unsupervised and semi-supervised learning on manifolds

Yannis Avrithis

Inria Rennes-Bretagne Atlantique

Rennes, March 2019

outline

ranking on manifolds
ranking as smoothing
mining on manifolds
label propagation

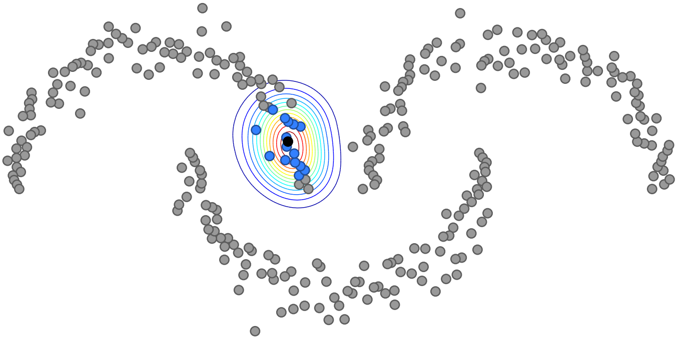
ranking on manifolds

ranking on manifolds

- data points (•), query point (•), nearest neighbors (•)

Zhou, Weston, Gretton, Bousquet and Schölkopf. NIPS 2003. Ranking on Data Manifolds.

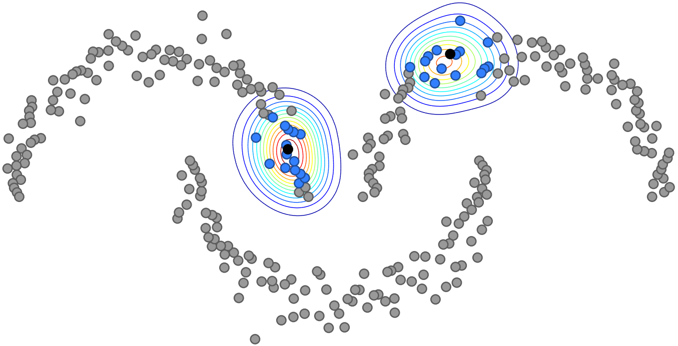
ranking on manifolds: single query



- data points (∘), query point (•), nearest neighbors (∘)

Zhou, Weston, Gretton, Bousquet and Schölkopf. NIPS 2003. Ranking on Data Manifolds.

ranking on manifolds: multiple queries



- data points (∘), query points (•), nearest neighbors (∘)

Zhou, Weston, Gretton, Bousquet and Schölkopf. NIPS 2003. Ranking on Data Manifolds.

ranking on manifolds: random walk

[Zhou et al. 2003]

- reciprocal k-nearest neighbor graph on n data points
- non-negative, symmetric, sparse adjacency matrix $W \in \mathbb{R}^{n \times n}$, with zero diagonal (no self-loops)
- symmetrically normalized adjacency matrix

$$\mathcal{W} := D^{-1/2} W D^{-1/2}$$

  where $D = \mathrm{diag}(W\mathbf{1})$ is the degree matrix
- query: vector $\mathbf{y} \in \mathbb{R}^n$ with $y_i = \mathbb{1}[i \text{ is query}]$
- random walk: starting with any $\mathbf{f}^{(0)} \in \mathbb{R}^n$, iterate

$$\mathbf{f}^{(\tau)} = \alpha \mathcal{W} \mathbf{f}^{(\tau-1)} + (1-\alpha)\mathbf{y}$$

  where $\alpha \in [0, 1)$ (typically close to 1)
- rank data points by descending order of $\mathbf{f}$

Zhou, Weston, Gretton, Bousquet and Schölkopf. NIPS 2003. Ranking on Data Manifolds.

ranking on manifolds: random walk

[Zhou et al. 2003]

- reciprocal k-nearest neighbor graph on n data points
- non-negative, symmetric, sparse adjacency matrix $W \in \mathbb{R}^{n \times n}$, with zero diagonal (no self-loops)
- symmetrically normalized adjacency matrix

$$\mathcal{W} := D^{-1/2} W D^{-1/2}$$

where $D = \operatorname{diag}(W\mathbf{1})$ is the degree matrix

- query: vector $\mathbf{y} \in \mathbb{R}^n$ with $y_i = \mathbb{1}[i \text{ is query}]$
- random walk: starting with any $\mathbf{f}^{(0)} \in \mathbb{R}^n$, iterate

$$\mathbf{f}^{(\tau)} = \alpha \mathcal{W} \mathbf{f}^{(\tau-1)} + (1-\alpha)\mathbf{y}$$

where $\alpha \in [0, 1)$ (typically close to 1)

- rank data points by descending order of $\mathbf{f}$

Zhou, Weston, Gretton, Bousquet and Schölkopf. NIPS 2003. Ranking on Data Manifolds.

ranking on manifolds: random walk

[Zhou et al. 2003]

- reciprocal k-nearest neighbor graph on n data points
- non-negative, symmetric, sparse adjacency matrix $W \in \mathbb{R}^{n \times n}$, with zero diagonal (no self-loops)
- symmetrically normalized adjacency matrix

$$\mathcal{W} := D^{-1/2} W D^{-1/2}$$

  where $D = \operatorname{diag}(W\mathbf{1})$ is the degree matrix
- query: vector $\mathbf{y} \in \mathbb{R}^n$ with $y_i = \mathbb{1}[i \text{ is query}]$
- random walk: starting with any $\mathbf{f}^{(0)} \in \mathbb{R}^n$, iterate

$$\mathbf{f}^{(\tau)} = \alpha \mathcal{W} \mathbf{f}^{(\tau-1)} + (1-\alpha)\mathbf{y}$$

  where $\alpha \in [0, 1)$ (typically close to 1)
- rank data points by descending order of $\mathbf{f}$

Zhou, Weston, Gretton, Bousquet and Schölkopf. NIPS 2003. Ranking on Data Manifolds.

ranking on manifolds: random walk

[Zhou et al. 2003]

- reciprocal k-nearest neighbor graph on n data points
- non-negative, symmetric, sparse adjacency matrix $W \in \mathbb{R}^{n \times n}$, with zero diagonal (no self-loops)
- symmetrically normalized adjacency matrix

$$\mathcal{W} := D^{-1/2} W D^{-1/2}$$

  where $D = \operatorname{diag}(W\mathbf{1})$ is the degree matrix
- query: vector $\mathbf{y} \in \mathbb{R}^n$ with $y_i = \mathbb{1}[i \text{ is query}]$
- random walk: starting with any $\mathbf{f}^{(0)} \in \mathbb{R}^n$, iterate

$$\mathbf{f}^{(\tau)} = \alpha \mathcal{W} \mathbf{f}^{(\tau-1)} + (1-\alpha)\mathbf{y}$$

  where $\alpha \in [0, 1)$ (typically close to 1)
- rank data points by descending order of $\mathbf{f}$

Zhou, Weston, Gretton, Bousquet and Schölkopf. NIPS 2003. Ranking on Data Manifolds.

ranking on manifolds: random walk

[Zhou et al. 2003]

- reciprocal k-nearest neighbor graph on n data points
- non-negative, symmetric, sparse adjacency matrix $W \in \mathbb{R}^{n \times n}$, with zero diagonal (no self-loops)
- symmetrically normalized adjacency matrix

$$\mathcal{W} := D^{-1/2} W D^{-1/2}$$

  where $D = \operatorname{diag}(W\mathbf{1})$ is the degree matrix
- query: vector $\mathbf{y} \in \mathbb{R}^n$ with $y_i = \mathbb{1}[i \text{ is query}]$
- random walk: starting with any $\mathbf{f}^{(0)} \in \mathbb{R}^n$, iterate

$$\mathbf{f}^{(\tau)} = \alpha \mathcal{W} \mathbf{f}^{(\tau-1)} + (1-\alpha)\mathbf{y}$$

  where $\alpha \in [0, 1)$ (typically close to 1)
- rank data points by descending order of $\mathbf{f}$

Zhou, Weston, Gretton, Bousquet and Schölkopf. NIPS 2003. Ranking on Data Manifolds.

