SIEMENS

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

ESWC 2016 / Machine Learning Track *01.06.2016*

Embedding Mapping Approaches
for Tensor Factorization and Knowledge Graph Modelling

Yinchong Yang[1,2], *Cristóbal Esteban*[1,2] *and Volker Tresp*[1,2]

[1] Siemens AG, Corporate Technology, Munich, Germany
[2] Ludwig-Maximilians-Universität München, Munich, Germany

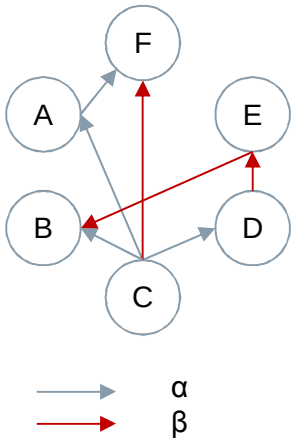
Agenda

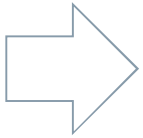
- ***Introduction*** *to Latent Embedding Models*
- ***Motivation*** *to Embedding Mapping*
- ***Embedding Mapping Approaches*** *in Details*
- ***Experiments***
- ***Conclusions***

Introduction to Latent Embedding Models

The Starting Point: From Graph to Tensor

- Knowledge Graph represented as adjacency matrix/tensor $\underline{\boldsymbol{Y}}$:
- $y_{i,j,k} = 1$ for the triple (entity i, relation k, entity j) observed; and 0 otherwise;





α	A	B	C	D	E	F
A	0	0	0	0	0	1
B	0	0	0	0	0	0
C	1	1	0	1	0	0
D	0	0	0	0	0	0
E	0	0	0	0	0	0
F	0	0	0	0	0	0

β	A	B	C	D	E	F
A	0	0	0	0	0	0
B	0	0	0	0	0	0
C	0	0	0	0	0	1
D	0	0	0	0	1	0
E	0	1	0	0	0	0
F	0	0	0	0	0	0

$\underline{\boldsymbol{Y}}$ of dimensions (6 x 6 x 2)

- 'Feature space' being usually highly dimensional and highly sparse;

Introduction to Latent Embedding Models

Modelling of Adjacency Tensor

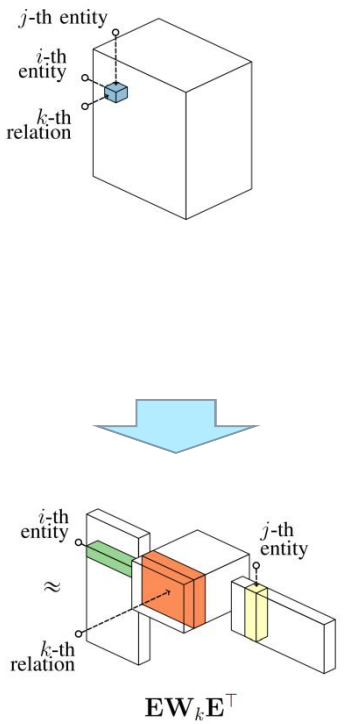
"Latent embedding models" aka. "factorization models" can model the KG-Tensors by

- Learning a latent embedding vector/matrix for each entity/relation;
- Learning an interaction rule of such vectors/matrices;

- And fulfil therefore the tasks i.a. of:
  - Prediction of unobserved links;
  - Distance-based entity resolution;

- Representative models: PARAFAC, Tucker, RESCAL, Multiway Neural Network, etc.;
- Successful applications in e.g.:
  - Recommender Systems [Koren et al. 2009] (matrices);
  - Knowledge Graph [Nickel et al. 2015, Dong et al. 2014] (tensors);

The RESCAL model [Nickel et al. 2011, 2015]

