News Recommendation and Time Series Forecasting

Adaptive Pointwise-Pairwise Learning-to-Rank for Content-based Recommendation & Period-aware Time Series Forecasting using Recurrent Neural Networks

Yagmur Gizem Cinar

November 2, 2020

2015-2019 **PhD in Computer Science**, Univ. Grenoble Alpes, France
Sequence Prediction with Recurrent Neural Networks in the Context of Time Series and Information Retrieval Search Sessions

2018-2020 **Postdoctoral Researcher**, Naver Labs Europe, France
Learning-to-rank and Sequential Contextualized News Recommendation

2020- **Postdoctoral Researcher**, LIG, Univ. Grenoble Alpes, France
Document Representation and Learning-to-Rank

Table of contents

Adaptive Pointwise-Pairwise Learning-to-Rank for Content-based Personalized Recommendation

Yagmur Gizem Cinar, Jean-Michel Renders

NAVER LABS Europe

Recommendation and User Feedback

1

[1]Photo credit: http://www.toptal.com/.

Recommendation

Estimating how much a user will like an item (*e.g.*, news article)

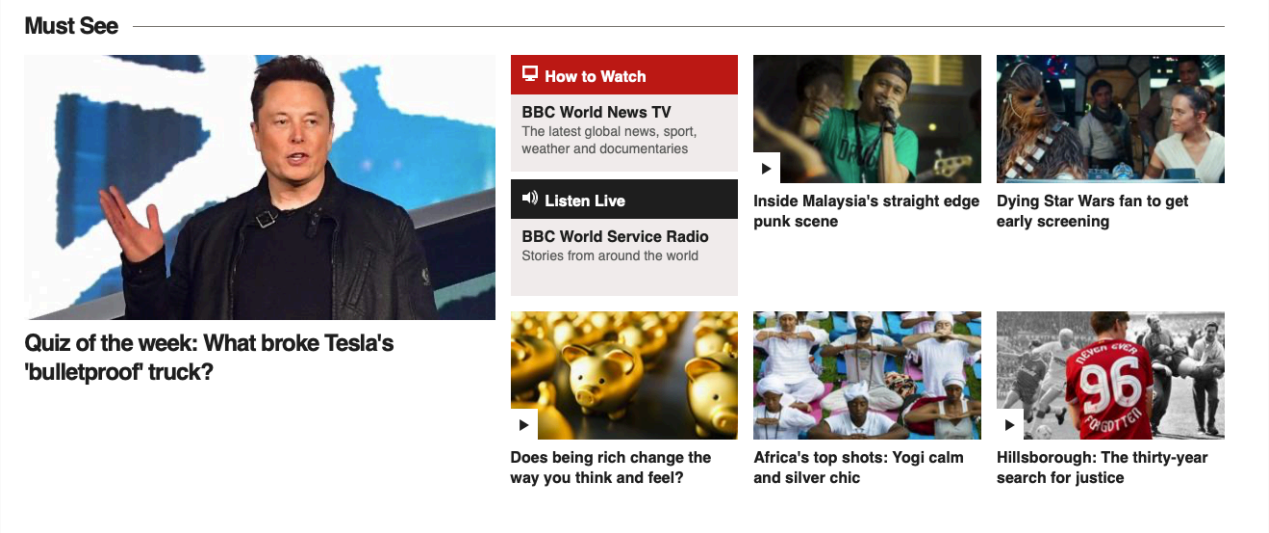


Figure 1: BBC News Must See Section [2]

[2] Taken from https://www.bbc.com/news on 29 Nov 2019.

Recommendation from User Implicit Feedback

User explicit feedback corresponds the user's preferences that is expressed by the user explicitly

- Movie ratings
- Product reviews

User implicit feedback corresponds to user interactions with the system

- **clicks**
- bookmarks
- *reading a news article*

User Clicks

Why click?

- Title/snippet is attractive
- Ranked higher
- Relevant item

Why not click?

- Seeing title/snippet is enough
- Simply not seen, or requires extra effort to be seen
- User-side reasons

User Clicks (2)

Pros:

- Mostly aligned with user interests
- No user effort required (not intrusive)
- Easy to collect, no annotation required
- Vast in amount

Cons:

- Noisy, might not exactly reflect user interest
- Biased (position bias)

Learning-to-Rank

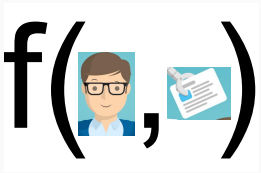
Learning-to-Rank Approaches

Learning-to-rank approaches categorized according to their (surrogate) loss:

Learning-to-Rank Approaches

Learning-to-rank approaches categorized according to their (surrogate) loss:

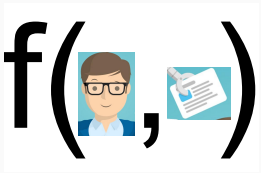
Pointwise



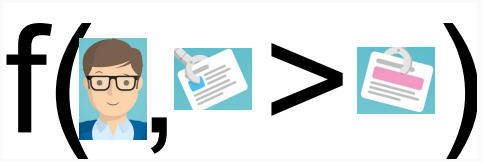
Learning-to-Rank Approaches

Learning-to-rank approaches categorized according to their (surrogate) loss:

Pointwise



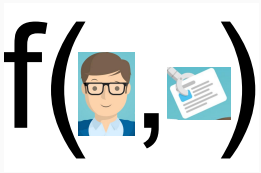
Pairwise



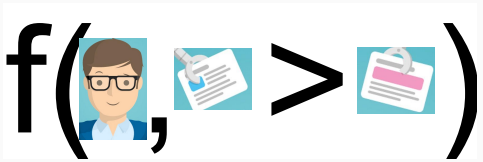
Learning-to-Rank Approaches

Learning-to-rank approaches categorized according to their (surrogate) loss:

Pointwise



Pairwise



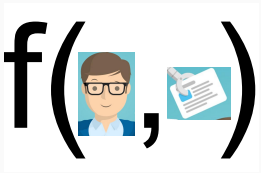
Listwise

Learning-to-Rank Approaches

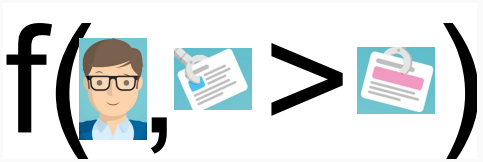
Learning-to-rank approaches categorized according to their (surrogate) loss:

Pointwise

Direct relevance of an item



Pairwise



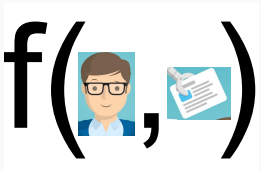
Listwise

Learning-to-Rank Approaches

Learning-to-rank approaches categorized according to their (surrogate) loss:

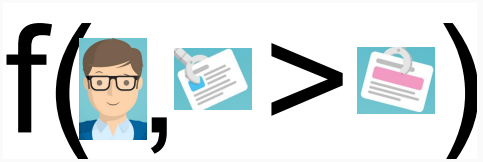
Pointwise

Direct relevance of an item



Pairwise

Pairwise preference over two items



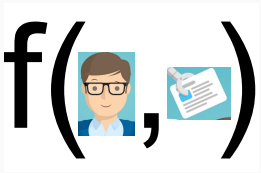
Listwise

Learning-to-Rank Approaches

Learning-to-rank approaches categorized according to their (surrogate) loss:

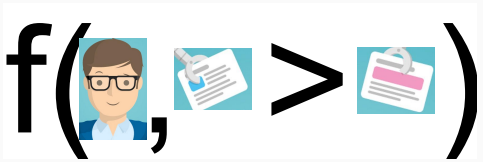
Pointwise

Direct relevance of an item



Pairwise

Pairwise preference over two items



Listwise

Ranking loss over entire list

Pointwise Learning-to-Rank (1)

Pointwise ranking directly estimates the relevance of an item i for a user u

- Classification
- Ordinal regression (for graded relevances)

Relevance probability of an item i for user u

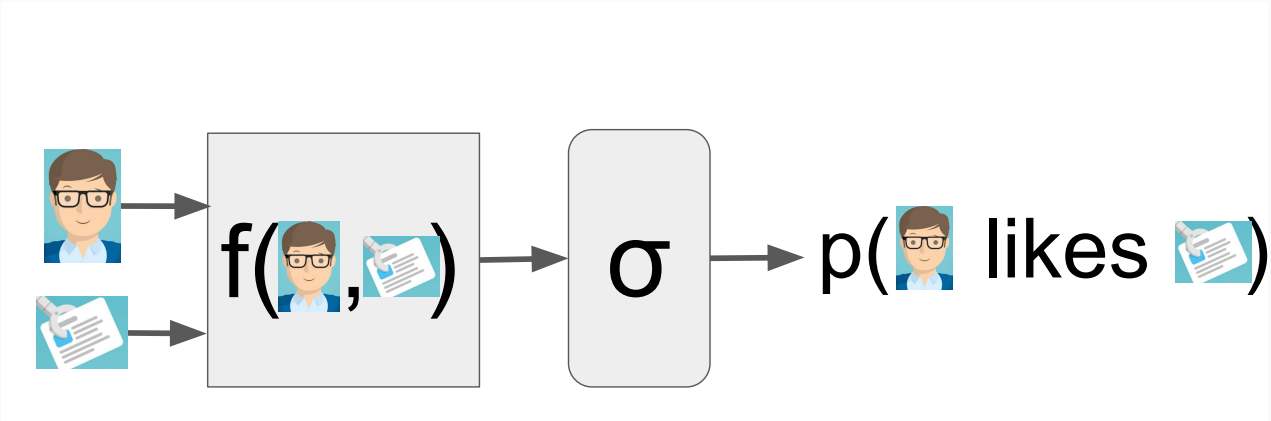
$$p(i|u) = \sigma(f(u, i|\theta))$$

$f(u, i|\theta)$ is a scoring function estimates the relevance score of the item i for the user u

Pointwise Learning-to-Rank (2)

