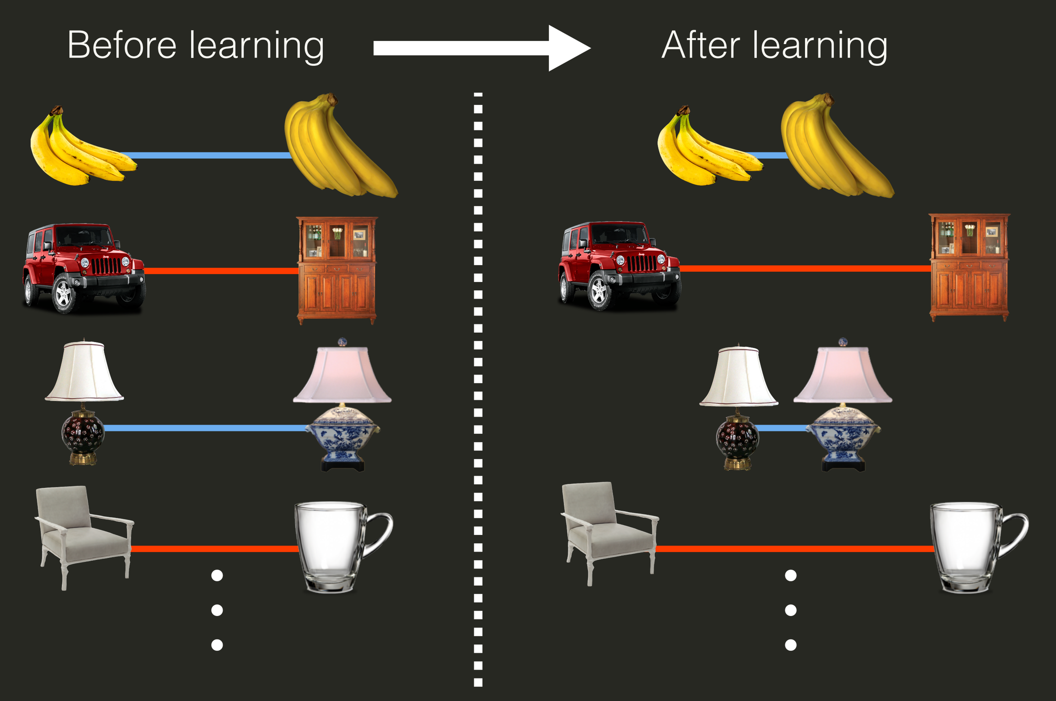
Deep Metric Learning via Facility Location

Hyun Oh Song[1], Stefanie Jegelka[2], Vivek Rathod[1], Kevin Murphy[1]

