Contrasting quadratic assignments for set-based representation learning

Artem Moskalev, Ivan Sosnovik, Volker Fischer, Arnold Smeulders
UvA-Bosch Delta Lab, University of Amsterdam, Netherlands
Bosch Center for Artificial Intelligence (BCAI)

UvA - BOSCH DELTA LAB

ECCV TEL AVIV 2022

UNIVERSITEIT VAN AMSTERDAM

Summary

The standard approach to contrastive learning is to maximize the agreement between different views of the data ordered in pairs.

The approach of **considering individual pairs** cannot account for both intra-set and inter-set similarities when the sets are formed from the views of the data. It thus **limits the information content** of the supervisory signal available to train representations (Figure 2).

We propose to go beyond contrasting individual pairs of objects by focusing on **contrasting objects as sets**. For this, we use combinatorial quadratic assignment theory designed to evaluate set and graph similarities (Figure 1) and derive set-contrastive objective as a regularizer for contrastive learning methods.

Contributions

- We propose to learn representations by contrasting objects as sets rather than as pairs of individual instances.
- We derive a computationally efficient implementation, where the set contrastive part is implemented as a regularizer for existing contrastive learning methods.

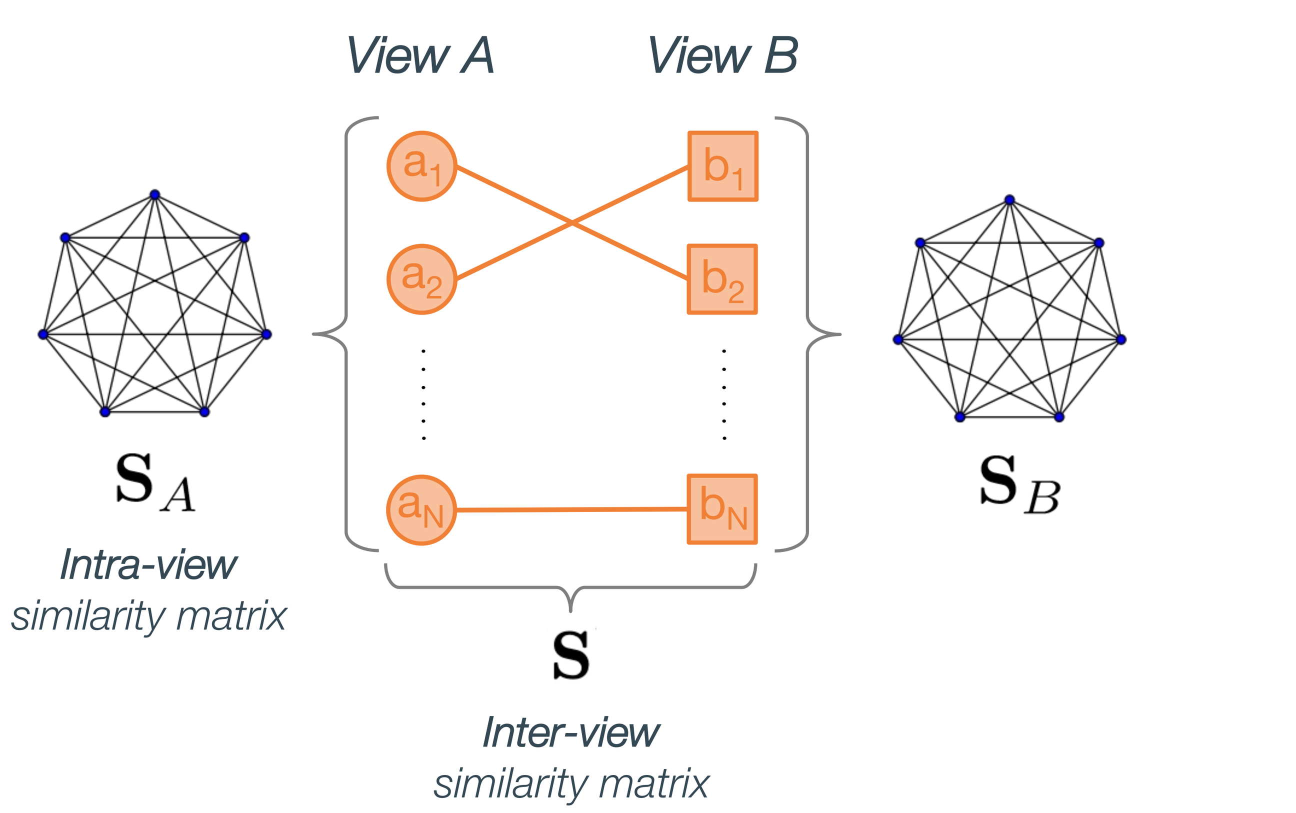
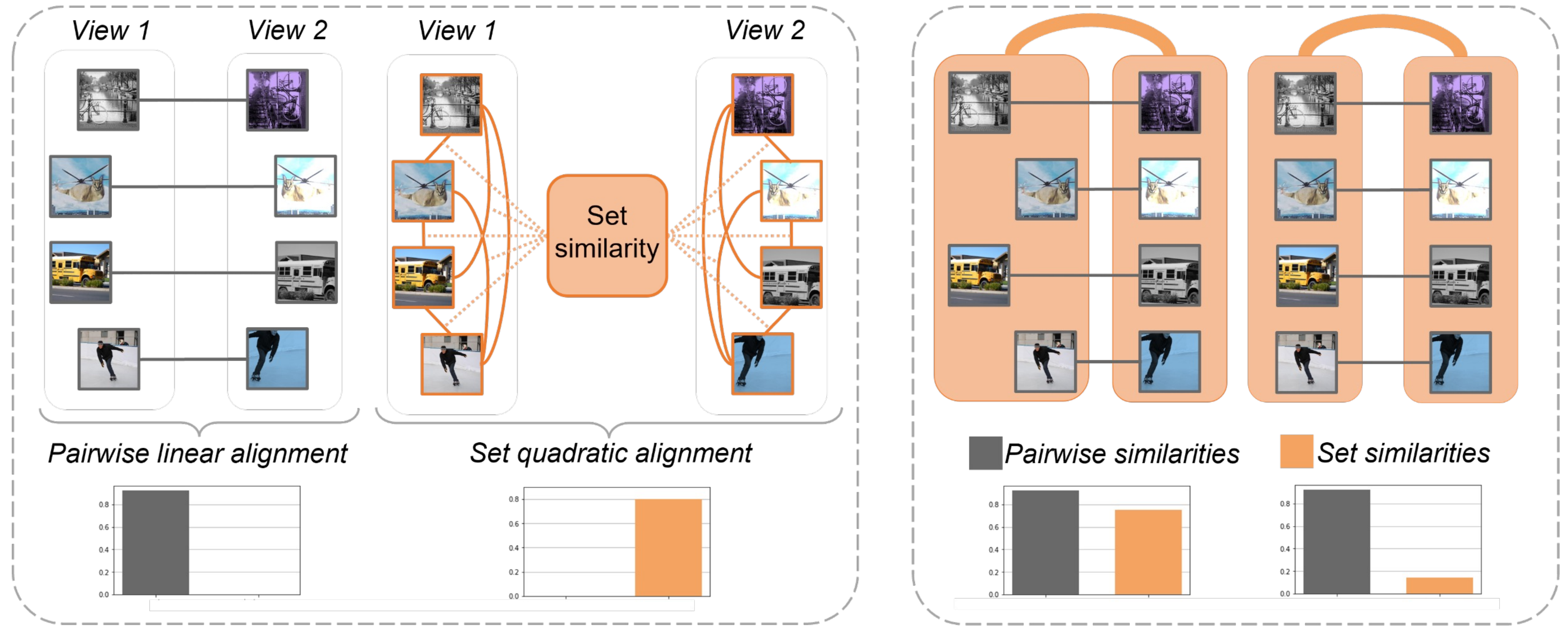


Figure 1:
Linear assignment problem matches individual instances. Quadratic assignment problem matches sets / graphs of instances.