Pointwise ranking directly estimates the relevance of an item

Relevance probability of an item i for user u



$$\mathcal{L}_{\text{pointwise}}(\theta) = -\sum_{(u,i)\in D} \left(y_{u,i} \log \sigma\big(f(u,i|\theta)\big) + (1 - y_{u,i}) \log \left(1 - \sigma\big(f(u,i|\theta)\big)\right)\right)$$

Pairwise Learning-to-Rank (1)

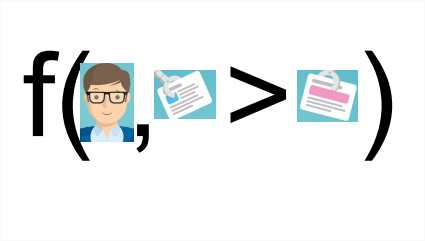
Pairwise ranking estimates the relative order between a pair of items

Optimizes the model parameters (during training) by maximizing the probability of an item i to be preferred over an item j for a user u

Relevance probability for a triplet

The probability of an item i preferred over an item j for a user u

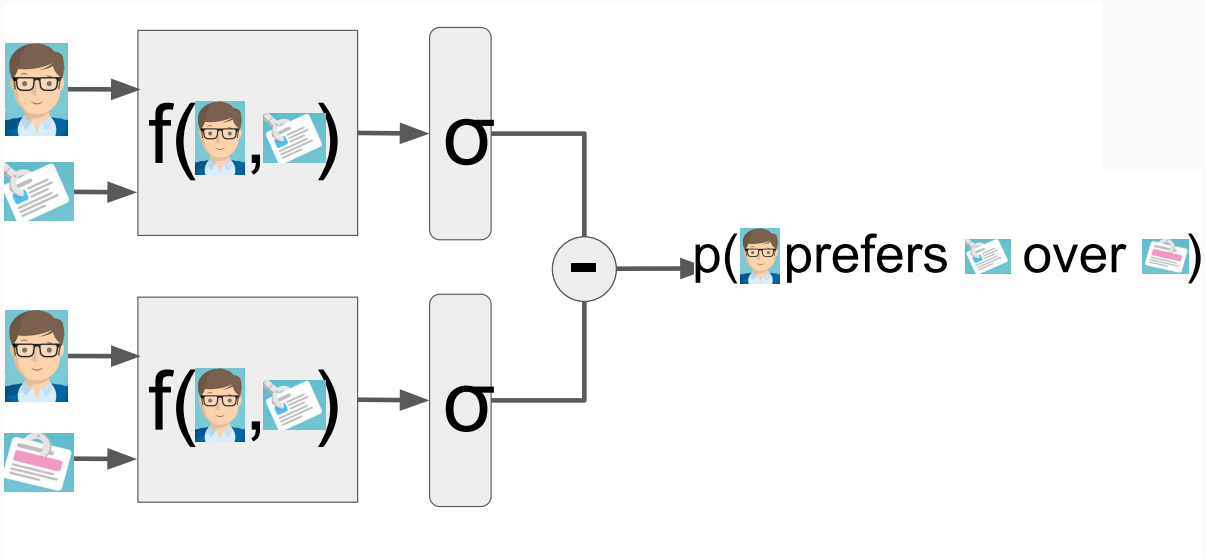
$$p(i > j|u) = \sigma(f(u, i|\theta) - f(u, j|\theta)) \tag{1}$$



Pairwise Learning-to-Rank (2)

Pairwise ranking estimates the relative order between a pair of items

Relevance probability for a triplet



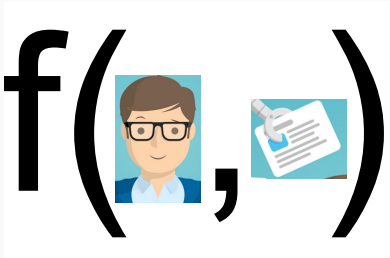
$$\mathcal{L}_{\text{pairwise}}(\theta) = -\sum_{(u,i,j)\in D'} \Big(y_{u,i>j} \log \sigma\big(f(u,i|\theta) - f(u,j|\theta)\big) + (1 - y_{u,i>j}) \log\Big(1 - \sigma\big(f(u,i|\theta) - f(u,j|\theta)\big)\Big)\Big)$$

Pointwise versus Pairwise Ranking

In general,

Pointwise Ranking

+ Less sensitive to noisy labels
- More sensitive to class imbalance

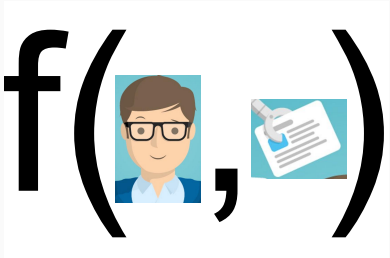


Pointwise versus Pairwise Ranking

In general,

Pointwise Ranking

+ Less sensitive to noisy labels
- More sensitive to class imbalance

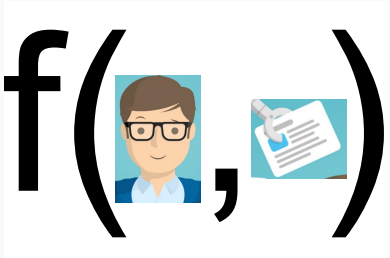


Pointwise versus Pairwise Ranking

In general,

Pointwise Ranking

+ Less sensitive to noisy labels
- More sensitive to class imbalance

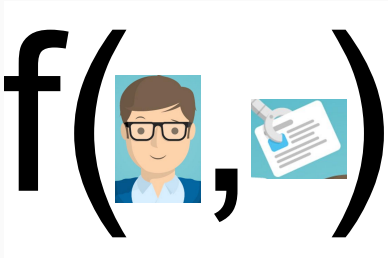


Pointwise versus Pairwise Ranking

In general,

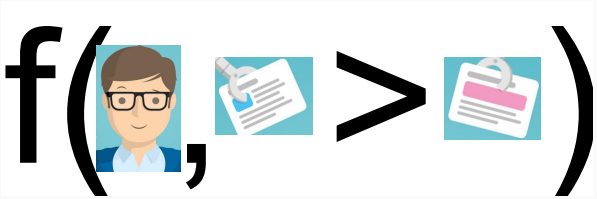
Pointwise Ranking

+ Less sensitive to noisy labels
- More sensitive to class imbalance



Pairwise Ranking

+ Better formulates the ranking problem
+ Less sensitive to class imbalance
- More sensitive to noisy labels

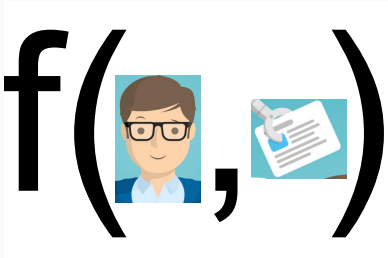


Pointwise versus Pairwise Ranking

In general,

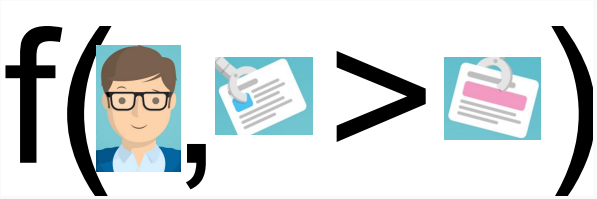
Pointwise Ranking

+ Less sensitive to noisy labels
- More sensitive to class imbalance



Pairwise Ranking

+ Better formulates the ranking problem
+ Less sensitive to class imbalance
- More sensitive to noisy labels

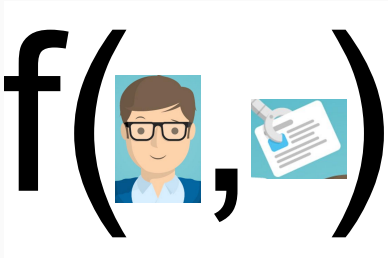


Pointwise versus Pairwise Ranking

In general,

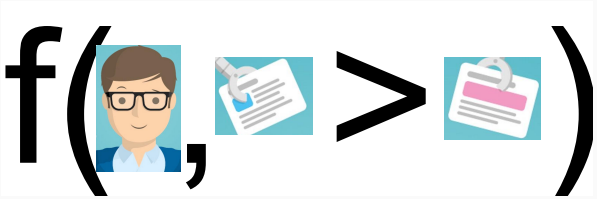
Pointwise Ranking

+ Less sensitive to noisy labels
- More sensitive to class imbalance



Pairwise Ranking

+ Better formulates the ranking problem
+ Less sensitive to class imbalance
- More sensitive to noisy labels

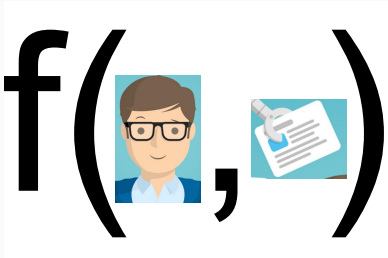


Pointwise versus Pairwise Ranking

In general,

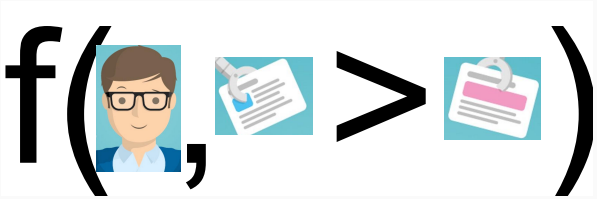
Pointwise Ranking

+ Less sensitive to noisy labels
- More sensitive to class imbalance



Pairwise Ranking

+ Better formulates the ranking problem
+ Less sensitive to class imbalance
- More sensitive to noisy labels

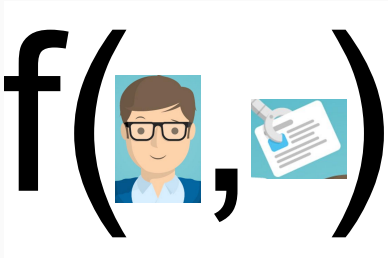


Pointwise versus Pairwise Ranking

In general,

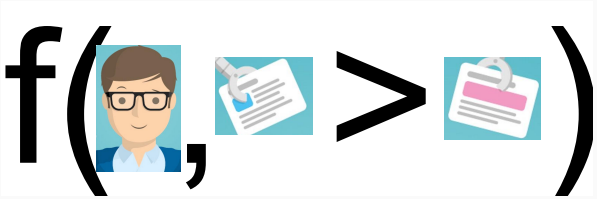
Pointwise Ranking

+ Less sensitive to noisy labels
- More sensitive to class imbalance



Pairwise Ranking

+ Better formulates the ranking problem
+ Less sensitive to class imbalance
- More sensitive to noisy labels



Can we *combine* the two *adaptively* and *optimally*?