[1]Google Research, [2]MIT

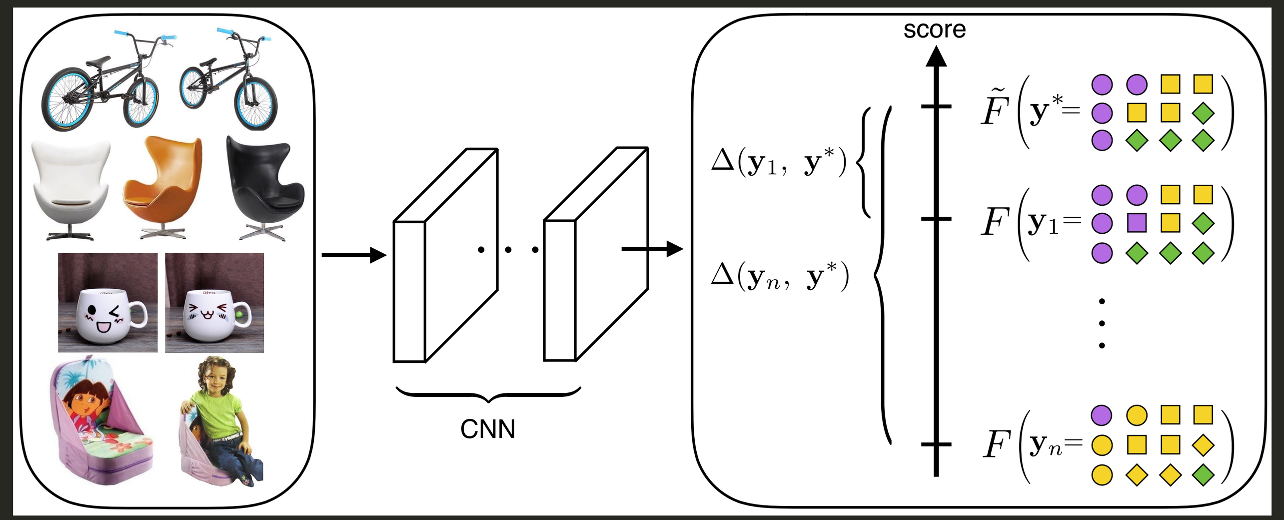
Deep metric learning

- Most deep metric learning methods enforce a local loss function to pull similar examples to each other and push dissimilar examples farther apart in an embedding space.
- However, these approaches suffer from the discrepancy between the training objective, and the actual evaluation metrics used in tasks such as clustering and retrieval.



Deep metric learning via facility location

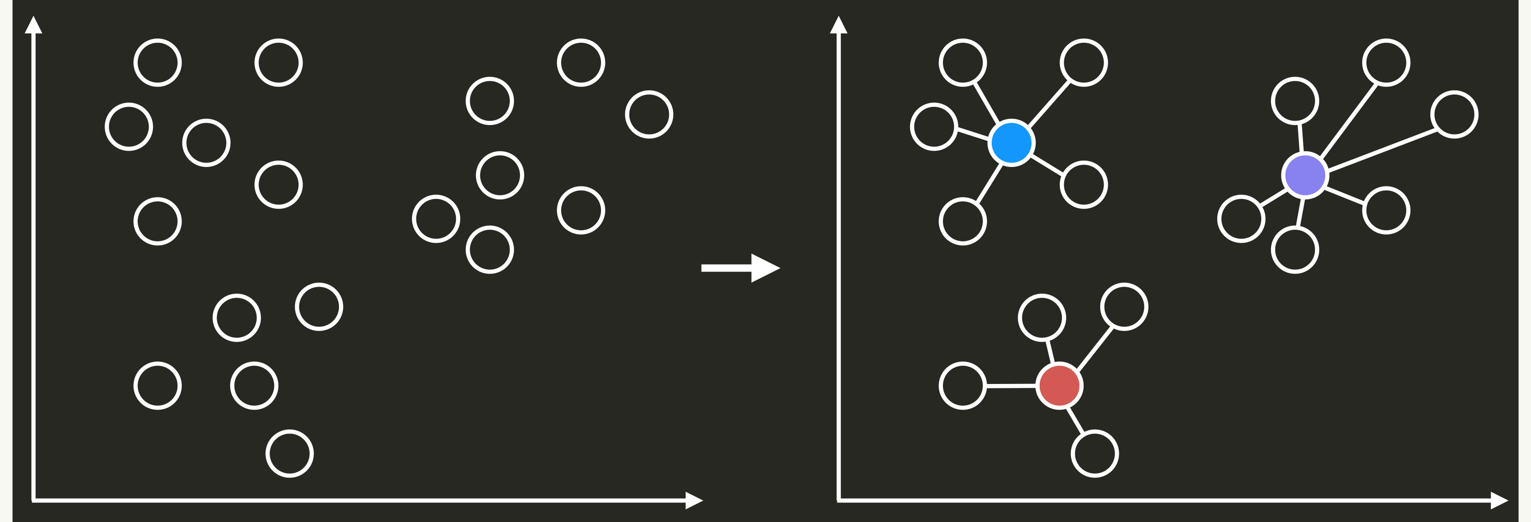
- Learn to cluster a batch of data, and makes use of the evaluation metric such as normalized mutual information during training.



Recent related works

- [Schroff *et al.*] FaceNet: a unified embedding for face recognition and clustering. In *CVPR* 2015
- [Song *et al.*] Deep metric learning via lifted structured feature embedding.
- In *CVPR* 2016
- [Sohn *et al.*] Improved deep metric learning with multi-class n-pair loss objective. In *NIPS* 2016

Facility Location Problem



$$F(S=\{\bullet,\bullet,\bullet\}) = -\sum_{i\in V} \min_{j\in S} D_\Theta(X_i, X_j)$$

Metric Learning via Facility Location

- Enforce the facility location score given the ground truth clustering assignment to be "higher" than the score given any other clustering assignments "at least" by the structured margin delta.