(a) Contrasting pairs and contrasting sets

(b) Comparing alignment costs

Figure 2:
(a) Pairwise similarity versus similarity over sets with set-wise quadratic alignment.
(b) Total pairwise similarities versus set similarities for different configurations of representation graphs. In both configurations, the total similarity over pairs remains the same, being unable to discriminate between the internal structures of different views.

Contrasting quadratic assignments on sets

$$\mathcal{L}(\mathbf{S}) = \max_{\mathbf{Y}\in\Pi}\{tr(\mathbf{S}\mathbf{Y}^T)\}$$

$$L_{LAP}(\mathbf{S},\mathbf{Y}_{gt}) = \mathcal{L}(\mathbf{S}) - tr(\mathbf{S}\mathbf{Y}_{gt}^T)$$

Prior work: rely on linear (pairwise) comparisons

Example: SimCLR

$$\underbrace{\mathcal{L}(\mathbf{S})}_{\text{linear}} \longrightarrow \underbrace{\mathcal{Q}(\mathbf{S},\mathbf{S}_\mathcal{A},\mathbf{S}_\mathcal{B})}_{\text{quadratic}}$$

This work: generalizes to quadratic comparisons

$$\mathcal{Q}(\mathbf{S},\mathbf{S}_\mathcal{A},\mathbf{S}_\mathcal{B}) = \max_{\mathbf{Y}\in\Pi}\{tr(\mathbf{S}\mathbf{Y}^T) + tr(\mathbf{S}_\mathcal{A}\mathbf{Y}\mathbf{S}_\mathcal{B}^T\mathbf{Y}^T)\}$$

$$L_{QAP}(\cdot) = \mathcal{Q}(\mathbf{S},\mathbf{S}_\mathcal{A},\mathbf{S}_\mathcal{B}) - tr(\mathbf{S}\mathbf{Y}_{gt}^T)$$

Approximation through regularization

$$L_{QAP}(\cdot) \le \underbrace{\langle\lambda_\mathcal{A},\lambda_\mathcal{B}\rangle_+}_{\text{Quadratic Assignment Regularization (QARe)}} + \underbrace{\max_{\mathbf{Y}\in\Pi}\{tr(\mathbf{S}\mathbf{Y}^T)\} - tr(\mathbf{S}\mathbf{Y}_{gt}^T)}_{\text{Pairwise contrastive loss (e.g. SimCLR)}}$$

Matching different views

Method	Triplet	SparseCLR	InfoNCE	NTLogistic
pairwise	54.85±0.77	53.03±0.43	57.87±1.19	40.30±0.42
+QARe	**58.48±1.67**	**54.84±1.40**	**61.96±1.38**	**43.48±1.41**
	+3.6%	+1.8%	+4.1%	+3.2%

The goal is to train representations of objects from different views, such that the learned representations provide a correct matching between identities in the dataset. This model is trained by contrasting views of the data (the CUHK03 dataset).

Self-supervised classification

	ResNet-32		
Method	CIFAR-10	CIFAR-100	tiny-ImageNet
Supervised	90.87±0.41	65.32±0.22	50.09±0.32
Random Weights	27.47±0.83	7.65±0.44	3.24±0.43
SimCLR	76.45±0.10	40.18±0.13	23.44±0.27
SparseCLR (Ours)	70.37±0.12	41.05±0.20	25.87±0.59
SimCLR+QARe (Ours)	**76.85±0.39**	41.20±0.34	25.05±0.18
SparseCLR+QARe (Ours)	71.39±0.30	**42.07±0.35**	**27.03±0.40**
	+0.4%	+1.9%	+3.6%

The goal in this experiment is to learn a representation space from the unlabelled data that provides a linear separation between classes in a dataset.

Code: https://github.com/amoskalev/contrasting_quadratic