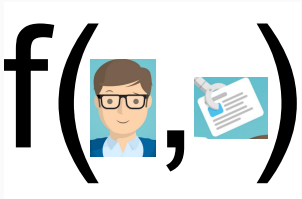
Adaptive Pointwise-Pairwise Learning-to-Rank

Adaptive Pointwise-Pairwise Learning-to-Rank (1)

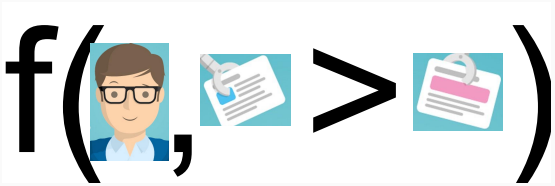
Adaptive Pointwise-Pairwise Ranking
3

- Adaptive combination of the two approaches

Pointwise



Pairwise

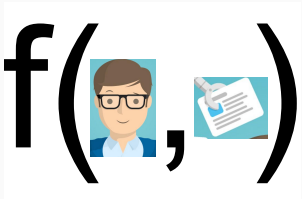


Adaptive Pointwise-Pairwise Learning-to-Rank (1)

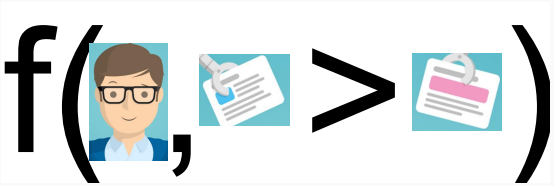
Adaptive Pointwise-Pairwise Ranking
3

- Adaptive combination of the two approaches

Pointwise

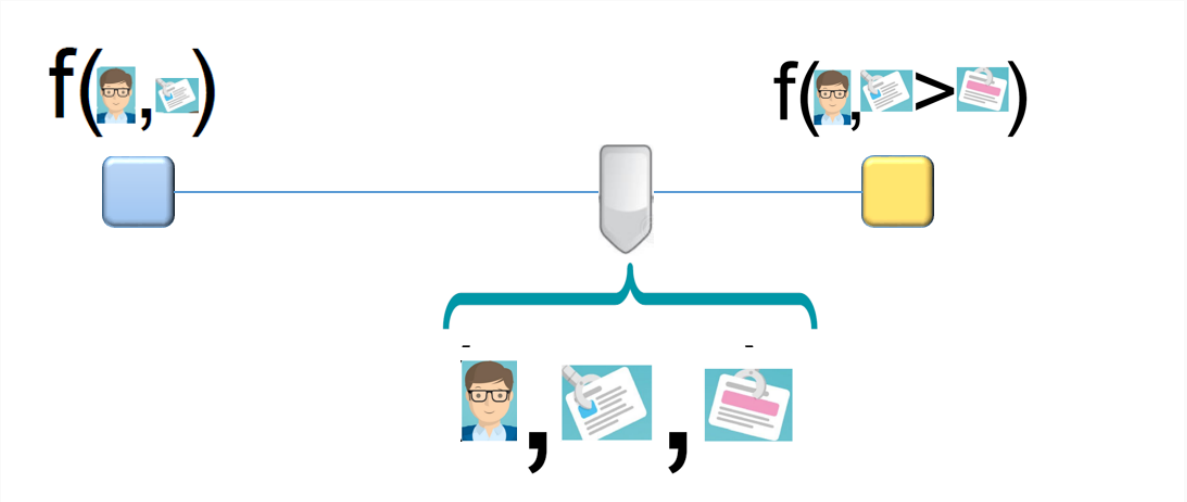


Pairwise



- The precise balance between pointwise and pairwise

Adaptive Pointwise-Pairwise

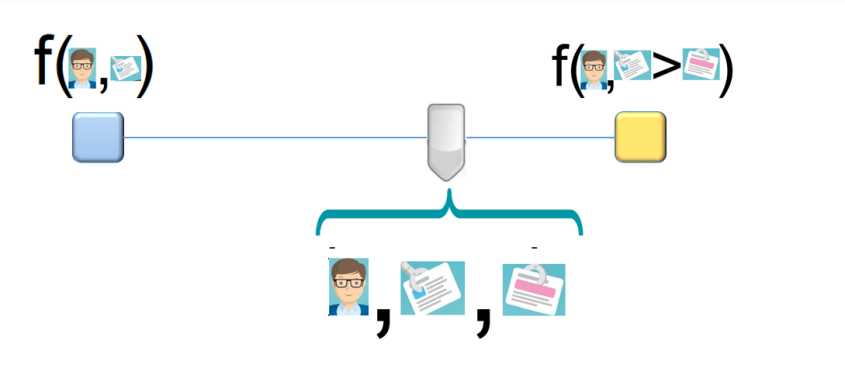


Adaptive Pointwise-Pairwise Learning-to-Rank (2)

Adaptive Pointwise-Pairwise Relevance probability for a triplet

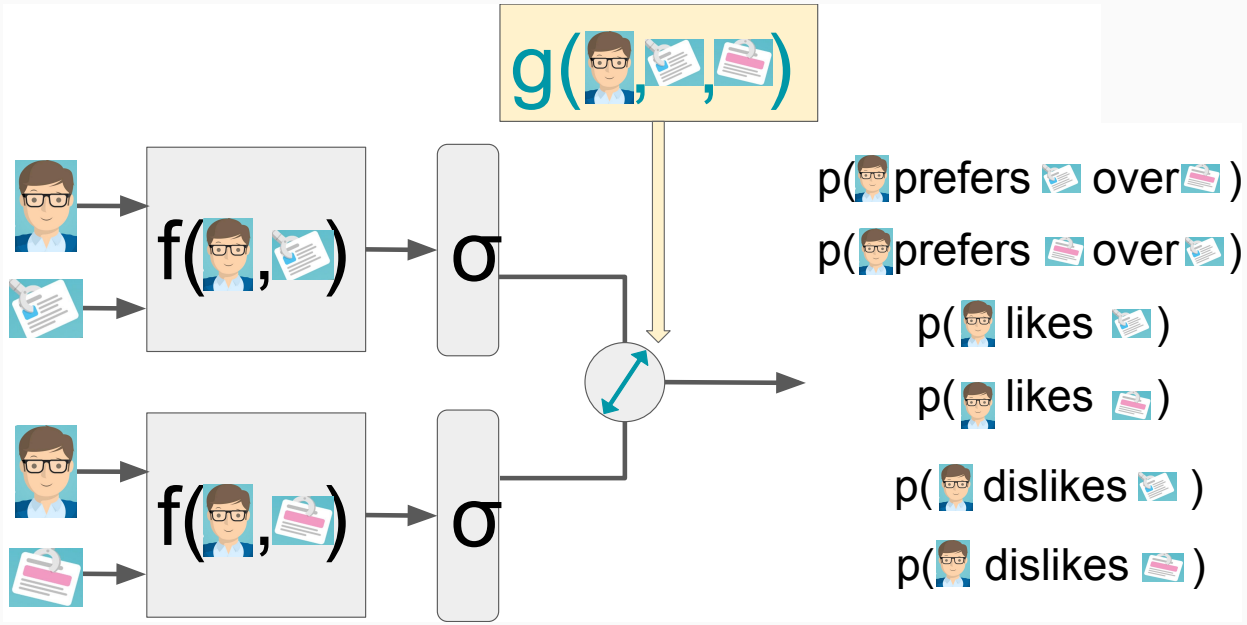
$$p(i > j|u) = \sigma(f(u, i|\theta) - \gamma f(u, j|\theta)) \tag{2}$$

- γ can take values between $[0, 1]$
- Computed as a function of user u, items i and j, $\gamma = g(u, i, j|\theta_g)$



Adaptive Pointwise-Pairwise Learning-to-Rank (3)

Adaptive Pointwise-Pairwise Relevance probability for a triplet



$$\mathcal{L}_{\text{adaptive}}(\theta, \theta_g) = -\sum_{(u,i,j)\in D'} \Big(y_{u,i>j} \log \sigma\big(f(u,i|\theta) - g(u,i,j|\theta_g) f(u,j|\theta)\big)$$

$$+(1 - y_{u,i>j}) \log\Big(1 - \sigma\big(f(u,i|\theta) - g(u,i,j|\theta_g) f(u,j|\theta)\big)\Big)\Big)$$

Personalized Recommendation Model Details (1)

Content-based user representation

A user is represented by this user's previous interactions (clicked and not clicked items)

$$\mathbf{x}_u = \boldsymbol{\mu}_u^+ - \boldsymbol{\beta} \odot \boldsymbol{\mu}_u^-$$

- $\boldsymbol{\mu}_u^+ \in \mathbb{R}^{d \times 1}$ is the mean of the user u's clicked items' embedding
- $\boldsymbol{\mu}_u^- \in \mathbb{R}^{d \times 1}$ is the mean of the user u's non-clicked items' embedding
- $\boldsymbol{\beta} \in \mathbb{R}^{d \times 1}$ scales the user negative centroid

Personalized Recommendation Model Details (2)

