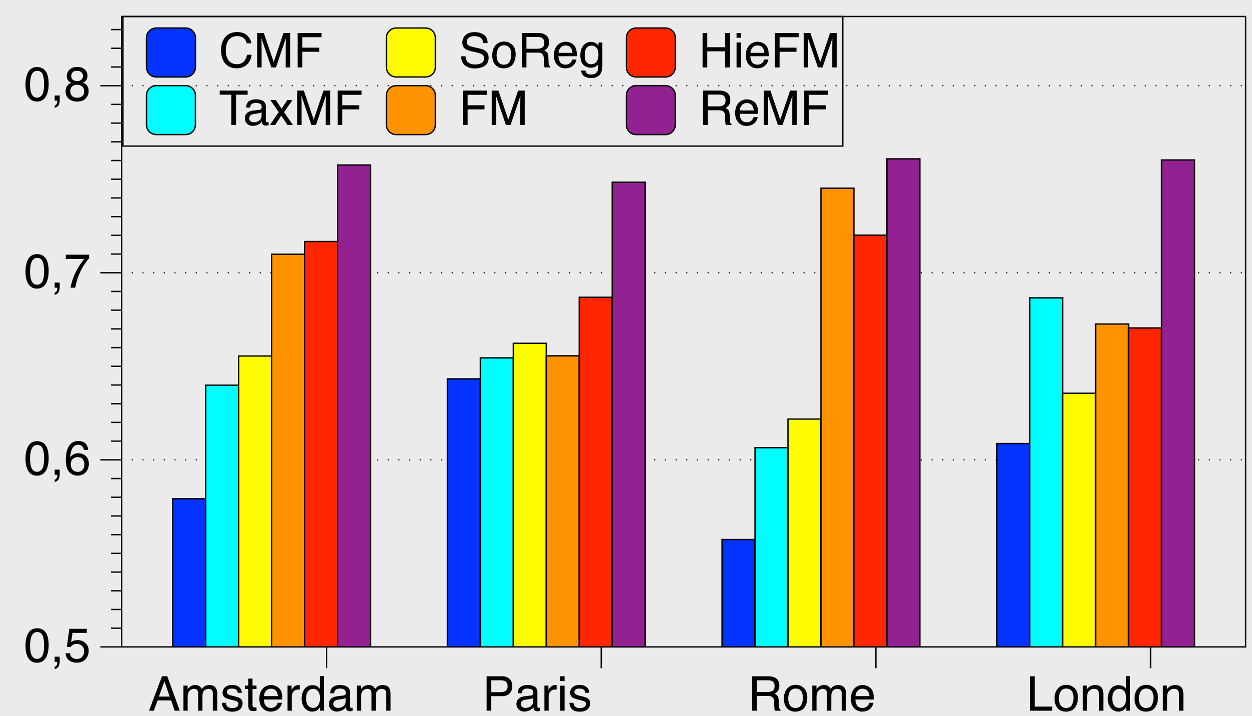
ReMF achieves higher performance in coping with the cold start problem compared to the state-of-the-art methods.