ranking as solving a linear system

[Iscen et al. 2017]

- regularized Laplacian

$$\mathcal{L}_\alpha = \frac{I - \alpha \mathcal{W}}{1 - \alpha}$$

- solve linear system

$$\mathcal{L}_\alpha \mathbf{f} = \mathbf{y}$$

by conjugate gradient (CG) method

Iscen, Tolias, Avrithis, Furon and Chum. CVPR 2017. Efficient Diffusion on Region Manifolds: Recovering Small Objects With Compact CNN Representations.

ranking as solving a linear system

[Iscen et al. 2017]

- regularized Laplacian

$$\mathcal{L}_\alpha = \frac{I - \alpha\mathcal{W}}{1 - \alpha}$$

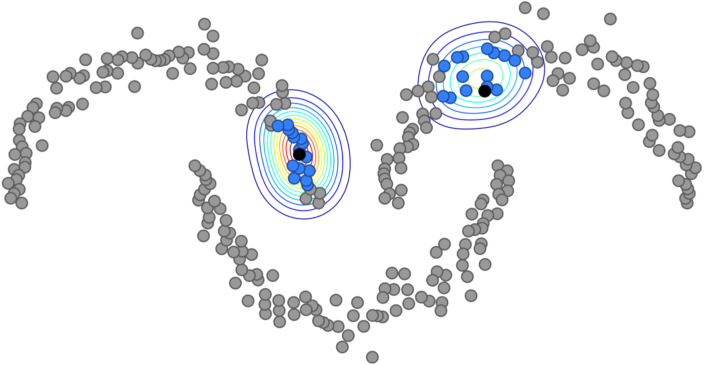
- solve linear system

$$\mathcal{L}_\alpha \mathbf{f} = \mathbf{y}$$

by conjugate gradient (CG) method

Iscen, Tolias, Avrithis, Furon and Chum. CVPR 2017. Efficient Diffusion on Region Manifolds: Recovering Small Objects With Compact CNN Representations.

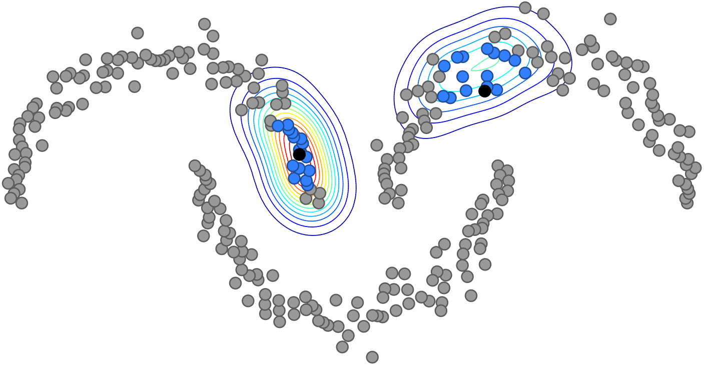
ranking by conjugate gradient (CG)



- data points (∘), query points (•), nearest neighbors (∘)
- iteration 0×2

Iscen, Tolias, Avrithis, Furon and Chum. CVPR 2017. Efficient Diffusion on Region Manifolds: Recovering Small Objects With Compact CNN Representations.

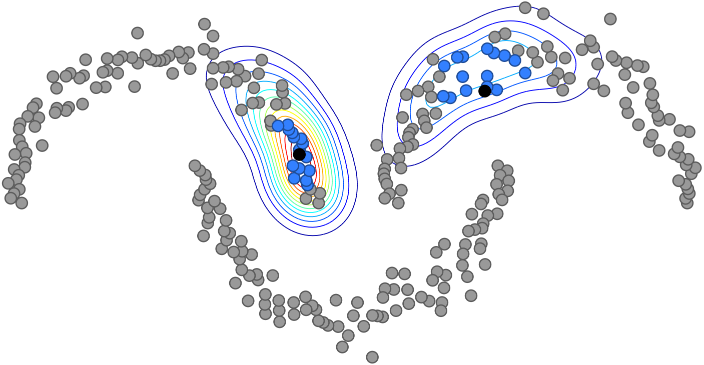
ranking by conjugate gradient (CG)



- data points (∘), query points (•), nearest neighbors (∘)
- iteration 1×2

Iscen, Tolias, Avrithis, Furon and Chum. CVPR 2017. Efficient Diffusion on Region Manifolds: Recovering Small Objects With Compact CNN Representations.

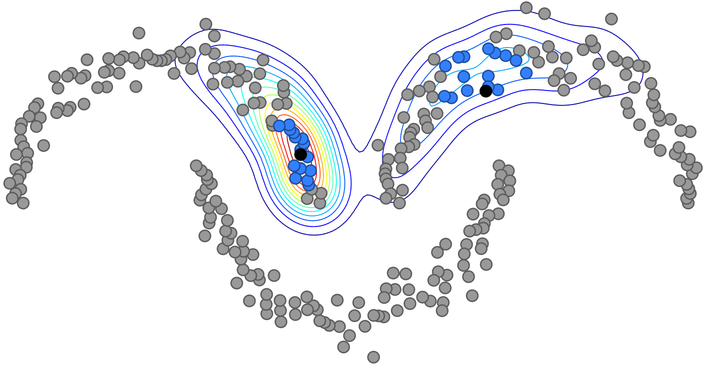
ranking by conjugate gradient (CG)



- data points (∘), query points (•), nearest neighbors (∘)
- iteration 2×2

Iscen, Tolias, Avrithis, Furon and Chum. CVPR 2017. Efficient Diffusion on Region Manifolds: Recovering Small Objects With Compact CNN Representations.

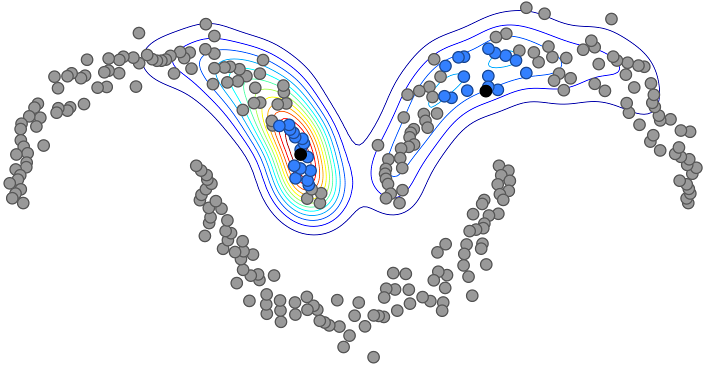
ranking by conjugate gradient (CG)



- data points (∘), query points (•), nearest neighbors (•)
- iteration 3×2

Iscen, Tolias, Avrithis, Furon and Chum. CVPR 2017. Efficient Diffusion on Region Manifolds: Recovering Small Objects With Compact CNN Representations.

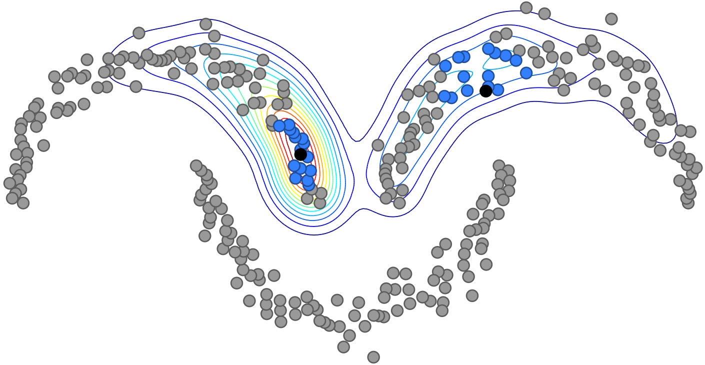
ranking by conjugate gradient (CG)



- data points (∘), query points (•), nearest neighbors (•)
- iteration 4×2

Iscen, Tolias, Avrithis, Furon and Chum. CVPR 2017. Efficient Diffusion on Region Manifolds: Recovering Small Objects With Compact CNN Representations.