Relevance Scoring

Relevance scoring of an item i for a user u is calculated from a simple bilinear form:

$$f(u, i|W) = \mathbf{x}_i^\top \mathbf{W} \mathbf{x}_u$$

- $\mathbf{W} \in \mathbb{R}^{d \times d}$ is a diagonal matrix

Adaptive Mixing

Pair scoring of an item i for a user u is calculated from a normalized simple bilinear form:

$$g(i, j) = \frac{\exp(\mathbf{x}_i^\top \mathbf{W_g} \mathbf{x}_j)}{\sum_{(i', j') \in \mathcal{B}} \exp(\mathbf{x}_{i'}^\top \mathbf{W_g} \mathbf{x}_{j'})} \tag{3}$$

- $\mathbf{W_g} \in \mathbb{R}^{d \times d}$ is a diagonal matrix
- $g(i, j)$ take values between $[0, 1]$

Experiments

Datasets

INR news

- 7 days of click activity logs

	Training	Validation	Test
days	1-3	4	5-8

- 2 news categories (vocabulary size):
  - Entertainment (40K)
  - Sports (30K)
- 28 articles are recommended
- 1000 users selected at random for each run with a predefined seed initialization

Datasets (2)

Outbrain click prediction

- 14 days of click activity logs

	Training	Validation	Test
days	1-7	8-10	11-14

- An article is represented as a concatenation of
  - category
  - topic
  - entity

  features (total dimension of 24K)
- On average 5-6 articles are recommended, and 12 articles maximum
- 5000 users selected at random for each run with a predefined seed initialization

Experimental Setup (1)

- We compare pointwise-pairwise ranking approach with
  - pointwise
  - pairwise [Rendle et al., 2009]
  - listwise (listNET [Cao et al., 2007], λRank [Burges, 2010], ListAP[Revaud et al., 2019])
  - other combined pointwise-pairwise (alternating [Lei et al., 2017], joint [Wang et al., 2016])
- ADAM optimization used to update model weights with adaptive stochastic gradient descent
- Batch norm and L1 regularization is applied
- Hyperparameters search on validation set

Learning rate	{0.01, 0.0001, 0.0001}
Regularization constant	{0.0001, 0.0001, 0.00001, 0.00001}

Experimental Setup (2)

Evaluation

- Average Precision (*AP*)
- Normalized discounted cumulative gain at K (*NDCG@K*):

$$DCG\ @K = \sum_{k=1}^{K} \frac{2^{rel(i_k)-1}}{\log_2(k+1)}$$

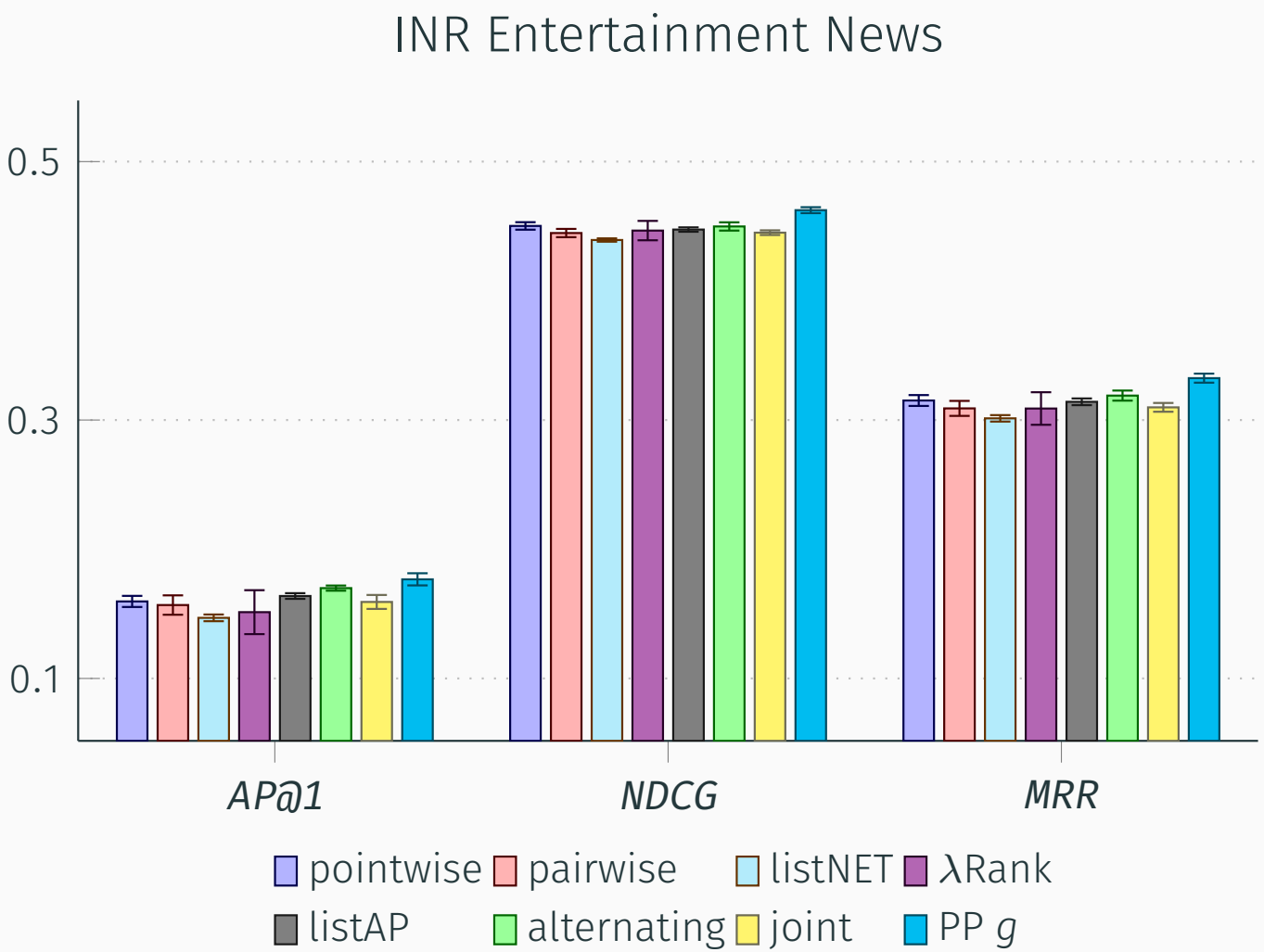
$$NDCG@K = \frac{DCG@K}{iDCG@K}$$

- Mean reciprocal rank (*MRR*)

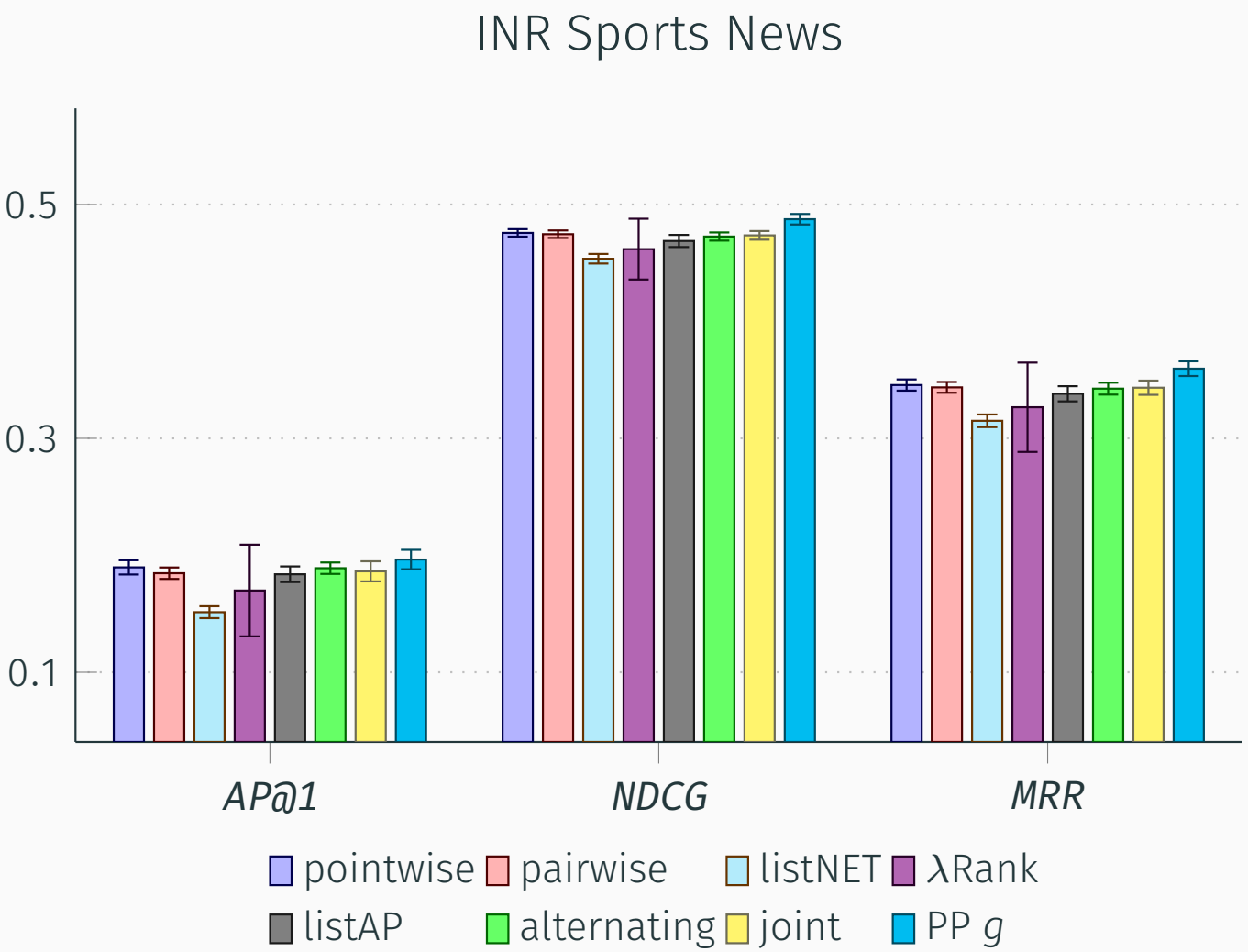
$$MRR = \frac{1}{N} \sum_{n=1}^{N} \frac{1}{\text{rank}_n}$$