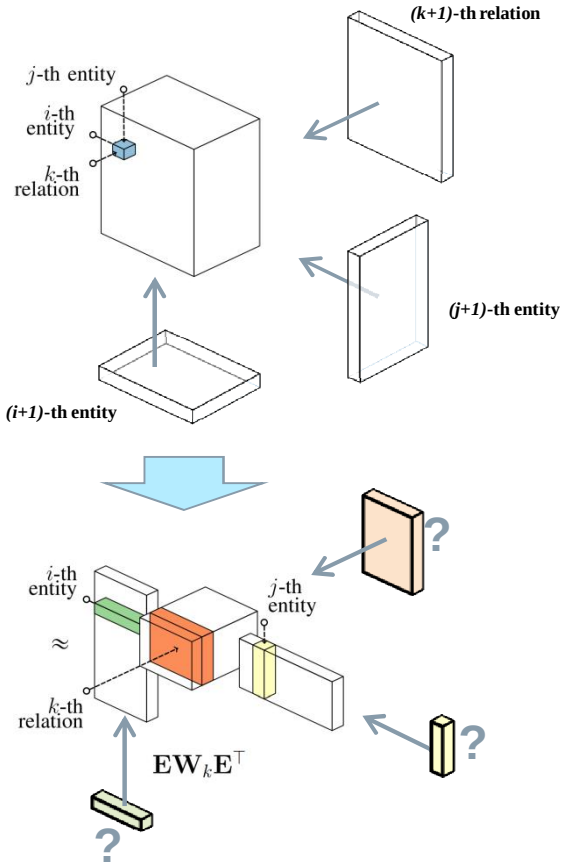


Motivation to Embedding Mapping

Embedding Mapping to handle new entities

- One limitation: deriving embeddings for new entities/relations (→):
  - Current known solution:
    - Retraining the entire model → Not efficient;
    - KNN approximation → Not effective;
- We introduce a Embedding Mapping ('Emma')
  - from the feature space (i.e. the set all known links between an entity and all other entities)
  - to the latent embedding space;

The RESCAL model [Nickel et al. 2011, 2015]



Approach 0:
Factorization Models with Closed-Form Mappings

Canonical solutions

- There are a few factorization models where such mapping are for free as inverting. For instance:
  - SVD: $X = UDV^T \rightarrow U = X(DV^T)^\dagger$ therefore generally assumed: $u_{new}^T = x_{new}^T(DV^T)^\dagger$
  - Tucker: $\mathcal{X} \approx \mathcal{G} \times_1 A \times_2 B \times_3 C$ i.e. $X_{(1)} = AG_{(1)}(C \otimes B)^T$ analogously: $a_{new}^T = x_{new}^T(G_{(1)}(C \otimes B)^T)^\dagger$

- What these models have in common:
  - Existence of matrix M_d independent of latent embeddings $A^{(d)}$ for each dimension d so that:
    - $X_{(d)} = A^{(d)} (M_d)^\dagger$ i.e. $A^{(d)} = X_{(d)} M_d$

- Other models do not enjoy such advantages:
  - RESCAL: shared embedding;
  - Multiway Neural Network (mwNN): non-linearity

Approach 1: Post Mapping

The most intuitive thought – "Emma-Post"

- Step 1: Train the factorization model as usual;
- Step 2: Calculate a linear regression

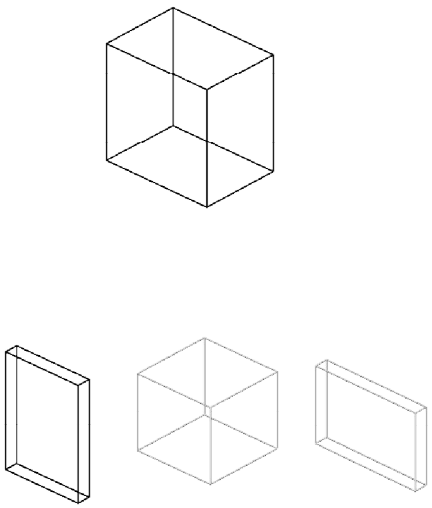
$$\boldsymbol{a}_i^{(d)} = \boldsymbol{W}^{(d)T} vec(\boldsymbol{X}_i) + \boldsymbol{b}^{(d)},$$

  where $\boldsymbol{a}_i^{(d)}$ is the embedding vector and $\boldsymbol{X}_i$ is the i-th slice of the d-th dimension of the tensor $\underline{\boldsymbol{X}}$.
- So that the same regression relation generalizes to any new entity:

$$\boldsymbol{a}_{new}^{(d)} = \boldsymbol{W}^{(d)T} vec(\boldsymbol{X}_{new}) + \boldsymbol{b}^{(d)}$$

