# SoDeep: A Sorting Deep Net to Learn Ranking Loss Surrogates

June, 2019

Martin Engilberge, Louis Chevallier,
Patrick Pérez, Matthieu Cord

Machine Learning &
Deep Learning for
Information Access

# SoDeep: A Sorting Deep Net to Learn Ranking Loss Surrogates

June, 2019

Martin Engilberge, Louis Chevallier,
Patrick Pérez, Matthieu Cord

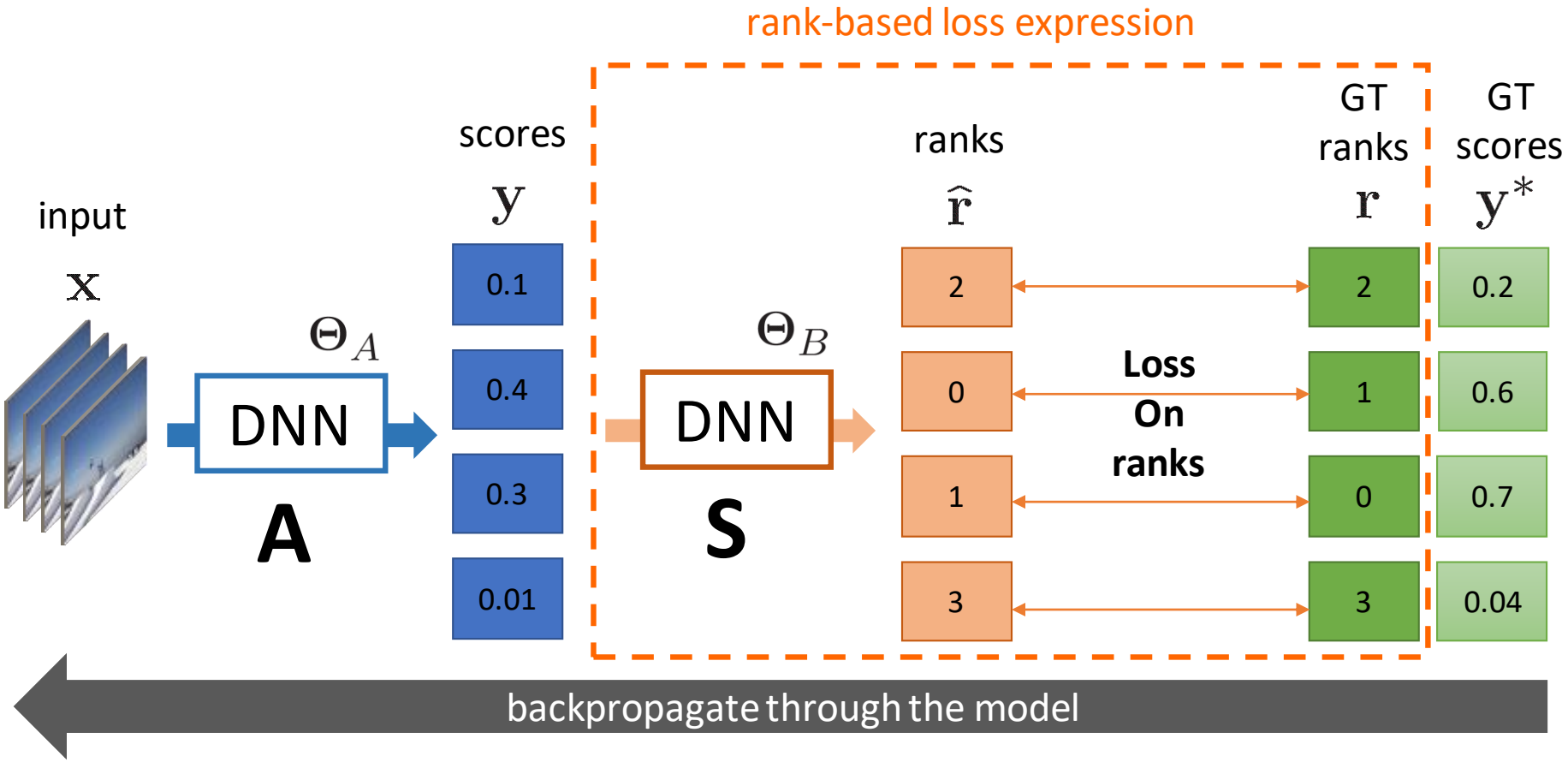
MLIA Machine Learning & Deep Learning for Information Access