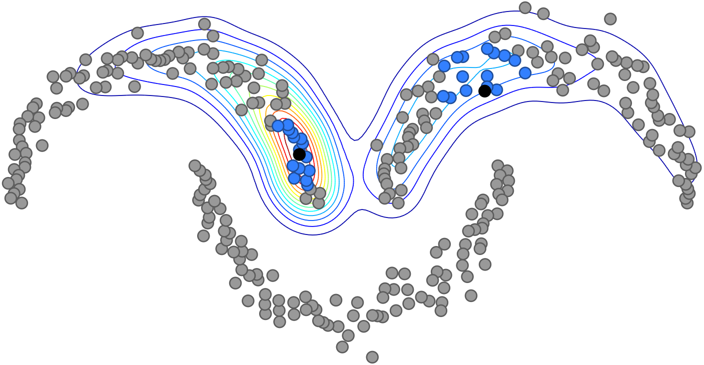
ranking by conjugate gradient (CG)



- data points (•), query points (•), nearest neighbors (•)
- iteration 5×2

Iscen, Tolias, Avrithis, Furon and Chum. CVPR 2017. Efficient Diffusion on Region Manifolds: Recovering Small Objects With Compact CNN Representations.

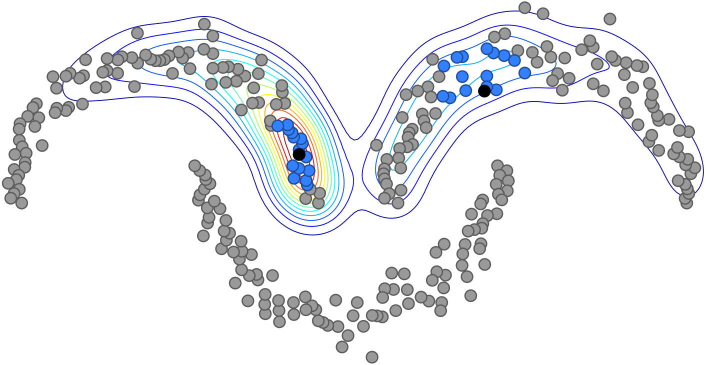
ranking by conjugate gradient (CG)



- data points (•), query points (•), nearest neighbors (•)
- iteration 6×2

Iscen, Tolias, Avrithis, Furon and Chum. CVPR 2017. Efficient Diffusion on Region Manifolds: Recovering Small Objects With Compact CNN Representations.

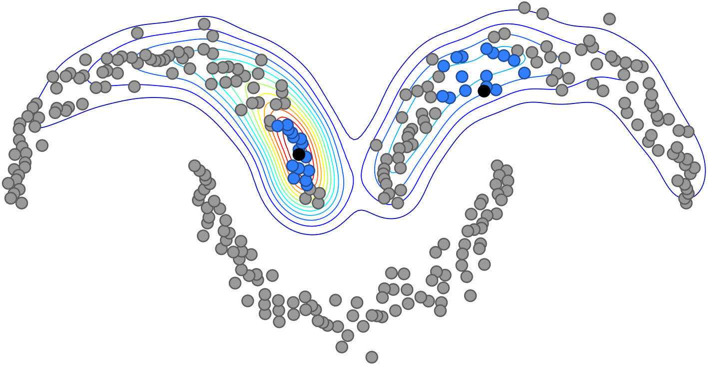
ranking by conjugate gradient (CG)



- data points (∘), query points (•), nearest neighbors (∘)
- iteration 7×2

Iscen, Tolias, Avrithis, Furon and Chum. CVPR 2017. Efficient Diffusion on Region Manifolds: Recovering Small Objects With Compact CNN Representations.

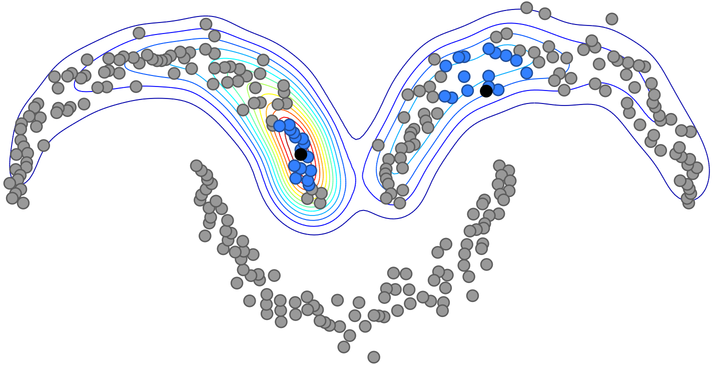
ranking by conjugate gradient (CG)



- data points (∘), query points (•), nearest neighbors (∘)
- iteration 8×2

Iscen, Tolias, Avrithis, Furon and Chum. CVPR 2017. Efficient Diffusion on Region Manifolds: Recovering Small Objects With Compact CNN Representations.

ranking by conjugate gradient (CG)



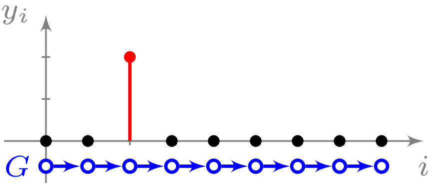
- data points (•), query points (•), nearest neighbors (•)
- iteration 9×2

Iscen, Tolias, Avrithis, Furon and Chum. CVPR 2017. Efficient Diffusion on Region Manifolds: Recovering Small Objects With Compact CNN Representations.

ranking as smoothing

ranking on manifolds as smoothing

[Iscen et al. 2018]

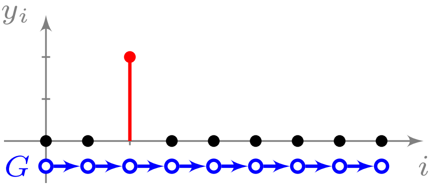


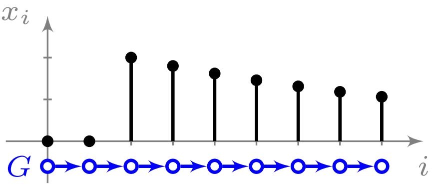
- exponential moving average filter
- output given by $x_i := (1-\alpha)\sum_{t=0}^{\infty} \alpha^t y_{i-t}$
- or by recurrence $x_i = \alpha x_{i-1} + (1-\alpha) y_i$
- impulse response $h_i = (1-\alpha)\alpha^i u_i$
- transfer function $H(z) := (1-\alpha)\sum_{t=0}^{\infty}(az^{-1})^t = (1-\alpha)/(1-\alpha z^{-1})$

Iscen, Avrithis, Tolias, Furon, Chum. CVPR 2018. Fast Spectral Ranking for Similarity Search.

ranking on manifolds as smoothing

[Iscen et al. 2018]

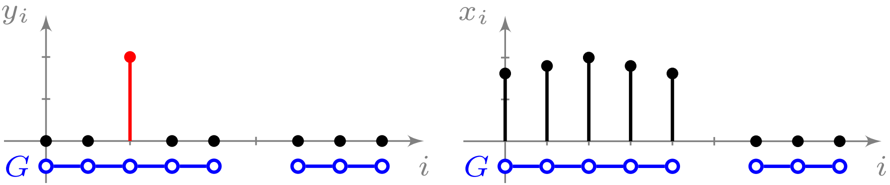




- exponential moving average filter
- output given by $x_i := (1-\alpha)\sum_{t=0}^{\infty} \alpha^t y_{i-t}$
- or by recurrence $x_i = \alpha x_{i-1} + (1-\alpha) y_i$
- impulse response $h_i = (1-\alpha)\alpha^i u_i$
- transfer function $H(z) := (1-\alpha)\sum_{t=0}^{\infty}(az^{-1})^t = (1-\alpha)/(1-\alpha z^{-1})$

ranking on manifolds as smoothing

[Iscen et al. 2018]



- using a weighted undirected graph G instead
- information "flows" in all directions, controlled by edge weights

Iscen, Avrithis, Tolias, Furon, Chum. CVPR 2018. Fast Spectral Ranking for Similarity Search.