$$\min_{\Theta\in\Omega} \sum_i \left[\max_{\substack{S\subseteq V_i \\ |S|\le k}} F(S;\Theta) + \gamma \Delta_{\mathrm{NMI}}(S, S_i^*) - F(S_i^*;\Theta) \right]_+$$

$\binom{|V_i|}{k}$ possible combinations

$$F : 2^V \to \mathbb{R}, \text{ submodular}$$

$$F(S) = -\sum_{i\in V} \min_{j\in S} D_\Theta(X_i, X_j)$$

Structured margin

$$\underbrace{\Delta(\mathbf{y}, \mathbf{y}^*)}_{\text{Structured margin}} = 1 - \underbrace{\mathrm{NMI}(\mathbf{y}, \mathbf{y}^*)}_{\text{Normalized mutual information}}$$

- Normalized mutual information compares the quality of clustering assignment. (0: worst clustering, 1: perfect clustering)

Loss augmented inference & refinement

- First approximately optimize the loss augmented inference via greedy maximization. This procedure incrementally selects cluster medoids which provide the best marginal benefit to the loss augmented score.
- Then, refine the greedy solution by looping over each clusters and swapping the cluster medoid with any other cluster elements which increases the score.

$$\max_{\substack{S\subseteq V_i \\ |S|\le k}} F(S;\Theta) + \gamma \Delta_{\mathrm{NMI}}(S, S_i^*)$$

Algorithm 1: Loss augmented inference for $(*)$

Input : $X \in \mathbb{R}^{m\times d}$, $\mathbf{y}^* \in |\mathcal{Y}|^m$, γ
Output : $S \subseteq \mathcal{V}$
Initialize: $S = \{\emptyset\}$
Define : $A(S) := F(X, S;\Theta) + \gamma\Delta(g(S), \mathbf{y}^*)$

while $|S| < |\mathcal{Y}|$ **do**
  $i^* = \arg\max_{i\subseteq \mathcal{V}\setminus S} A(S\cup\{i\}) - A(S)$
  $S := S \cup \{i^*\}$
end
return S

Algorithm 2: Loss augmented refinement for $(*)$

Input : $X \in \mathbb{R}^{m\times d}$, $\mathbf{y}^* \in |\mathcal{Y}|^m$, S_{init}, γ, T
Output : S
Initialize: $S = S_{\text{init}}, t = 0$

```
for t < T do
    // Perform cluster assignment
    y_PAM = g(S)
    // Update each medoids per cluster
    for k < |Y| do
        // Swap the current medoid in
           cluster k if it increases the
           score.
        S[k] = argmax_{j ∈ {i: y_PAM[i]=k}} F(X_{i: y_PAM[i]=k}, {j}; Θ)
               + γΔ(g(S \ {S[k]} ∪ {j}), y*)
    end
end
return S
```

$\mathbf{y}_{\text{PAM}} = g(S)$

$$S[k] = \arg\max_{j\in\{i:\, \mathbf{y}_{\text{PAM}}[i]=k\}} F\left(X_{\{i:\, \mathbf{y}_{\text{PAM}}[i]=k\}}, \{j\};\Theta\right) + \gamma\Delta\left(g\left(S\setminus\{S[k]\}\cup\{j\}\right), \mathbf{y}^*\right)$$

Clustering & Retrieval Experiments

CUB200

	NMI	R@1	R@2	R@4	R@8
Triplet semihard (CVPR15)	55.38	42.59	55.03	66.44	77.23
Lifted struct (CVPR16)	56.50	43.57	56.55	68.59	79.63
Npairs (NIPS16)	57.24	45.37	58.41	69.51	79.49
Clustering (Ours)	**59.23**	**48.18**	**61.44**	**71.83**	**81.92**