Comparative Results

AUC



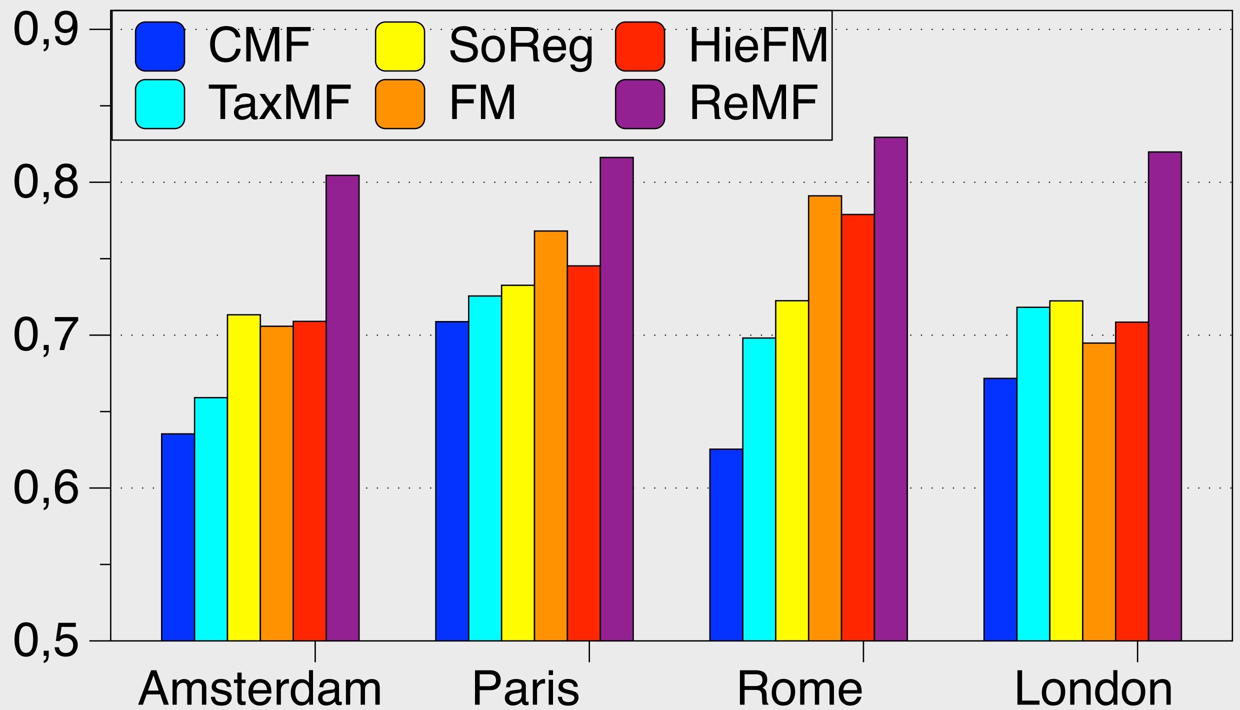
Instagram



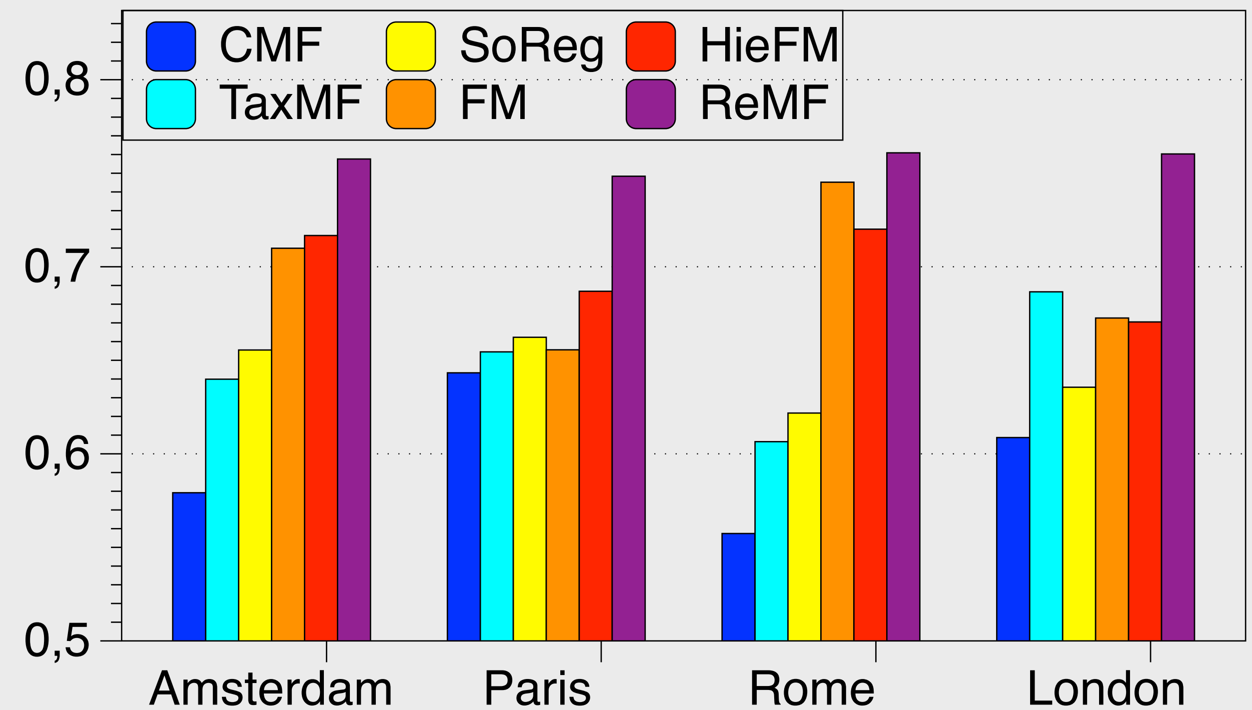
Twitter

Comparative Results

AUC



Instagram



Twitter

ReMF consistently outperform the comparative methods in terms of ranking performance.

Take away

Recursive Regularisation, as a *parameterised* regularisation function, can better exploit feature hierarchy for recommendation by *learning structured feature influence* from data

Thank you!

j.yang-3@tudelft.nl

TU Delft

NANYANG TECHNOLOGICAL UNIVERSITY