# Problem

- Metrics often define machine learning tasks
- Goal: Use metric directly as loss function
- Focus on ranking metrics:
  - **mean Average Precision** (mAP)
  - **Spearman correlation**
  - **Recall@threshold**
- Computation of rank is non-differentiable

# Approach

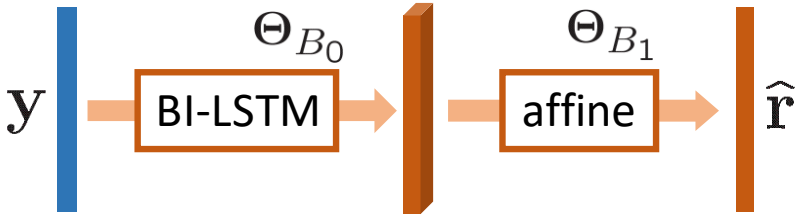
- Pretrained network computes rank from scores
- Ranking metrics expressed as a function of the rank



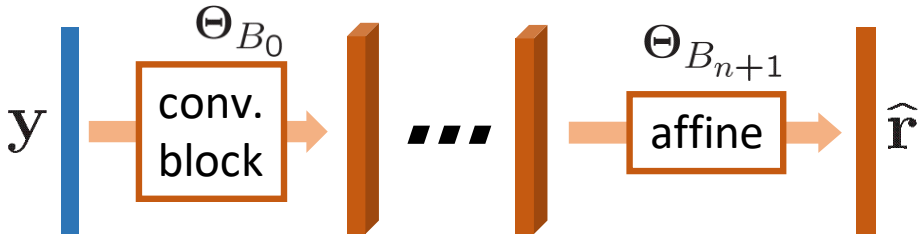
# Training a differentiable sorter

## Sorter architecture:

- LSTM based

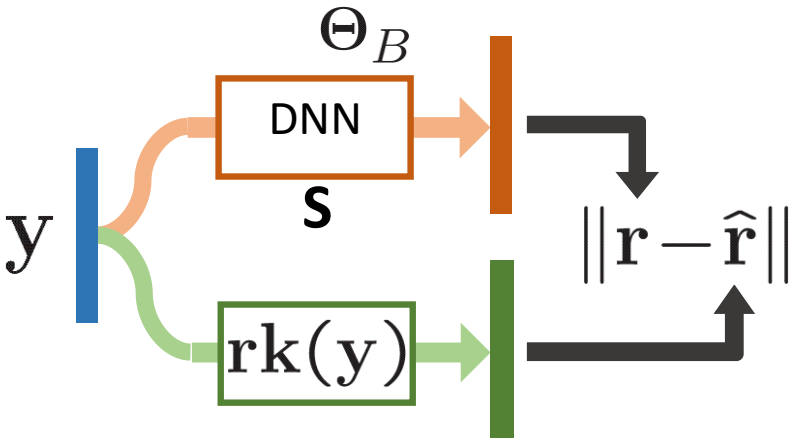



- Convolution based




## Using only synthetic data:

- Uniform distribution over [-1,1]
- Normal distribution with $\mu = 0$ and $\sigma = 1$
- Evenly spaced numbers in random sub-range of [-1,1]



# Spearman correlation loss

**Spearman correlation as a loss function:**

- Spearman correlation:

$$\mathrm{r}_s = 1 - \frac{6\|\mathbf{rk}(\mathbf{y}) - \mathbf{rk}(\mathbf{y}')\|_2^2}{d(d^2-1)}$$