- Equivalence to inverting in cases of factorization models with closed-form mapping (details in paper);

Step 1:



Approach 1: Post Mapping

The most intuitive thought – "Emma-Post"

- Step 1: Train the factorization model as usual;
- Step 2: Calculate a linear regression

$$\boldsymbol{a}_i^{(d)} = \boldsymbol{W}^{(d)T} vec(\boldsymbol{X}_i) + \boldsymbol{b}^{(d)},$$

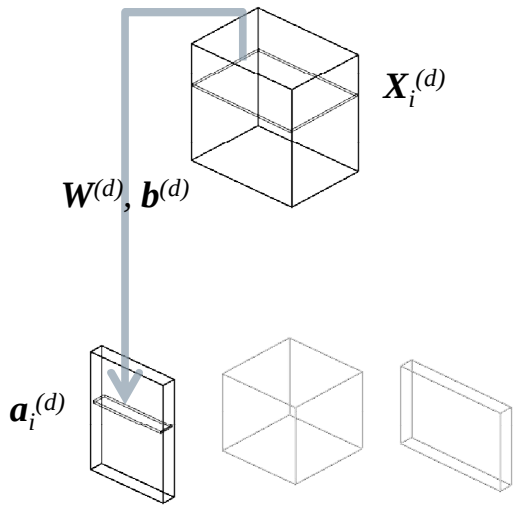
where $\boldsymbol{a}_i^{(d)}$ is the embedding vector and $\boldsymbol{X}_i$ is the i-th slice of the d-th dimension of the tensor $\underline{\boldsymbol{X}}$.

- So that the same regression relation generalizes to any new entity:

$$\boldsymbol{a}_{new}^{(d)} = \boldsymbol{W}^{(d)T} vec(\boldsymbol{X}_{new}) + \boldsymbol{b}^{(d)}$$

- Equivalence to inverting in cases of factorization models with closed-form mapping (details in paper);

Step 2:



Approach 1: Post Mapping

The most intuitive thought – "Emma-Post"

- Step 1: Train the factorization model as usual;
- Step 2: Calculate a linear regression

$$a_i^{(d)} = W^{(d)T} vec(X_i) + b^{(d)},$$

where $a_i^{(d)}$ is the embedding vector and X_i is the i-th slice of the d-th dimension of the tensor $\underline{X}$.

- So that the same regression relation generalizes to any new entity:

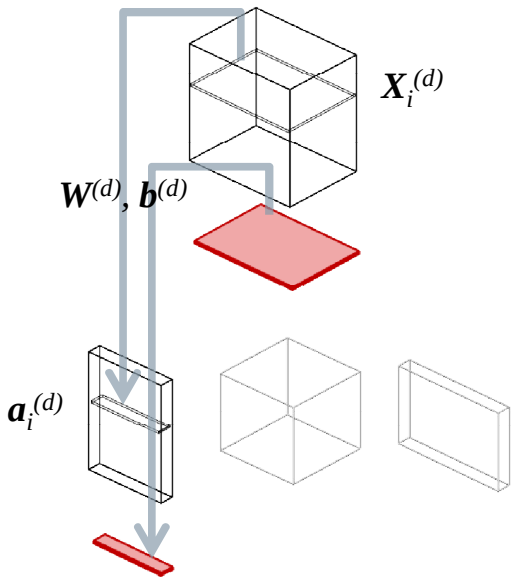
$$a_{new}^{(d)} = W^{(d)T} vec(X_{new}) + b^{(d)}$$

- Equivalence to inverting in cases of factorization models with closed-form mapping (details in paper);

Reducing mapping error function after factorization error;

+ Simple, applicable for arbitrary factorization model;
- Regression error reduced only to a certain extent

Generalization:



Approach 2: Hatting Algorithm

Integrating the linear regression into the factorization – "Emma-Hatting"

- Calculate the linear regression after each iteration (ALS, gradient approaches) of the factorization learning (instead of only once after factorization completes);
- → Replacing the latent embedding vector with its Least-Square estimates (therefore 'Hatting') in each iteration:

for each iteration of the factorization:

  for each dimension d of target tensor $\underline{\boldsymbol{X}}$:

    calculate the latent embedding $\boldsymbol{A}^{(d)}$ as prescribed by the factorization model

    replace $\boldsymbol{A}^{(d)}$ with its LS estimate w.r.t. $\boldsymbol{X}_{(d)}$ as regressor:

    $\boldsymbol{A}^{(d)} \leftarrow \hat{\boldsymbol{A}}_{\boldsymbol{LS}}^{(d)} = \boldsymbol{X}_{(d)} (\boldsymbol{X}_{(d)}^{T} \boldsymbol{X}_{(d)} - \lambda \boldsymbol{I})^{-1} \boldsymbol{X}_{(d)}^{T} \boldsymbol{A}^{(d)}$

return $\boldsymbol{A}^{(d)}$ as latent embeddings;

  $\boldsymbol{M}_d = (\boldsymbol{X}_{(d)}^{T} \boldsymbol{X}_{(d)})^{-1} \boldsymbol{X}_{(d)}^{T} \boldsymbol{A}^{(d)}$ as mapping matrices for each d

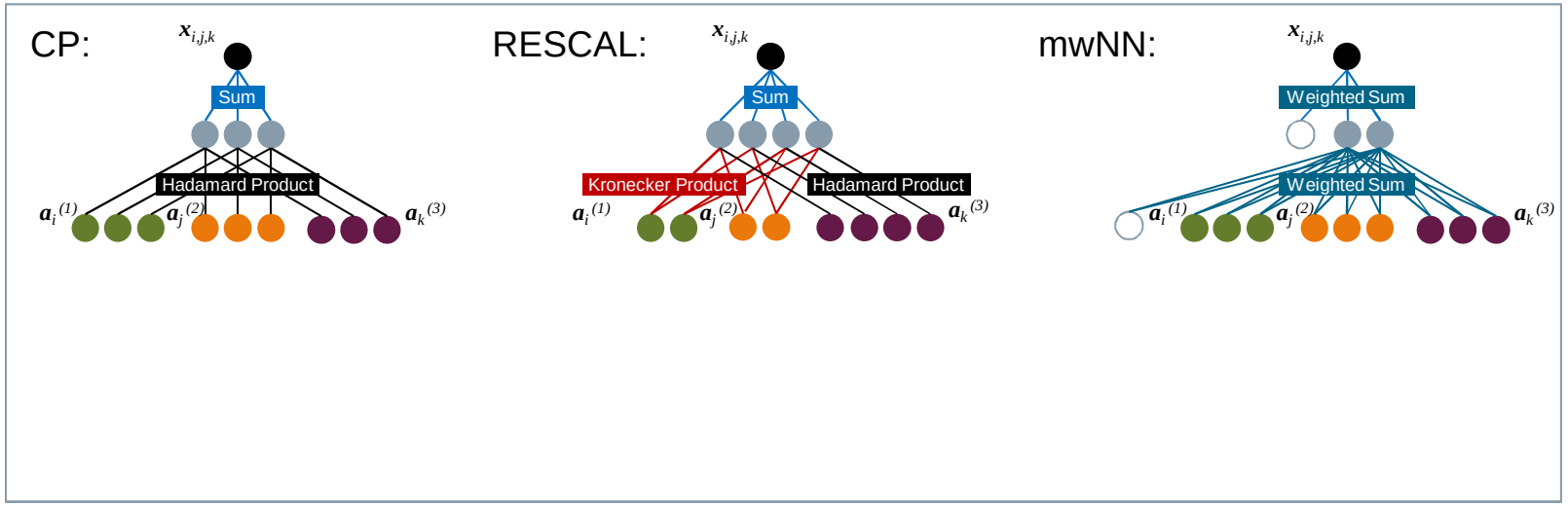
Reducing the two errors of factorization and mapping alternatively;

+ The regression error almost eliminated;
+ Though applied in each iteration, the update consists of only one matrix multiplication;
- But how does the Hatting update affects the gradient approaches?