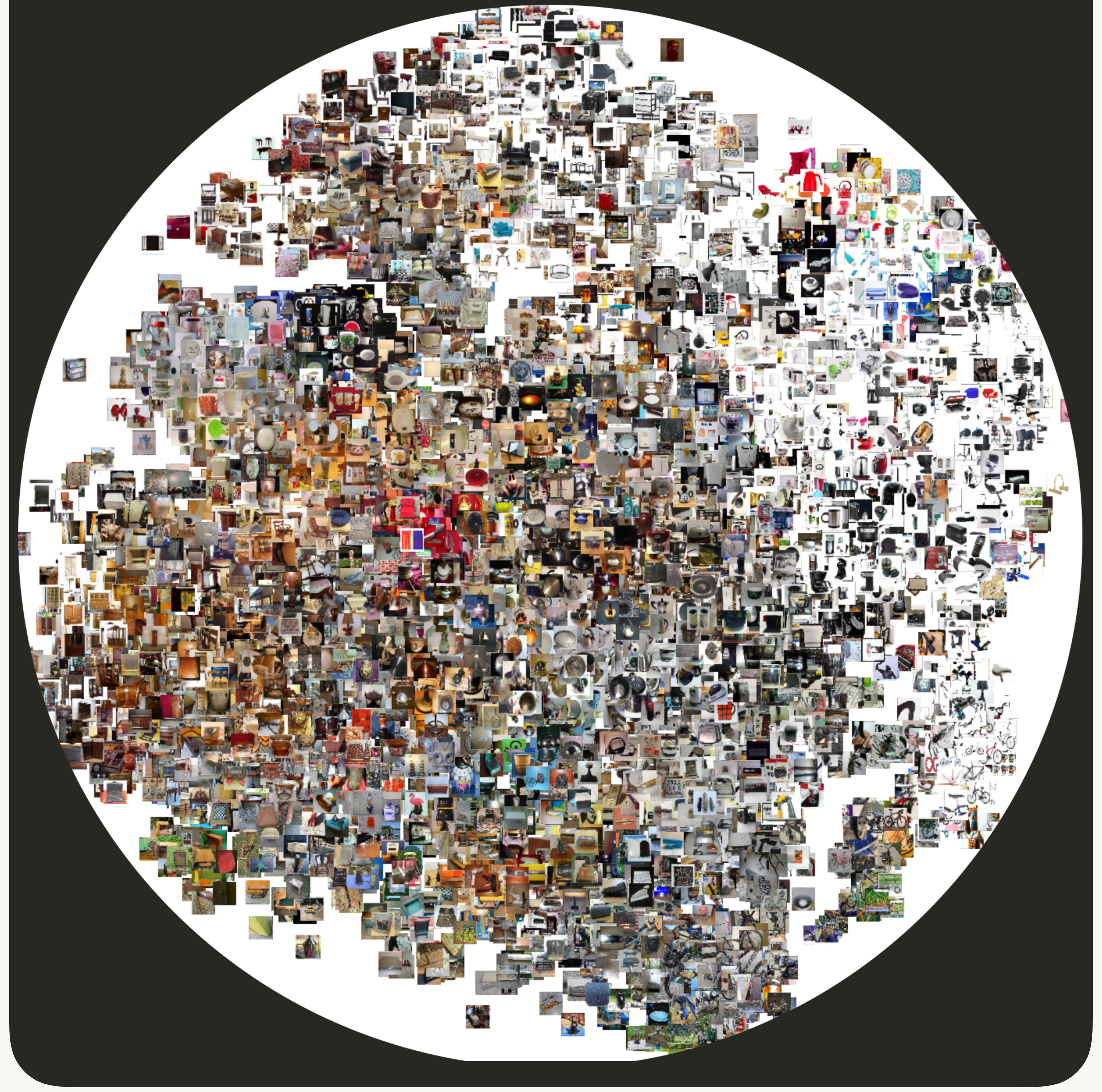
Cars196

	NMI	R@1	R@2	R@4	R@8
Triplet semihard (CVPR15)	53.35	51.54	63.78	73.52	82.41
Lifted struct (CVPR16)	56.88	52.98	65.70	76.01	84.27
Npairs (NIPS16)	57.79	53.90	66.76	77.75	86.35
Clustering (Ours)	**59.04**	**58.11**	**70.64**	**80.27**	**87.81**

Stanford Products

	NMI	R@1	R@10	R@100
Triplet semihard (CVPR15)	89.46	66.67	82.39	91.85
Lifted struct (CVPR16)	88.65	62.46	80.81	91.93
Npairs (NIPS16)	89.37	66.41	83.24	93.00
Clustering (Ours)	**89.48**	**67.02**	**83.65**	**93.23**

tSNE on Stanford Online Products



Summary

- Structural prediction framework for directly optimizing the clustering metric (NMI) with a global view of the embedding space
- State of the art results on clustering and retrieval measured in NMI and recall@K evaluation metrics.

Contact

hyunsong@google.com