# Spearman correlation loss

**Spearman correlation as a loss function:**

- Spearman correlation:

$$\mathrm{r}_s = 1 - \frac{6\|\mathbf{rk}(\mathbf{y}) - \mathbf{rk}(\mathbf{y}')\|_2^2}{d(d^2-1)}$$

- Maximizing spearman correlation:

$$\min_{\boldsymbol{\Theta}_A} \sum_{n=1}^{N} \left\|\mathbf{rk}(\mathbf{y}^{(n)}) - \mathbf{rk}(\mathbf{y}^{*(n)})\right\|_2^2$$

# Spearman correlation loss

**Spearman correlation as a loss function:**

- Spearman correlation:

$$\mathrm{r}_s = 1 - \frac{6\|\mathbf{rk}(\mathbf{y}) - \mathbf{rk}(\mathbf{y}')\|_2^2}{d(d^2-1)}$$

- Maximizing spearman correlation:

$$\min_{\mathbf{\Theta}_A} \sum_{n=1}^{N} \left\|\mathbf{rk}(\mathbf{y}^{(n)}) - \mathbf{rk}(\mathbf{y}^{*(n)})\right\|_2^2$$

- Replacing $\mathbf{rk}$ with the trained sorter:

$$\mathcal{L}_{SPR}(\mathbf{\Theta}_A, \mathcal{B}) = \sum_{n=1}^{N} \left\| f_{\mathbf{\Theta}_B}(\mathbf{y}(\mathbf{\Theta}_A)^{(n)}) - \mathbf{rk}(\mathbf{y}^{*(n)}) \right\|_2^2$$

# Experiments

**Object recognition:** Evaluated on the Pascal VOC 2007 challenge



| Model | mAP |
|---|---|
| VGG 16 | 89.3% |
| SoDeep | 94.0% |

**Memorability prediction:**


| Model | Spear. corr. |
|---|---|
| Baseline | 46.0% |
| MSE loss | 48.6% |
| SoDeep | **49.4%** |

# Experiments

**Cross modal retrieval:** Evaluated on MS-CoCo image/caption pairs

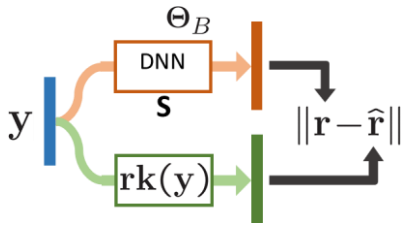
Query: A cat on a sofa



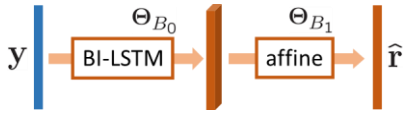

| | Caption retrieval | | | | Image retrieval | | | |
|---|---|---|---|---|---|---|---|---|
| Model | R@1 | R@5 | R@10 | Med. r | R@1 | R@5 | R@10 | Med. r |
| DSVE-Loc | 69.8 | 91.9 | 96.6 | 1 | 55.9 | 86.9 | 94.0 | 1 |
| GXN | 68.5 | - | **97.9** | 1 | **56.6** | - | **94.5** | 1 |
| SoDeep | **71.5** | **92.8** | 97.1 | 1 | 56.2 | **87.0** | 94.3 | 1 |

# Conclusion and Perspectives

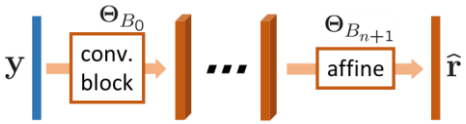
- Learning an approximation of the rank function




- Competitive results on real tasks




- Possiblity to extend to other non-differentiable functions




Thank you for your attention !

Poster #18

SoDeep: A Sorting Deep Net to Learn Ranking Loss Surrogates