ranking on manifolds as smoothing

- express $\mathcal{L}_\alpha^{-1}$ using a transfer function

$$\mathcal{L}_\alpha^{-1} = h_\alpha(\mathcal{W}) = (1-\alpha)(I - \alpha\mathcal{W})^{-1}$$

- given any matrix function h, we want to compute

$$\mathbf{x} = h(\mathcal{W})\mathbf{y}$$

  without computing $h(\mathcal{W})$
- which we do by eigenvalue decomposition and low-rank approximation of matrix $h(\mathcal{W})$, without ever computing the matrix itself

ranking on manifolds as smoothing

- express $\mathcal{L}_\alpha^{-1}$ using a transfer function

$$\mathcal{L}_\alpha^{-1} = h_\alpha(\mathcal{W}) = (1-\alpha)(I-\alpha\mathcal{W})^{-1}$$

- given any matrix function h, we want to compute

$$\mathbf{x} = h(\mathcal{W})\mathbf{y}$$

  without computing $h(\mathcal{W})$

- which we do by eigenvalue decomposition and low-rank approximation of matrix $h(\mathcal{W})$, without ever computing the matrix itself

Iscen, Avrithis, Tolias, Furon, Chum. CVPR 2018. Fast Spectral Ranking for Similarity Search.

ranking on manifolds as smoothing

- express $\mathcal{L}_\alpha^{-1}$ using a transfer function

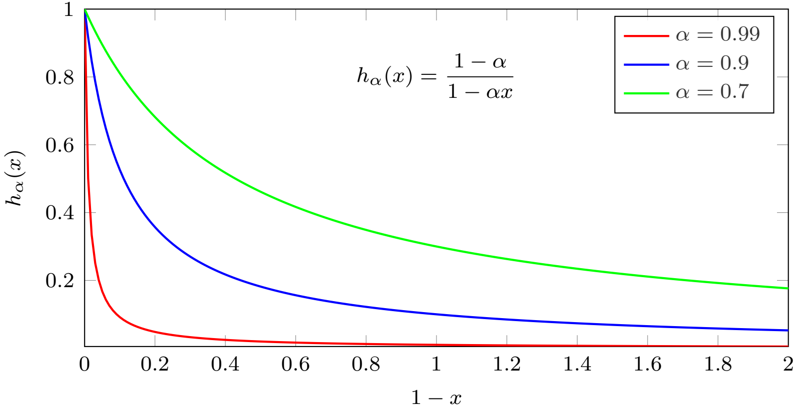
$$\mathcal{L}_\alpha^{-1} = h_\alpha(\mathcal{W}) = (1-\alpha)(I - \alpha\mathcal{W})^{-1}$$

- given any matrix function h, we want to compute

$$\mathbf{x} = h(\mathcal{W})\mathbf{y}$$

  without computing $h(\mathcal{W})$
- which we do by eigenvalue decomposition and low-rank approximation of matrix $h(\mathcal{W})$, without ever computing the matrix itself

interpretation: graph signal processing



- low-pass filtering in the frequency domain
- or, "soft" dimensionality reduction

Iscen, Avrithis, Tolias, Furon, Chum. CVPR 2018. Fast Spectral Ranking for Similarity Search.

mining on manifolds

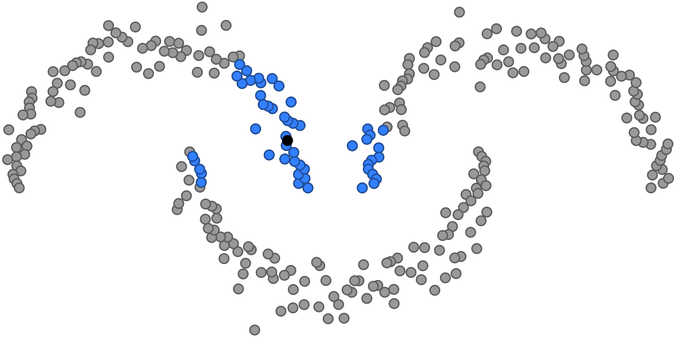
mining on manifolds

[Iscen et al. 2018]

- data points (∘), query point $\mathbf{x}$ (•)

mining on manifolds

[Iscen et al. 2018]



- data points (∘), query point $\mathbf{x}$ (•)
- Euclidean nearest neighbors $E(\mathbf{x})$ (∘)

mining on manifolds

[Iscen et al. 2018]

- data points (∘), query point $\mathbf{x}$ (•)
- manifold nearest neighbors $M(\mathbf{x})$ (∘)

mining on manifolds

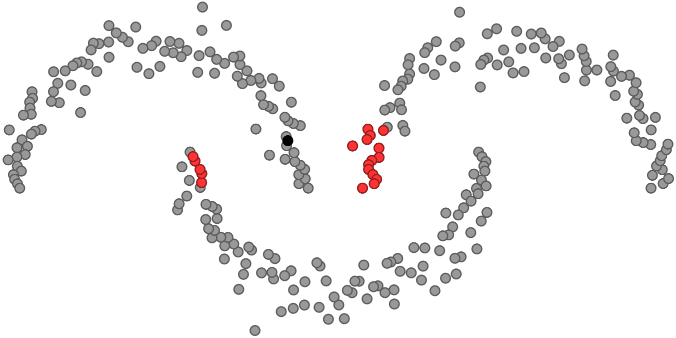
[Iscen et al. 2018]

- data points (∘), query point $\mathbf{x}$ (•)
- hard positives $S^+ = M(\mathbf{x}) \setminus E(\mathbf{x})$ (∘)

Iscen, Tolias, Avrithis and Chum. CVPR 2018. Mining on Manifolds: Metric Learning without Labels.

mining on manifolds

[Iscen et al. 2018]



- data points (∘), query point $\mathbf{x}$ (•)
- hard negatives $S^- = E(\mathbf{x}) \setminus M(\mathbf{x})$ (•)

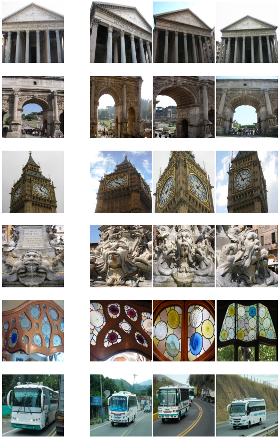
Iscen, Tolias, Avrithis and Chum. CVPR 2018. Mining on Manifolds: Metric Learning without Labels.

hard positive/negative examples

- query (anchor) $(\mathbf{x})$
- positives $S^+(\mathbf{x})$ *vs.* Euclidean neighbors $E(\mathbf{x})$
- negatives $S^-(\mathbf{x})$ *vs.* Euclidean non-neighbors $X \setminus E(\mathbf{x})$

Iscen, Tolias, Avrithis and Chum. CVPR 2018. Mining on Manifolds: Metric Learning without Labels.

hard positive/negative examples



- query (anchor) $(\mathbf{x})$
- positives $S^+(\mathbf{x})$ *vs.* Euclidean neighbors $E(\mathbf{x})$
- negatives $S^-(\mathbf{x})$ *vs.* Euclidean non-neighbors $X \setminus E(\mathbf{x})$

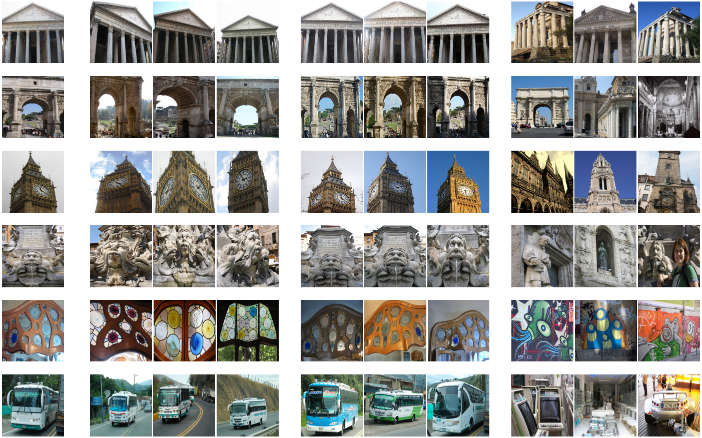
Iscen, Tolias, Avrithis and Chum. CVPR 2018. Mining on Manifolds: Metric Learning without Labels.

hard positive/negative examples

- query (anchor) $(\mathbf{x})$
- positives $S^+(\mathbf{x})$ *vs.* Euclidean neighbors $E(\mathbf{x})$
- negatives $S^-(\mathbf{x})$ *vs.* Euclidean non-neighbors $X \setminus E(\mathbf{x})$

Iscen, Tolias, Avrithis and Chum. CVPR 2018. Mining on Manifolds: Metric Learning without Labels.