- Wilcoxon signed-rank test with Bonferroni correction is used

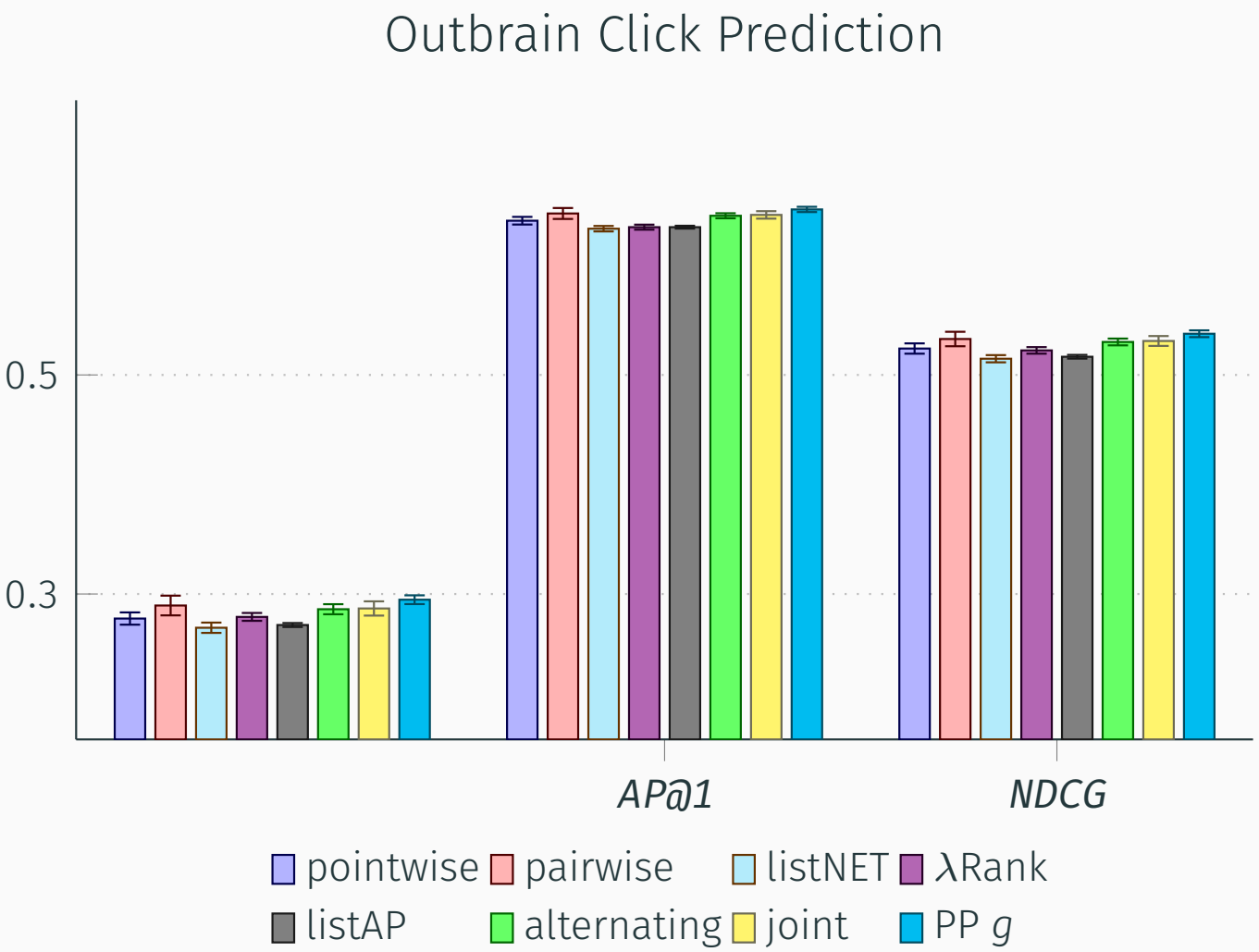
Results on News Recommendation (1)



Results on News Recommendation (2)



Results on News Recommendation (3)



Conclusion

Conclusion

Adaptive Pointwise-Pairwise surrogate loss for Personalized Content-based Recommendation

- The precise balance between pointwise and pairwise contributions could depend on the particular pair or triplet instance
- Adaptive Pointwise-Pairwise ranking significantly outperforms pointwise, pairwise, listwise and other (combined) pointwise-pairwise ranking approaches on several personalized news recommendation datasets
- Future work on combining with listwise learning-to-rank loss
- *https://github.com/ygcinar/pointwise-pairwise-recommendation*

Period-aware Time Series Forecasting using RNNs

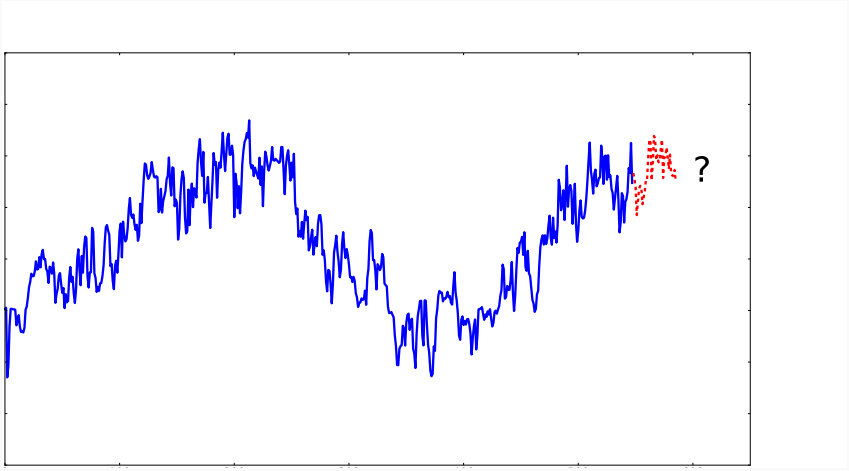
Yagmur Gizem Cinar, Hamid Mirisaee, Parantapa Goswami, Ali Aït-Bachir , Eric Gaussier

Sequence Prediction

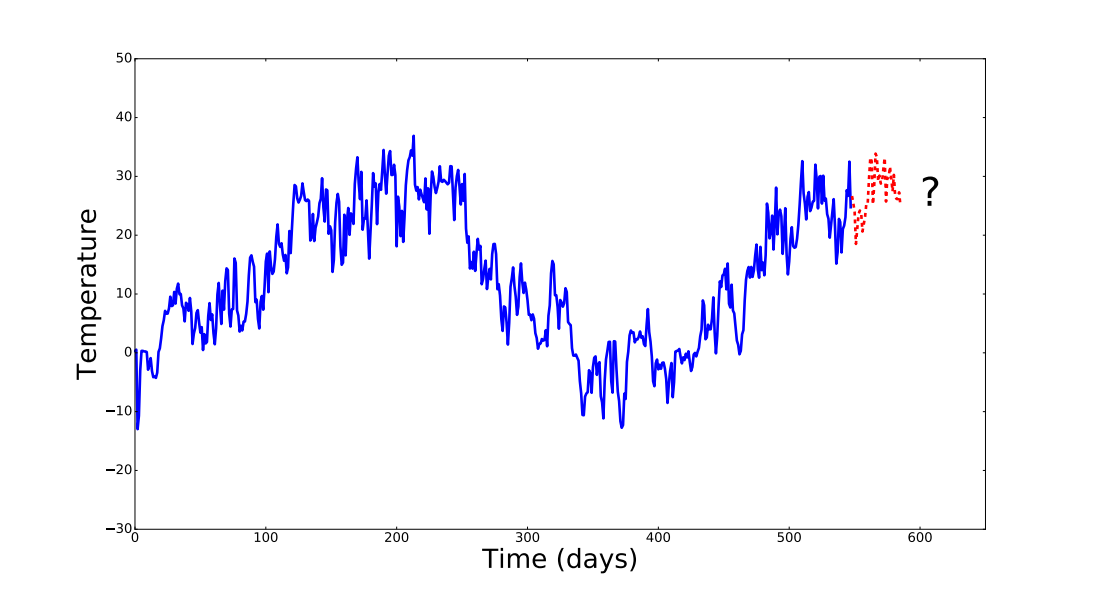
- Sequence prediction applies to various domains:
  - Predictive assistance in user activities

    e.g., auto-completing an email, next query prediction, or generating ambient music
  - Predicting usage of a resource or consumption of a product

    e.g., electricity consumption, email server load



Time Series



☐ Time Series

- Multivariate, multi-scale
- May contain pseudo-periods

  pseudo-periods: time intervals across which there is a strong correlation, positive or negative, between the values of the time series

- May (or may not) contain missing values: random and gaps

Time Series Forecasting

- State-of-the-art methods
  - Stochastic methods: ARIMA, VARIMA
    [Chatfield, 2003, Tiao and Box, 1981]
  - Kernel methods: Support Vector Machines
    [Müller et al., 1997, Sapankevych and Sankar, 2009]
  - Ensemble methods: Random Forests [Kane et al., 2014]
  - Neural Network methods: Recurrent Neural Networks
    [Werbos, 1988, Gers et al., 2001]

  Recent sequence modeling breakthrough:
  sequence-to-sequence RNNs
  [Sutskever et al., 2014, Bahdanau et al., 2014, Xingjian et al., 2015]

Time Series Forecasting

- State-of-the-art methods
  - Stochastic methods: ARIMA, VARIMA
    [Chatfield, 2003, Tiao and Box, 1981]
  - Kernel methods: Support Vector Machines
    [Müller et al., 1997, Sapankevych and Sankar, 2009]
  - Ensemble methods: Random Forests [Kane et al., 2014]
  - Neural Network methods: Recurrent Neural Networks
    [Werbos, 1988, Gers et al., 2001]

  Recent sequence modeling breakthrough:
  sequence-to-sequence RNNs
  [Sutskever et al., 2014, Bahdanau et al., 2014, Xingjian et al., 2015]
- Research questions
  - Can (sequence-to-sequence) Recurrent Neural Networks (RNNs) model pseudo-periods in time series?
  - *Are they robust to missing values?*