Approach 3: Back-Propagation

Inspired by an Neural Network aspect – "Emma-BP"

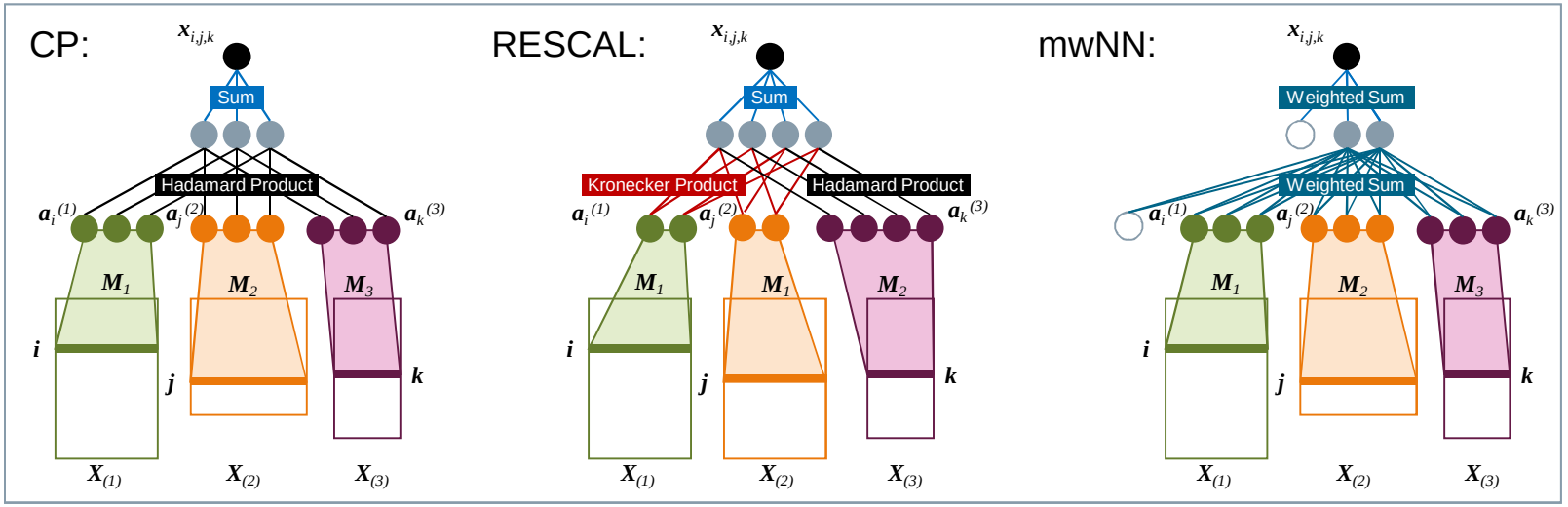
- Most factorization models rewritable as Neural Networks:
- 1) input layer = latent embeddings; 2) output = tensor entry; 3) architecture = the interaction rule:
  - $\boldsymbol{x}_{i,j,k} = f_{NN}(\boldsymbol{a}_i^{(1)}, \boldsymbol{a}_j^{(2)}, \boldsymbol{a}_k^{(3)})$



Approach 3: Back-Propagation

Inspired by an Neural Network aspect – "Emma-BP"

- Most factorization models rewritable as Neural Networks:
- 1) input layer = latent embeddings; 2) output = tensor entry; 3) architecture = the interaction rule:
  - $x_{i,j,k} = f_{NN}(a_i^{(1)}, a_j^{(2)}, a_k^{(3)})$
- Emma as one more linear layer → train factorization + Emma models as a compact one with BP
  - $x_{i,j,k} = h_{NN}((x_{(1)})_i , (x_{(2)})_j , (x_{(3)})_k)$