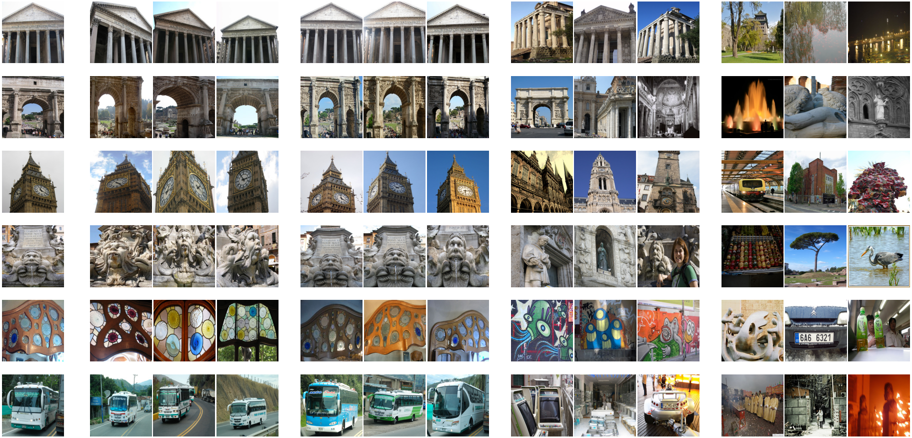
hard positive/negative examples



- query (anchor) $(\mathbf{x})$
- positives $S^+(\mathbf{x})$ *vs.* Euclidean neighbors $E(\mathbf{x})$
- negatives $S^-(\mathbf{x})$ *vs.* Euclidean non-neighbors $X \setminus E(\mathbf{x})$

Iscen, Tolias, Avrithis and Chum. CVPR 2018. Mining on Manifolds: Metric Learning without Labels.

hard positive/negative examples



- query (anchor) $(\mathbf{x})$
- positives $S^+(\mathbf{x})$ *vs.* Euclidean neighbors $E(\mathbf{x})$
- negatives $S^-(\mathbf{x})$ *vs.* Euclidean non-neighbors $X \setminus E(\mathbf{x})$

Iscen, Tolias, Avrithis and Chum. CVPR 2018. Mining on Manifolds: Metric Learning without Labels.

fine-tuning with hard example mining

- pre-train network
- extract descriptors on unlabeled dataset
- construct nearest neighbor graph
- sample anchors, measure Euclidean and manifold distances
- sample positives and negatives
- fine-tune using contrastive or triplet loss

Iscen, Tolias, Avrithis and Chum. CVPR 2018. Mining on Manifolds: Metric Learning without Labels.

fine-tuning with hard example mining

- pre-train network
- extract descriptors on unlabeled dataset
- construct nearest neighbor graph
- sample anchors, measure Euclidean and manifold distances
- sample positives and negatives
- fine-tune using contrastive or triplet loss

Iscen, Tolias, Avrithis and Chum. CVPR 2018. Mining on Manifolds: Metric Learning without Labels.

fine-grained categorization results

Method	Labels	R@1	R@2	R@4	R@8	NMI
Initial	No	35.0	46.8	59.3	72.0	48.1
Triplet+semi-hard	Yes	42.3	55.0	66.4	77.2	55.4
Lifted-Structure	Yes	43.6	56.6	68.6	79.6	56.5
Triplet+	Yes	45.9	57.7	69.6	79.8	58.1
Clustering	Yes	48.2	61.4	71.8	81.9	59.2
Triplet+++	Yes	49.8	62.3	74.1	83.3	59.9
Cyclic match	No	40.8	52.8	65.1	76.0	52.6
Ours	No	45.3	57.8	68.6	78.4	55.0

- CUB200-2011 dataset, 200 bird species, 100 training / 100 testing
- GoogLeNet pre-trained on ImageNet, then fine-tuned with triplet loss

particular object retrieval results

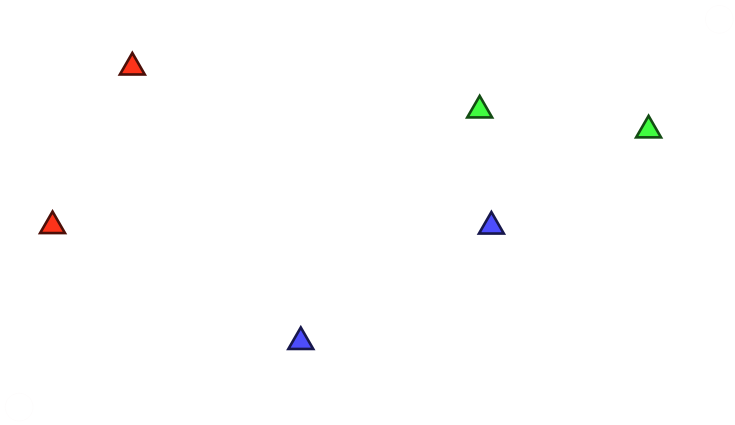
Model	Pooling	Labels	Oxf5k	Oxf105k	Par6k	Par106k	Hol	Instre
ImageNet	MAC	Human	58.5	50.3	73.0	59.0	79.4	48.5
From BoW		SfM	**79.7**	73.9	82.4	74.6	81.4	48.5
Ours		—	78.7	**74.3**	**83.1**	**75.6**	**82.6**	**55.5**
ImageNet	R-MAC	Human	68.0	61.0	76.6	72.1	87.0	55.6
From BoW		SfM	77.8	70.1	84.1	76.8	84.4	47.7
Ours		—	**78.2**	**72.6**	**85.1**	**78.0**	**87.5**	**57.7**

- VGG-16 pre-trained on ImageNet, then fine-tuned with constrastive loss on a 1M unlabeled dataset with MAC representation
- at test time, either MAC or R-MAC used

label propagation

semi-supervised learning

[Zhou et al. 2003]

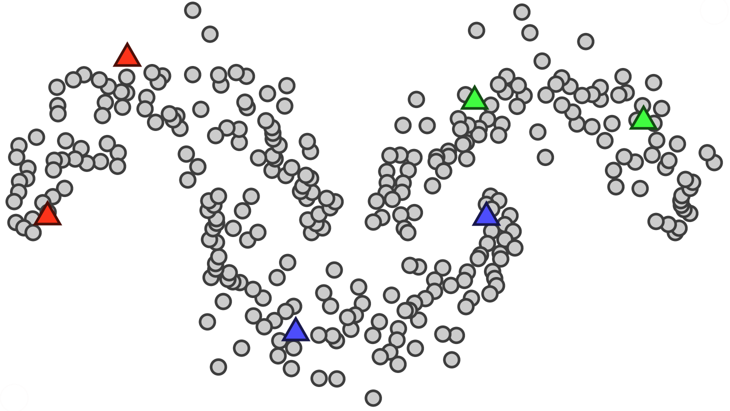


- labeled points (▲), unlabeled points $\mathbf{x}$ (○)

Zhou, Bousquet, Lal, Weston, Schölkopf. NIPS2003. Learning with Local and Global Consistency.

semi-supervised learning

[Zhou et al. 2003]



- labeled points (▲), unlabeled points $\mathbf{x}$ (∘)

Zhou, Bousquet, Lal, Weston, Schölkopf. NIPS2003. Learning with Local and Global Consistency.

label propagation (transductive)

- same graph representation as in manifold ranking

$$\mathcal{W} := D^{-1/2} W D^{-1/2}$$

- given labeled examples L and unlabeled examples U
- label matrix Y with elements

$$y_{ij} := \begin{cases} 1, & \text{if } i \in L \wedge y_i = j \\ 0, & \text{otherwise,} \end{cases}$$