Background: RNNs with attention mechanism

Deep Recurrent Neural Networks (RNNs) with attention mechanism

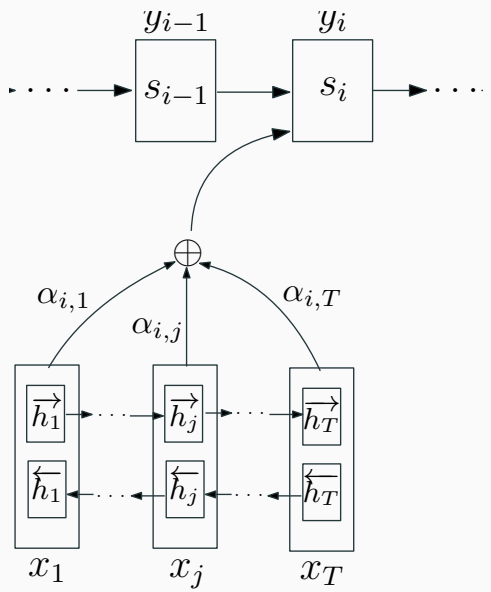
Sequence-to-sequence prediction with attention mechanism [Bahdanau et al., 2014]

Encoder: Forward and backward RNNs for an input sequence $\mathbf{x}$ (Bidirectional RNNs)

$$(x_1, ..., x_T) \Rightarrow (\overrightarrow{h}_1, ..., \overrightarrow{h}_T)$$

$$(x_T, ..., x_1) \Rightarrow (\overleftarrow{h}_1, ..., \overleftarrow{h}_T)$$

$$\mathbf{h}_j = \begin{pmatrix} \overrightarrow{h}_j^\top \\ \overleftarrow{h}_j^\top \end{pmatrix}$$



Attention:

$$e_{ij} = \mathbf{v}_a^T \tanh(\mathbf{W}_a \mathbf{s}_{i-1} + \mathbf{U}_a \mathbf{h}_j)$$

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T} \exp(e_{ik})}, \; \mathbf{c}_i = \sum_{j=1}^{T} \alpha_{ij} \mathbf{h}_j$$

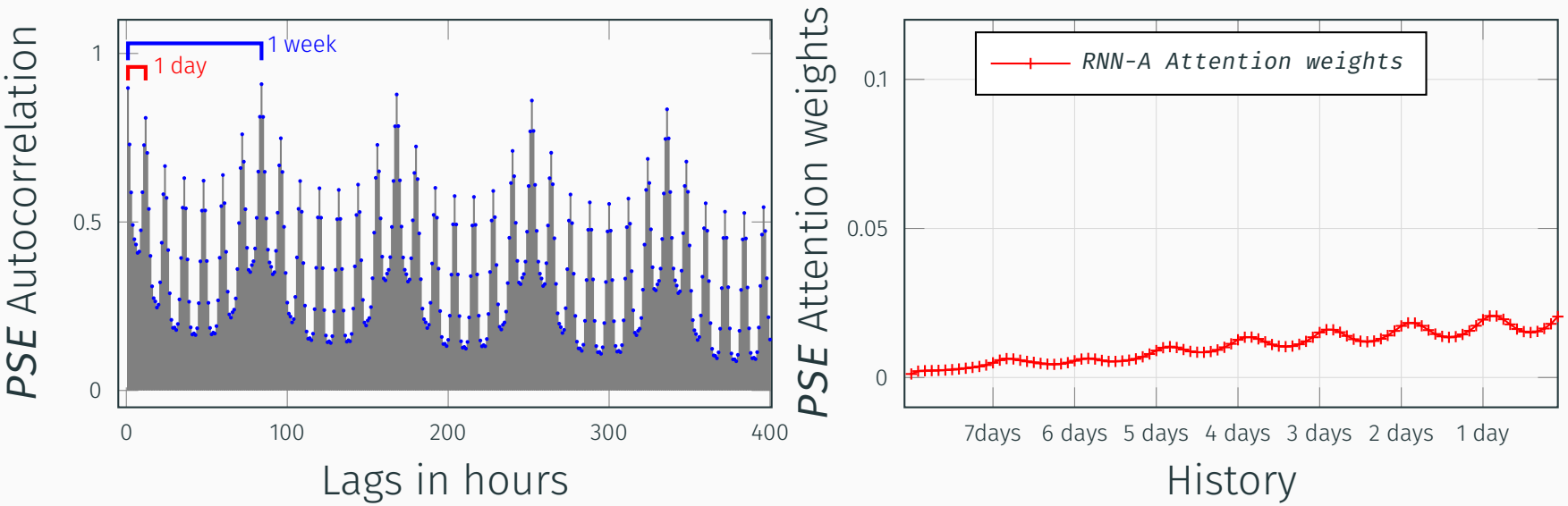
Decoder:

$$\mathbf{s}_i = g(y_{i-1}, \mathbf{s}_{i-1}, \mathbf{c}_i)$$

Visualization of Periods and Attention weights

Time Series Forecasting - Visualization of Periods and Attention weights

- Attention weights as an (indirect) indication of the capacity to capture periods



PSE: Polish electricity load time series

Modeling pseudo-periods

Period-aware content attention mechanism (1)

Modeling periods in time series[4]

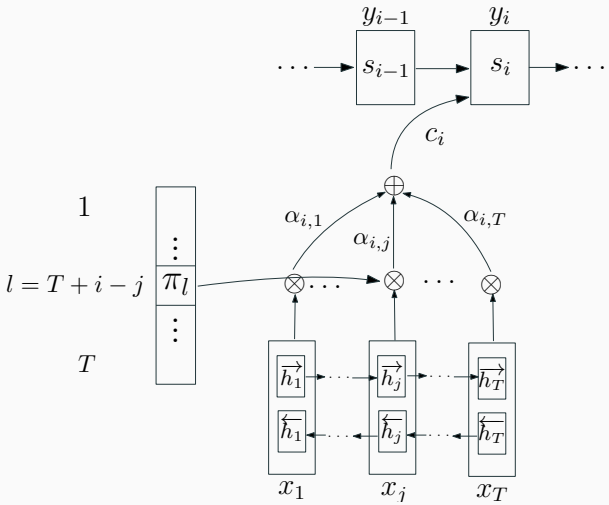
Explicitly model all relative positions and learn a weight to re-weigh the importance of given input according to relative position with respect to output

RNN-π: π is a **vector** of dimension T ($\in \mathbb{R}^T$)

$$e_{ij} = \mathbf{v}_a^\top \tanh(\mathbf{W}_a \mathbf{s}_{i-1} + \mathbf{U}_a(\boldsymbol{\pi}_{i-j} \mathbf{h}_j)) \mathbf{1}_{(i-j \leq T)}$$

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T} \exp(e_{ik})}$$

$$\mathbf{c}_i = \sum_{j=1}^{T} \alpha_{ij} \mathbf{h}_j$$



[4] Cinar et al., *Period-aware Content Attention RNNs for Time Series Forecasting with Missing Values*, Neurocomputing 2018.
Cinar et al., *Position-based Content Attention for Time Series Forecasting with Sequence-to-Sequence RNNs*, ICONIP 2017.

Period-aware content attention mechanism (2)

Modeling periods in time series [5]

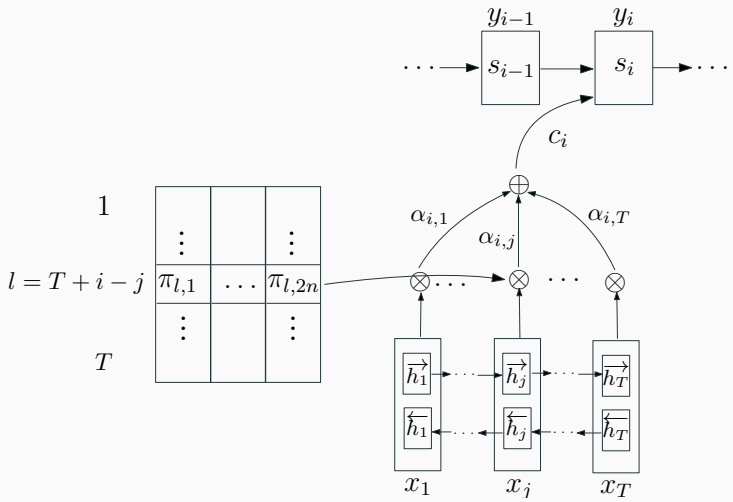
Explicitly model all relative positions and learn a weight to re-weigh the importance of given input according to relative position with respect to output

RNN-Π: Π is a **matrix** in $\mathbb{R}^{2n \times T}$

$$e_{ij} = \mathbf{v}_a^\top \tanh(\mathbf{W}_a \mathbf{s}_{i-1} + \mathbf{U}_a(\Pi_{.(i-j)} \odot \mathbf{h}_j)) \mathbf{1}_{(i-j \leq T)}$$