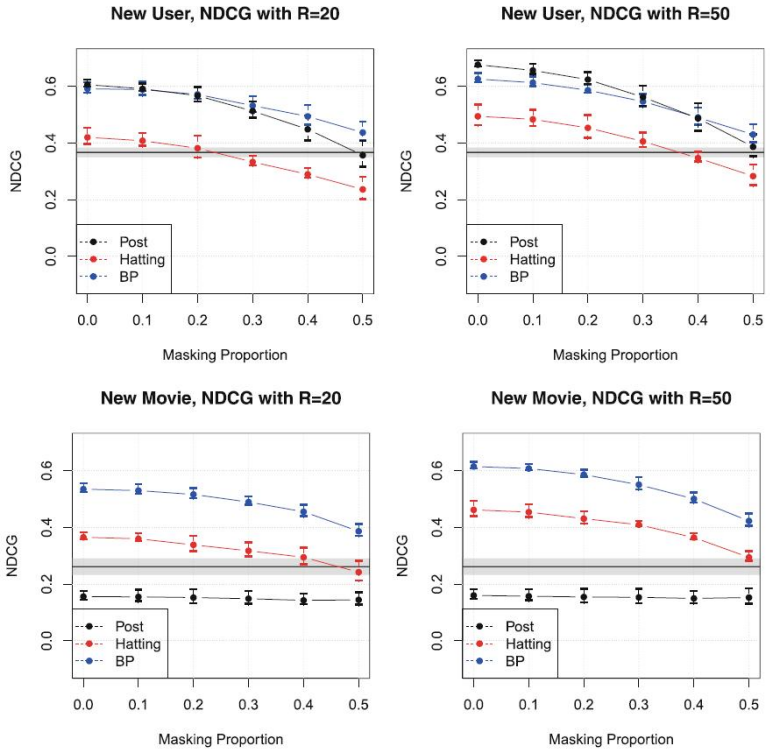


Reducing one compact cost function;
+ Guarantees a perfect mapping i.e. without mapping error;
- Numerically more expensive;

Experiment 1: MovieLens Data

Emma applied for matrix factorization and Collaborate Filtering

- **Data**: a binary user-item matrix (943 x 1682);
- **Task**: make movie recommendations to new users and user recommendations to new movies;
- **Settings**:
  - Baseline: Most-frequent Prediction;
  - Sampling 20% users/movies as test set;
  - Masking a proportion of [0%, 10%, 20%, 30%, 40%, 50%] of all entries, simulating incomplete information;
  - Mapping the test set into the latent space and calculate recommendations (recovering the masked ones)
  - Evaluation in term of NDCG@k
- **Models**: Matrix Factorization according to [Koren et al. 2009] x all 3 Emma approaches
- **Results**:
  - New users: Emma-Post, Emma-BP > Emma-Hatting
  - New movies: Emma-BP > Emma-Hatting > Emma-Post



Use case: Real-time recommendation for new user/item
Further reference: Rendle's MyMediaLite package provides NDCG's between 0.56 and 0.62, when generating recommendations for known users.*