- label propagation, again by CG

$$Z := (I - \alpha \mathcal{W})^{-1} Y$$

- prediction for unlabelled example x_i

$$\hat{y}_i := \arg\max_j z_{ij}$$

Zhou, Bousquet, Lal, Weston, Schölkopf. NIPS2003. Learning with Local and Global Consistency.

label propagation (transductive)

- same graph representation as in manifold ranking

$$\mathcal{W} := D^{-1/2} W D^{-1/2}$$

- given labeled examples L and unlabeled examples U
- label matrix Y with elements

$$y_{ij} := \begin{cases} 1, & \text{if } i \in L \wedge y_i = j \\ 0, & \text{otherwise,} \end{cases}$$

- label propagation, again by CG

$$Z := (I - \alpha \mathcal{W})^{-1} Y$$

- prediction for unlabelled example x_i

$$\hat{y}_i := \arg\max_j z_{ij}$$

Zhou, Bousquet, Lal, Weston, Schölkopf. NIPS2003. Learning with Local and Global Consistency.

label propagation (transductive)

- same graph representation as in manifold ranking

$$\mathcal{W} := D^{-1/2} W D^{-1/2}$$

- given labeled examples L and unlabeled examples U
- label matrix Y with elements

$$y_{ij} := \begin{cases} 1, & \text{if } i \in L \wedge y_i = j \\ 0, & \text{otherwise,} \end{cases}$$

- label propagation, again by CG

$$Z := (I - \alpha \mathcal{W})^{-1} Y$$

- prediction for unlabelled example x_i

$$\hat{y}_i := \arg\max_j z_{ij}$$

Zhou, Bousquet, Lal, Weston, Schölkopf. NIPS2003. Learning with Local and Global Consistency.

label propagation (transductive)

- same graph representation as in manifold ranking

$$\mathcal{W} := D^{-1/2} W D^{-1/2}$$

- given labeled examples L and unlabeled examples U
- label matrix Y with elements

$$y_{ij} := \begin{cases} 1, & \text{if } i \in L \wedge y_i = j \\ 0, & \text{otherwise,} \end{cases}$$

- label propagation, again by CG

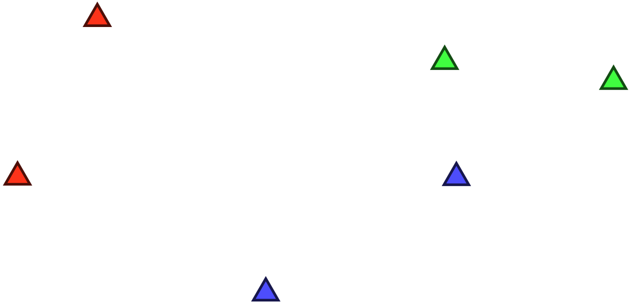
$$Z := (I - \alpha \mathcal{W})^{-1} Y$$

- prediction for unlabelled example x_i

$$\hat{y}_i := \arg\max_j z_{ij}$$

Zhou, Bousquet, Lal, Weston, Schölkopf. NIPS2003. Learning with Local and Global Consistency.

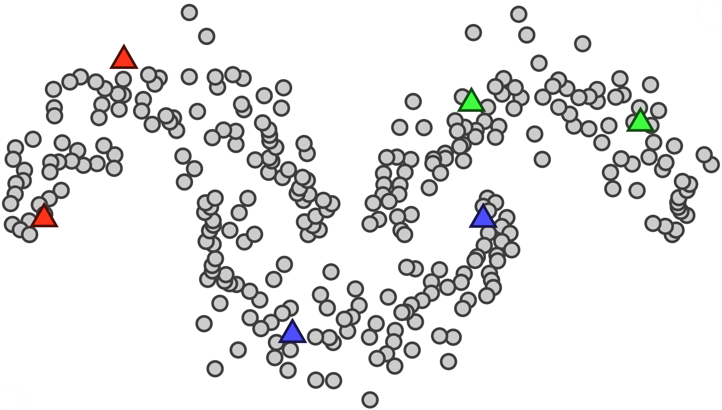
label propagation (transductive)



- labeled points (▲), unlabeled points $\mathbf{x}$ (●)
- propagated labels (●), certainty of prediction

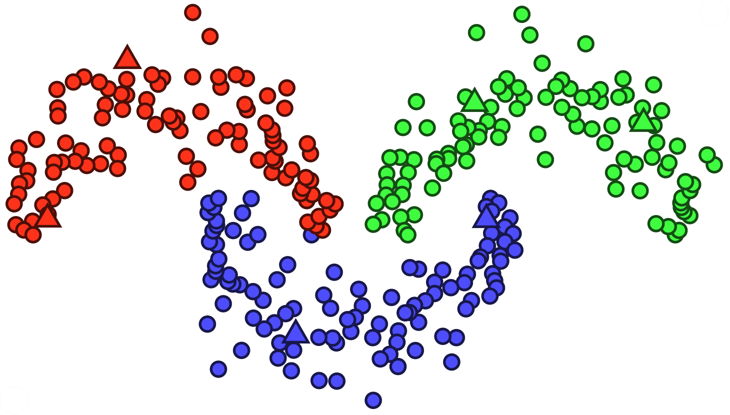
Zhou, Bousquet, Lal, Weston, Schölkopf. NIPS2003. Learning with Local and Global Consistency.

label propagation (transductive)



- labeled points (▲), unlabeled points $\mathbf{x}$ (●)
- propagated labels (●), certainty of prediction

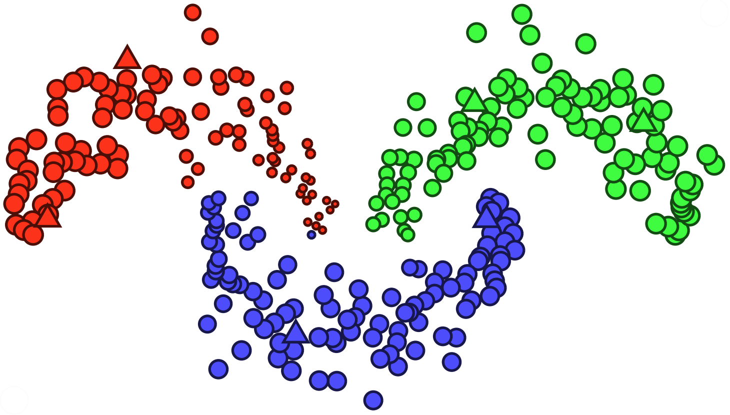
label propagation (transductive)



- labeled points (▲), unlabeled points $\mathbf{x}$ (●)
- propagated labels (●), certainty of prediction

Zhou, Bousquet, Lal, Weston, Schölkopf. NIPS2003. Learning with Local and Global Consistency.

label propagation (transductive)



- labeled points (▲), unlabeled points $\mathbf{x}$ (●)
- propagated labels (●), certainty of prediction

Zhou, Bousquet, Lal, Weston, Schölkopf. NIPS2003. Learning with Local and Global Consistency.

label propagation (inductive)

[Iscen et al. 2019]

- given labeled examples X_L, unlabeled examples X_U with $x_i \in \mathcal{X}$, and labels Y_L with $y_i \in C = \{1, \dots, c\}$
- we now want to learn
  - an explicit feature map $\phi_\theta : \mathcal{X} \to \mathbb{R}^d$
  - a classifier $f_\theta : \mathcal{X} \to \mathbb{R}^c$, consisting of ϕ_θ followed by a fully-connected (FC) layer and softmax

Iscen, Tolias, Avrithis and Chum. CVPR 2019. Label Propagation for Deep Semi-supervised Learning.

label propagation (inductive)

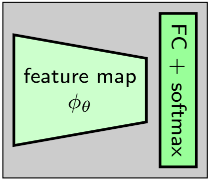
[Iscen et al. 2019]

- given labeled examples X_L, unlabeled examples X_U with $x_i \in \mathcal{X}$, and labels Y_L with $y_i \in C = \{1, \ldots, c\}$
- we now want to learn
  - an explicit feature map $\phi_\theta : \mathcal{X} \to \mathbb{R}^d$
  - a classifier $f_\theta : \mathcal{X} \to \mathbb{R}^c$, consisting of ϕ_θ followed by a fully-connected (FC) layer and softmax

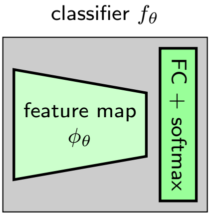
Iscen, Tolias, Avrithis and Chum. CVPR 2019. Label Propagation for Deep Semi-supervised Learning.