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T} \exp(e_{ik})}$$

$$\mathbf{c}_i = \sum_{j=1}^{T} \alpha_{ij} \mathbf{h}_j$$

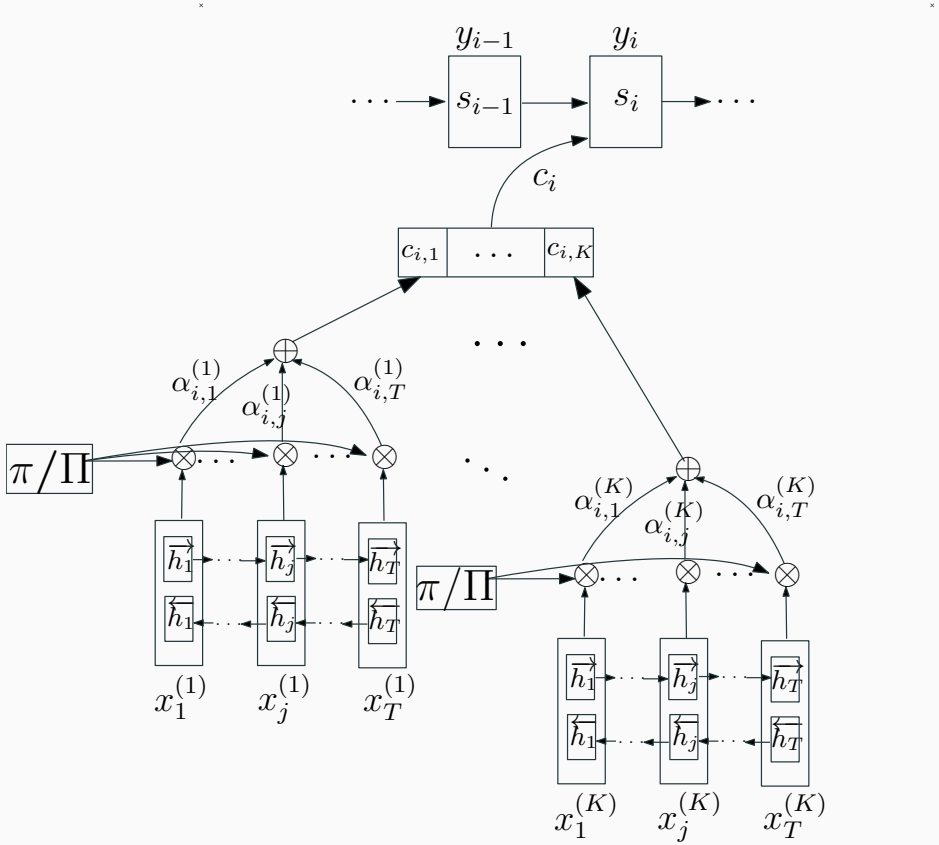


[5] Cinar et al., *Period-aware Content Attention RNNs for Time Series Forecasting with Missing Values*, Neurocomputing 2018.
Cinar et al., *Position-based Content Attention for Time Series Forecasting with Sequence-to-Sequence RNNs*, ICONIP 2017.

Multivariate Extensions (1)

For a K multivariate time series, each time series might have a particular pseudo-periods, $1 \leq k \leq K$

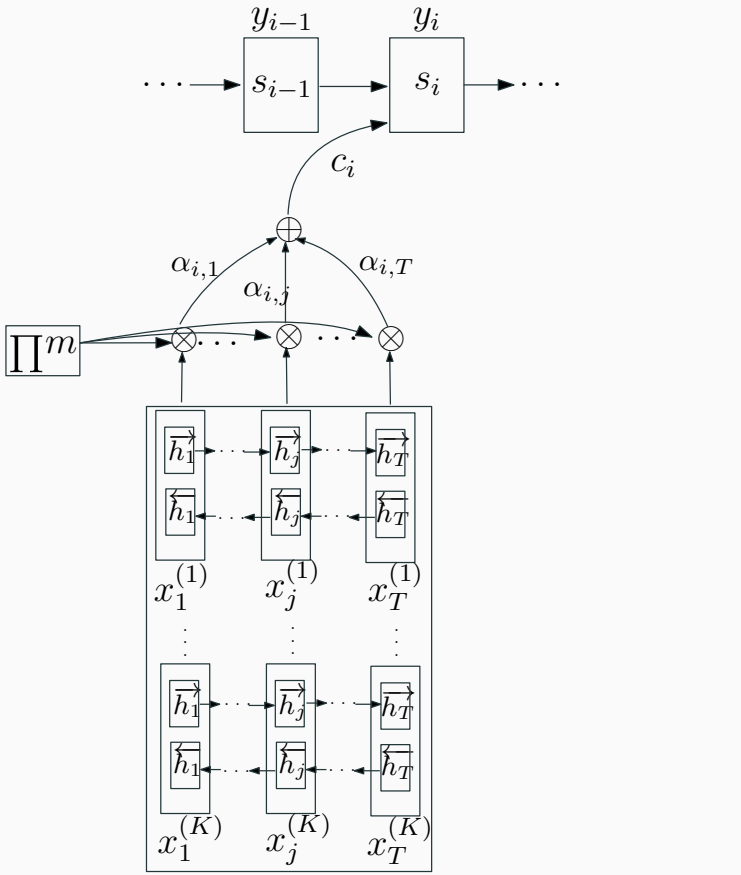
For RNN-π/Π:



Multivariate Extensions (2)

For a K multivariate time series, each time series might have a particular pseudo-periods, $1 \leq k \leq K$

For RNN-Π^m: Π^m is a matrix in $\mathbb{R}^{2Kn \times T}$



Datasets

Experiments on 6 publicly available time series data sets from different sources

History: the length of the historical input

Horizon: the length of the future, output to predict

Name	Usage	#Instances	History	Horizon	Sampling rate
Polish Electricity (*PSE*)	Univariate	46379	96	4	2 hours
Polish Weather (*PW*)	Univariate	4595	548	7	1 days
Numenta Benchmark (*NAB*)	Univariate	18050	72	6	5 minutes
Air Quality (*AQ*)	Univ./Multiv.	9471	192	6	1 hour
Appliances Energy Pred. (*AEP*)	Univ./Multiv.	19735	216	6	10 minutes
Ozone Level Detection (*OLD*)	Univ./Multiv.	2536	548	7	1 day

Evaluation

- Mean Squared Error (**MSE**)
- Symmetric Mean Absolute Percentage Error (**SMAPE**)

Univariate Forecasting results

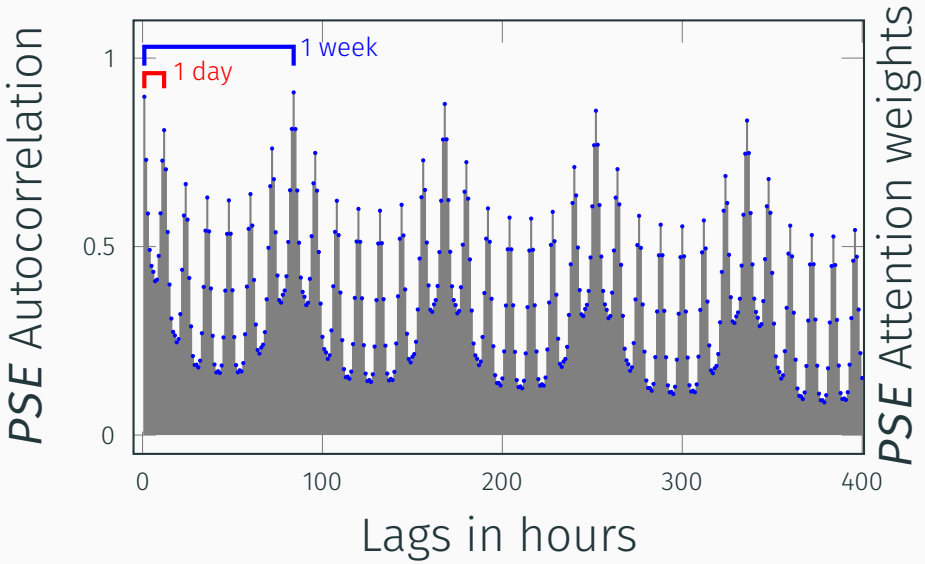
Overall results for univariate case with Mean Squared Error **MSE** (left value) and Symmetric Mean Absolute Percentage Error **SMAPE** (right value)

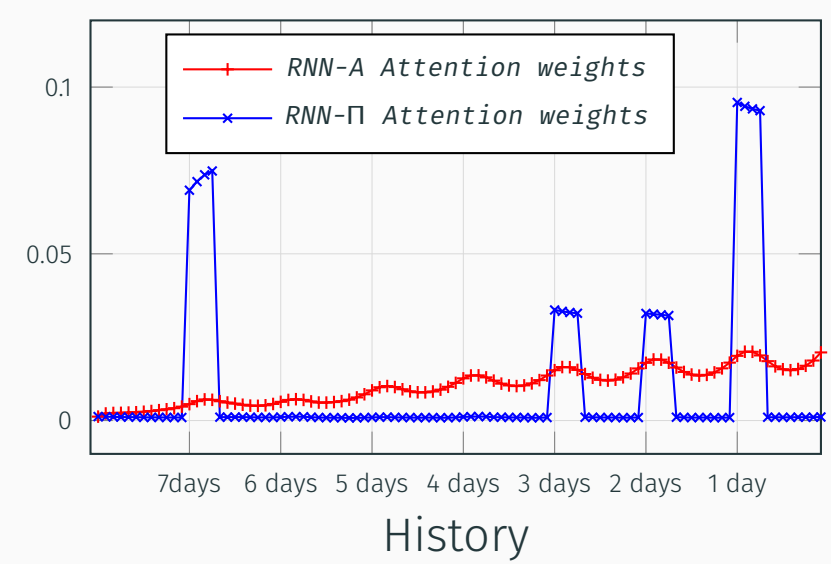
Data	RNN-A	RNN-π	RNN-Π	ARIMA	RF	*Selected*
AQ	0.282*(0.694)	0.257(0.661)	**0.250**(0.669)	0.546*(0.962)	0.299*(0.762)	Π
OLD	0.319*(0.595)	**0.271**(0.523)	0.275(0.586)	0.331*(0.619)	0.305*(0.606)	Π
AEP	0.025*(0.085)	0.029*(0.101)	0.027*(0.095)	**0.021**(0.066)	**0.021**(0.085)	Π
NAB	0.642*(0.442)	**0.475**(0.323)	0.540*(0.369)	1.677*(1.310)	0.779*(0.608)	Π
PW	0.166*(0.558)	**0.152**(0.547)	0.162*(0.565)	0.213*(0.610)	0.156(0.544)	π
PSE	0.034*(0.282)	**0.032**(0.264)	0.033*(0.256)	0.623*(1.006)	0.053*(0.318)	π

Attention weights

Period-aware attention weights

Attention weights as an (indirect) indication of the capacity to capture periods





Multivariate Forecasting results

Overall results for multivariate case with mean squared error (MSE)

Dataset	RNN-A	RNN-π	RNN-Π	RNN-Π^m	RF	*Selected model*
AQ	0.352*	0.276*	**0.268**	0.300*	0.450*	Π^m
OLD	0.336*	0.328*	0.327*	**0.274**	0.315*	Π^m
AEP	0.029*	**0.024**	0.036*	0.026*	0.027*	Π^m

Conclusion

- Two univariate period-aware extensions:
  1. RNN-π: models periods by using a vector of relative positions
  2. RNN-$\mathbf{\Pi}$: models periods by using a matrix of relative positions (finer granularity)
- Three multivariate extensions:
  1. Individual attention mechanism per variable: RNN-π, RNN-$\mathbf{\Pi}$
  2. One global attention mechanism over all variables: RNN-$\mathbf{\Pi}^m$
- *Two approaches for handling missing values:*
  - *Exponential weight decay*
  - *Relative position in the gap*
- Proposed models outperform baselines of standard RNNs, RFs and ARIMA

Thank You!