*

http://www.mymedialite.net/examples/item_recommendation_datasets.html

Experiment 2: FreeBase Data

Emma applied for tensor factorization and KG link prediction

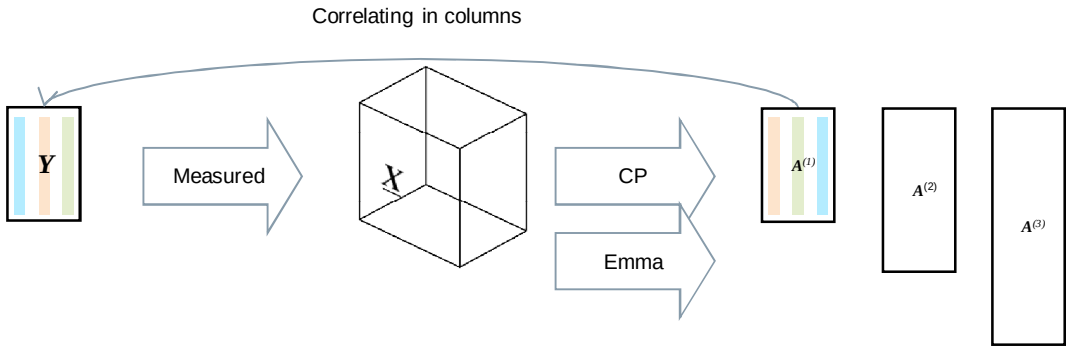
- **Data**: A sampled fraction of KG, represented as a binary tensor (39 x 115 x 115);
- **Task**: Predicting links for new entities
- **Settings**: 'Baseline': Retraining;
  - Sampling 20% of all entities and masking 20% links;
  - Mapping the entities into the latent space and to predict the masked links;
  - Evaluation in term of AUROC, AUPRC
- **Models**: RESCAL, mwNN in combination with all 3 Emma approaches
- **Results**:
  - mwNN outperform RESCAL, confirming [Krompaß et al. 2015];
  - Emma-Hatting and Emma-BP superior to Emma-Post;
  - The combination of mwNN and Emma-BP show performances close even to those from retraining;

Fac.	Mapping	R = 10		R = 30	
		AUROC	AUPRC	AUROC	AUPRC
RESCAL	Retraining	0.901 ± 0.039	0.820 ± 0.059	0.788 ± 0.054	0.616 ± 0.100
	Emma-Post	0.759 ± 0.096	0.600 ± 0.145	$0.693 \pm \mathbf{0.065}$	$0.432 \pm \mathbf{0.106}$
	Emma-Hatting	$\mathbf{0.778} \pm 0.112$	$\mathbf{0.605} \pm 0.152$	0.700 ± 0.091	0.481 ± 0.120
	Emma-BP	$0.700 \pm \mathbf{0.060}$	$0.485 \pm \mathbf{0.092}$	$\mathbf{0.740} \pm 0.090$	$\mathbf{0.509} \pm 0.134$
mwNN	Retraining	0.964 ± 0.008	0.886 ± 0.060	0.970 ± 0.010	0.923 ± 0.017
	Emma-Post	$0.844 \pm \mathbf{0.009}$	0.390 ± 0.101	$0.826 \pm \mathbf{0.035}$	$0.382 \pm \mathbf{0.063}$
	Emma-Hatting	0.847 ± 0.042	0.423 ± 0.118	0.843 ± 0.038	0.394 ± 0.081
	Emma-BP	$\mathbf{0.949} \pm 0.022$	$\mathbf{0.805} \pm \mathbf{0.080}$	$\mathbf{0.931} \pm 0.036$	$\mathbf{0.735} \pm 0.101$

Experiment 3: Amino Acid Data

Verifying Emma's plausibility and interpretability of the latent space

- **Data**:
  - 3 types of amino acid mixed according to 5 recipes: a real matrix of (5 x 3) $\boldsymbol{Y}$;
  - The 5 samples measured by fluorescence with excitation 250-300nm, emission 250-450nm: a real tensor of (5 x 61 x 201) $\underline{\underline{\boldsymbol{X}}}$;
  - With a rank 3 CP factorization, the latent embeddings representing recipes ($\boldsymbol{A}^{(1)}$) are expected to correlate column-wise perfectly with the recipe matrix itself (whilst the column order may vary).