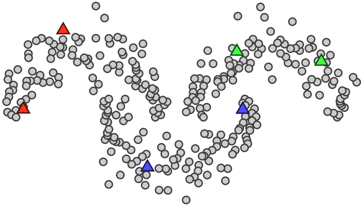
label propagation (inductive)

classifier f_θ



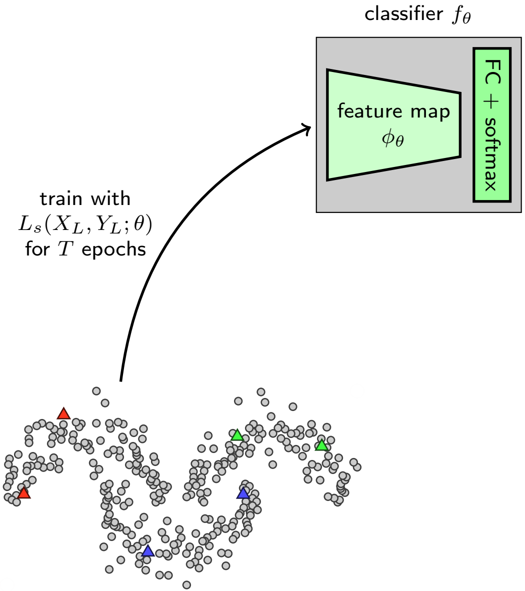
label propagation (inductive)





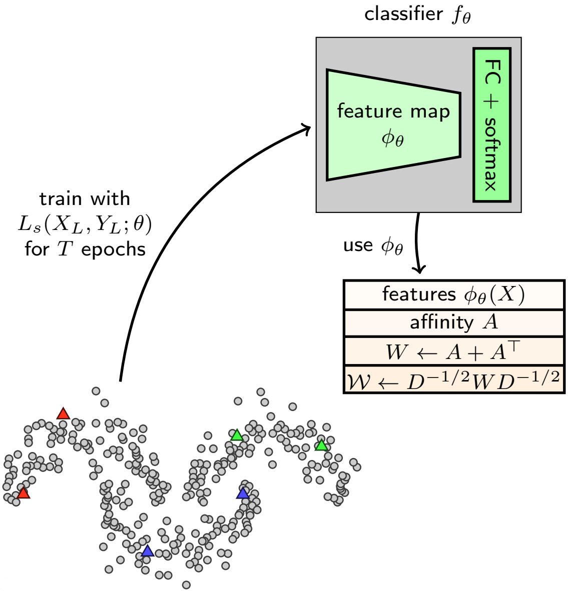
Iscen, Tolias, Avrithis and Chum. CVPR 2019. Label Propagation for Deep Semi-supervised Learning.

label propagation (inductive)



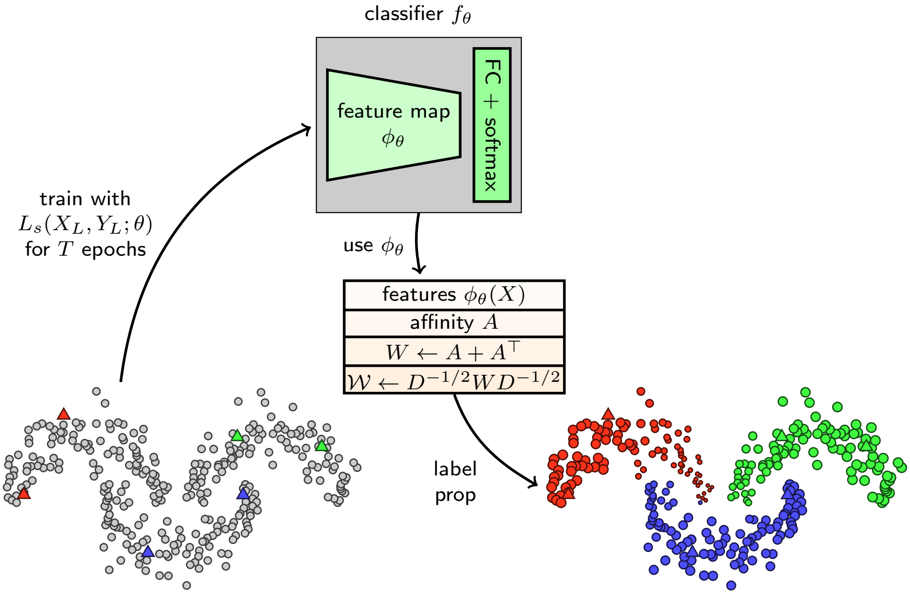
Iscen, Tolias, Avrithis and Chum. CVPR 2019. Label Propagation for Deep Semi-supervised Learning.

label propagation (inductive)



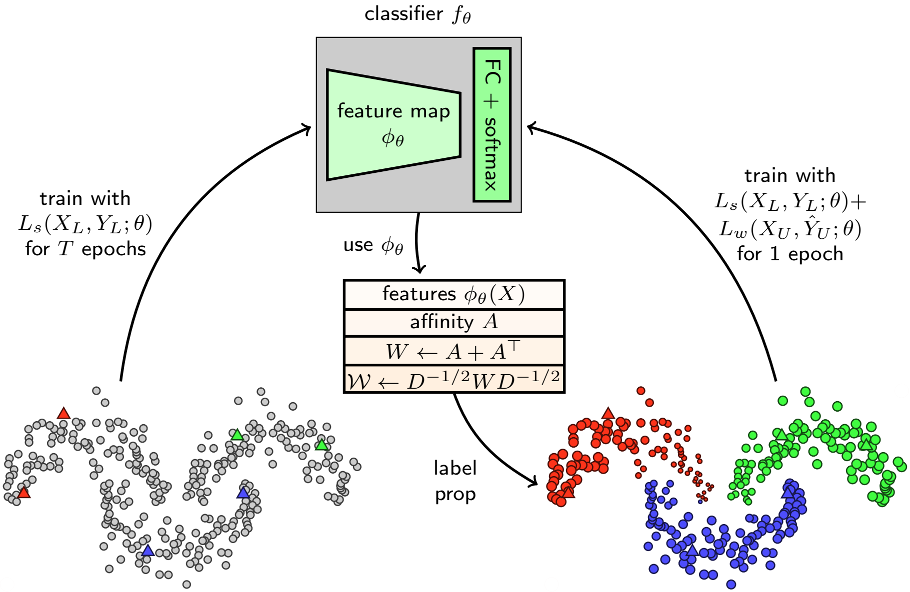
Iscen, Tolias, Avrithis and Chum. CVPR 2019. Label Propagation for Deep Semi-supervised Learning.

label propagation (inductive)