References i


Bahdanau, D., Cho, K., and Bengio, Y. (2014).
Neural machine translation by jointly learning to align and translate.
arXiv e-prints, abs/1409.0473.


Burges, C. J. (2010).
From ranknet to lambdarank to lambdamart: An overview.
Technical Report MSR-TR-2010-82.


Cao, Z., Qin, T., Liu, T.-Y., Tsai, M.-F., and Li, H. (2007).
Learning to rank: From pairwise approach to listwise approach.
In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pages 129–136, New York, NY, USA. ACM.

References ii


Chatfield, C. (2003).
The analysis of time series: an introduction.
CRC press.


Gers, F. A., Eck, D., and Schmidhuber, J. (2001).
Applying LSTM to time series predictable through time-window approaches.
In *International Conference on Artificial Neural Networks*, pages 669–676. Springer.


Kane, M. J., Price, N., Scotch, M., and Rabinowitz, P. (2014).
Comparison of arima and random forest time series models for prediction of avian influenza h5n1 outbreaks.
BMC bioinformatics, 15(1):276.

References iii

Lei, Y., Li, W., Lu, Z., and Zhao, M. (2017).
Alternating pointwise-pairwise learning for personalized item ranking.
In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, CIKM '17, pages 2155–2158, New York, NY, USA. ACM.

Müller, K.-R., Smola, A. J., Rätsch, G., Schölkopf, B., Kohlmorgen, J., and Vapnik, V. (1997).
Predicting time series with support vector machines.
In *International Conference on Artificial Neural Networks*, pages 999–1004. Springer.

References iv


Rendle, S., Freudenthaler, C., Gantner, Z., and Schmidt-Thieme, L. (2009).
Bpr: Bayesian personalized ranking from implicit feedback.
In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, UAI 2009, pages 452–461, Arlington, Virginia, United States. AUAI Press.


Revaud, J., Almazán, J., de Rezende, R. S., and de Souza, C. R. (2019).
Learning with average precision: Training image retrieval with a listwise loss.
CoRR, abs/1906.07589.


Sapankevych, N. I. and Sankar, R. (2009).
Time series prediction using support vector machines: A survey.
IEEE Computational Intelligence Magazine, 4(2):24–38.

References v


Sutskever, I., Vinyals, O., and Le, Q. V. (2014).
Sequence to sequence learning with neural networks.
In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 27*, pages 3104–3112. Curran Associates, Inc.


Tiao, G. C. and Box, G. E. (1981).
Modeling multiple time series with applications.
Journal of the American Statistical Association, 76(376):802–816.


Wang, Y., Wang, S., Tang, J., Liu, H., and Li, B. (2016).
PPP: joint pointwise and pairwise image label prediction.
In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 6005–6013.

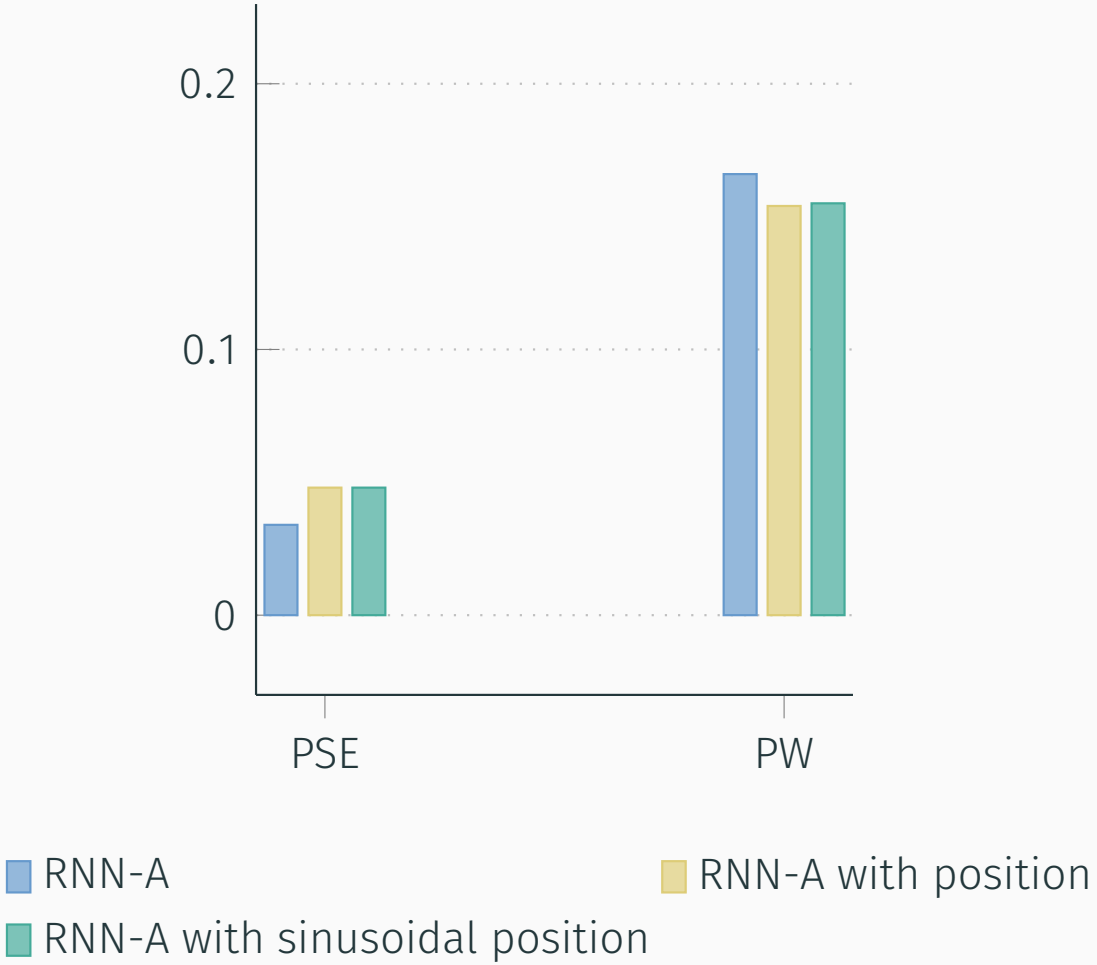
References vi

Werbos, P. J. (1988).
Generalization of backpropagation with application to a recurrent gas market model.
Neural Networks, 1(4):339 – 356.

Xingjian, S., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-K., and Woo, W.-c. (2015).
Convolutional LSTM network: A machine learning approach for precipitation nowcasting.
In *Advances in Neural Information Processing Systems*, pages 802–810.

Standard attention with position information

Mean squared error values for Sequence-to-sequence RNNs with attention (RNN-A) with and without position information on *PSE*.



Handling Missing Values

- Imputation of missing values

  padding, interpolation

- How much can we rely on imputed values?

Two reweighing schemes:

1. A decaying function: $\boldsymbol{\mu} \in \mathbb{R}^T$
   - *a priori* adapted to padding
2. At the beginning, middle or end of a gap: $\boldsymbol{M} \in \mathbb{R}^{3\times T}$
   - *a priori* adapted to interpolation methods

$$\omega(j) = \begin{cases} \exp(-\mu_j(j - j_{\text{last}})) \\ 1 + \boldsymbol{M}_{\cdot j}^{\mathsf{T}} \mathbf{Pos}(j; \theta_1^g, \theta_2^g) \end{cases}$$

Forecasting results of handling missing values

Different levels of missing values:

5 %, 10 %, 15 %, 20 %, 30 %, 40 %

Table 1: Univariate prediction on datasets with missing values and gaps, MSE(SMAPE)

Dataset	Selected-π / Π	Selected-π / Π-μ / M	RNN-A	RNN-π / Π	RNN-π / Π-μ / M	*Selected model*
PSE5	π	Π-μ	0.064*(0.345)	0.059*(0.316)	**0.055**(0.31)	Π-μ
PSE10	π	π-μ	0.066*(0.353)	0.055*(0.318)	**0.052**(0.314)	π-μ
PSE15	Π	π-μ	0.09*(0.36)	0.083(0.357)	**0.081**(0.332)	π-μ
PSE20	π	π-μ	0.079*(0.374)	0.078*(0.369)	**0.074**(0.344)	π-μ
PSE30	π	π-μ	0.104*(0.419)	0.102*(0.386)	**0.098**(0.384)	π-μ
PSE40	π	Π-M	0.113*(0.428)	0.119*(0.411)	**0.106**(0.393)	Π-M
PW5	π	π-M	0.161(0.556)	**0.157**(0.555)	0.164*(0.566)	π
PW10	π	π-M	0.16*(0.557)	**0.153**(0.54)	0.155(0.542)	π
PW15	π	π-μ	0.167(0.569)	0.167(0.556)	**0.163**(0.55)	π-μ
PW20	π	Π-M	0.209*(0.6)	0.182(0.551)	**0.177**(0.553)	Π-M
PW30	π	π-M	0.177*(0.571)	0.163(0.556)	**0.161**(0.549)	π-M
PW40	π	π-μ	0.181*(0.568)	0.172(0.564)	**0.164**(0.534)	π