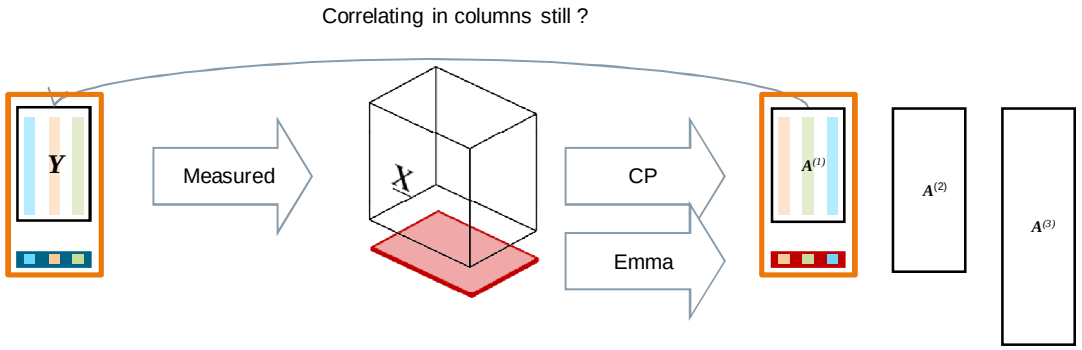


- **Task**:
  - What if there is a new measurement matrix (61 x 201) of a mixture according to a new recipe? Will the correspondingly mapped embedding vector still correlate as before?
- **Settings**: Leave-1-out and Leave-2-out, evaluation in term of Pearson Correlation
- **Model**: CP in combination with Emma-Post and Emma-Hatting
- **Results**: Leave-1-out: both 0.999; Leave-2-out: both 0.991;

Experiment 3: Amino Acid Data

Verifying Emma's plausibility and interpretability of the latent space

- **Data**:
  - 3 types of amino acid mixed according to 5 recipes: a real matrix of (5 x 3) $\boldsymbol{Y}$;
  - The 5 samples measured by fluorescence with excitation 250-300nm, emission 250-450nm: a real tensor of (5 x 61 x 201) $\underline{\underline{\boldsymbol{X}}}$;
  - With a rank 3 CP factorization, the latent embeddings representing recipes ($\boldsymbol{A}^{(1)}$) are expected to correlate column-wise perfectly with the recipe matrix itself (whilst the column order may vary).



- **Task**:
  - What if there is a new measurement matrix (61 x 201) of a mixture according to a new recipe? Will the correspondingly mapped embedding vector still correlate as before?
- **Settings**: Leave-1-out and Leave-2-out, evaluation in term of Pearson Correlation
- **Model**: CP in combination with Emma-Post and Emma-Hatting
- **Results**: Leave-1-out: both 0.999; Leave-2-out: both 0.991;

Conclusions and Outlooks

- Embedding Mapping Approaches derive latent embeddings for new entities in real time;
- Two classes of mapping:
  a) Explicit mapping that are learned
     - after the factorization (Emma-Post);
     - during the factorization (Emma-Hatting);
  b) Implicit mapping that are learned
     - as one linear activated layer of specific NN model (Emma-BP);
     - as closed-form mapping (inverting) in case of certain models;
- Experiments
  a) To predict new links in recommender system and knowledge graph;
  b) To verify interpretability of mapped embeddings;

- Future works
  a) Improve the scalability by attacking the bottleneck due to unfolding operation;
  b) Combining the work of [Gantner et al., 2010]: two-fold mapping: content attributes → feature space → latent space

(Selected) References

- Nickel, Maximilian, Volker Tresp, and Hans-Peter Kriegel. "A three-way model for collective learning on multi-relational data." *Proceedings of the 28th international conference on machine learning (ICML-11)*. 2011.
- Nickel, Maximilian, et al. "A review of relational machine learning for knowledge graphs: From multi-relational link prediction to automated knowledge graph construction." *arXiv preprint arXiv:1503.00759* (2015).
- Krompaß, Denis, Stephan Baier, and Volker Tresp. "Type-constrained representation learning in knowledge graphs." *The Semantic Web-ISWC 2015*. Springer International Publishing, 2015. 640-655.
- Koren, Yehuda, Robert Bell, and Chris Volinsky. "Matrix factorization techniques for recommender systems." *Computer* 8 (2009): 30-37.
- Gantner, Zeno, et al. "Learning attribute-to-feature mappings for cold-start recommendations." *Data Mining (ICDM), 2010 IEEE 10th International Conference on*. IEEE, 2010.
- Dong, Xin, et al. "Knowledge vault: A web-scale approach to probabilistic knowledge fusion." *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2014.

Vielen Dank für Ihre Aufmerksamkeit

Thank you for your attention

πολλές ευχαριστίες

Yinchong Yang
Doktorand

Ludwig-Maximilians-Universität München

Siemens AG, Corporate Technology

Kontakt:
yinchong.yang@siemens.com