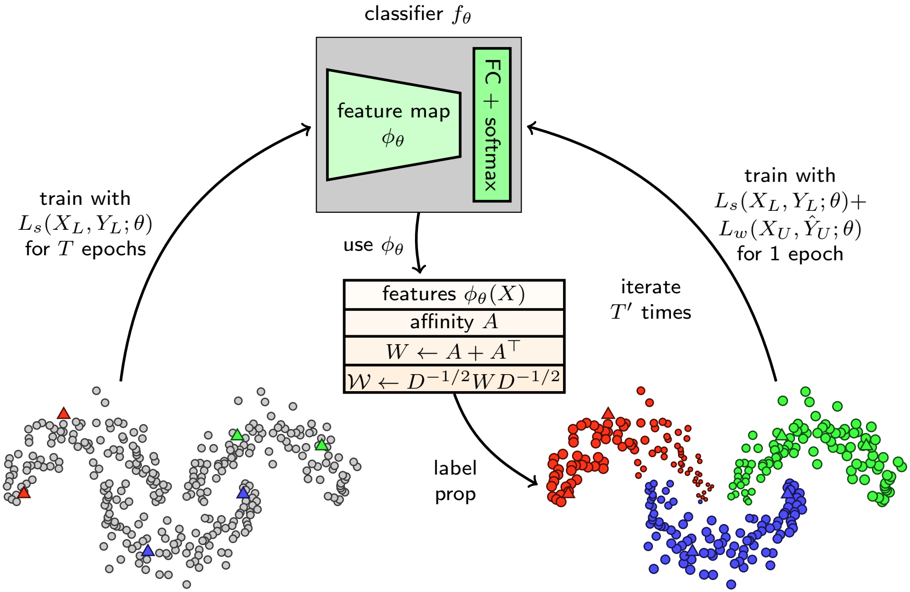
Iscen, Tolias, Avrithis and Chum. CVPR 2019. Label Propagation for Deep Semi-supervised Learning.

label propagation (inductive)



Iscen, Tolias, Avrithis and Chum. CVPR 2019. Label Propagation for Deep Semi-supervised Learning.

label propagation (inductive)



Iscen, Tolias, Avrithis and Chum. CVPR 2019. Label Propagation for Deep Semi-supervised Learning.

loss functions

- supervised loss

$$L_s(X_L, Y_L; \theta) := \sum_{i \in L} \ell_s\left(f_\theta(x_i), y_j\right)$$

where $\ell_s(\mathbf{s}, y) := -\log \mathbf{s}_y$ is cross-entropy loss

- weighted pseudo-label loss

$$L_w(X_U, \hat{Y}_U; \theta) := \sum_{i \in U} \omega_i \zeta_{\hat{y}_i} \ell_s\left(f_\theta(x_i), \hat{y}_i\right)$$

- certainty of the prediction for example x_i

$$\omega_i := 1 - \frac{H(\hat{\mathbf{z}}_i)}{\log c}$$

- class weight for class j, balancing class contribution

$$\zeta_j := (|L_j| + |U_j|)^{-1}$$

loss functions

- supervised loss

$$L_s(X_L, Y_L; \theta) := \sum_{i \in L} \ell_s\left(f_\theta(x_i), y_j\right)$$

where $\ell_s(\mathbf{s}, y) := -\log \mathbf{s}_y$ is cross-entropy loss

- weighted pseudo-label loss

$$L_w(X_U, \hat{Y}_U; \theta) := \sum_{i \in U} \omega_i \zeta_{\hat{y}_i} \ell_s\left(f_\theta(x_i), \hat{y}_i\right)$$

- certainty of the prediction for example x_i

$$\omega_i := 1 - \frac{H(\hat{\mathbf{z}}_i)}{\log c}$$

- class weight for class j, balancing class contribution

$$\zeta_j := (|L_j| + |U_j|)^{-1}$$

Iscen, Tolias, Avrithis and Chum. CVPR 2019. Label Propagation for Deep Semi-supervised Learning.

loss functions

- supervised loss

$$L_s(X_L, Y_L; \theta) := \sum_{i \in L} \ell_s\left(f_\theta(x_i), y_j\right)$$

where $\ell_s(\mathbf{s}, y) := -\log \mathbf{s}_y$ is cross-entropy loss

- weighted pseudo-label loss

$$L_w(X_U, \hat{Y}_U; \theta) := \sum_{i \in U} \omega_i \zeta_{\hat{y}_i} \ell_s\left(f_\theta(x_i), \hat{y}_i\right)$$

- certainty of the prediction for example x_i

$$\omega_i := 1 - \frac{H(\hat{\mathbf{z}}_i)}{\log c}$$

- class weight for class j, balancing class contribution

$$\zeta_j := (|L_j| + |U_j|)^{-1}$$

loss functions

- supervised loss

$$L_s(X_L, Y_L; \theta) := \sum_{i \in L} \ell_s\left(f_\theta(x_i), y_j\right)$$

where $\ell_s(\mathbf{s}, y) := -\log \mathbf{s}_y$ is cross-entropy loss

- weighted pseudo-label loss

$$L_w(X_U, \hat{Y}_U; \theta) := \sum_{i \in U} \omega_i \zeta_{\hat{y}_i} \ell_s\left(f_\theta(x_i), \hat{y}_i\right)$$

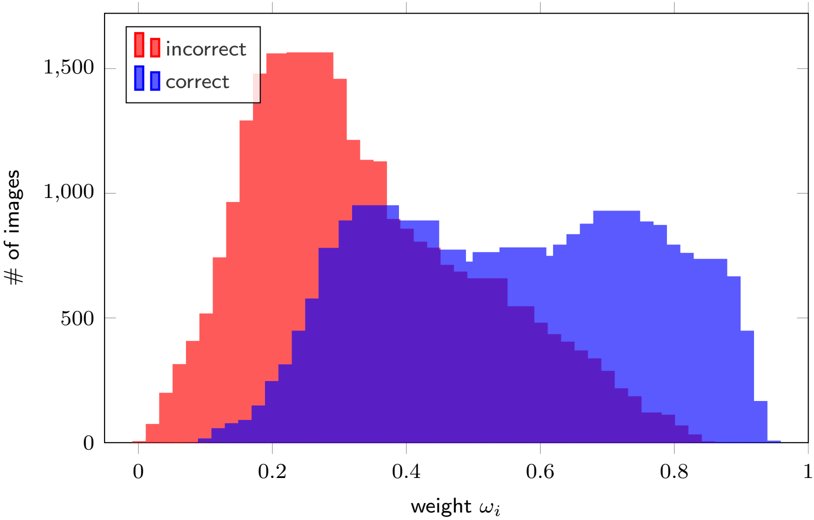
- certainty of the prediction for example x_i

$$\omega_i := 1 - \frac{H(\hat{\mathbf{z}}_i)}{\log c}$$

- class weight for class j, balancing class contribution

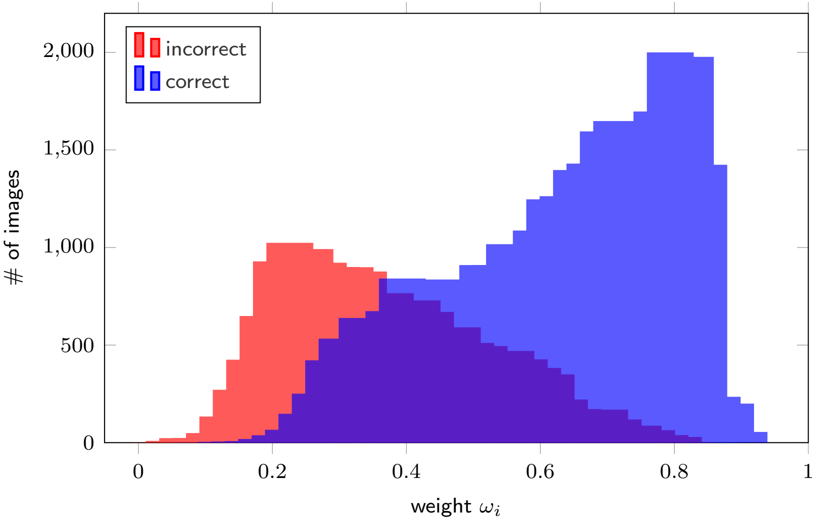
$$\zeta_j := \left(|L_j| + |U_j|\right)^{-1}$$

certainty weight distribution (epoch 00)



Iscen, Tolias, Avrithis and Chum. CVPR 2019. Label Propagation for Deep Semi-supervised Learning.

certainty weight distribution (epoch 90)



classification error on CIFAR10

Dataset	CIFAR-10			
Nb. labeled images	500	1000	2000	4000
Fully supervised	49.08 ± 0.83	40.03 ± 1.11	29.58 ± 0.93	21.63 ± 0.38
TDCNN [33]†	-	32.67 ± 1.93	22.99 ± 0.79	16.17 ± 0.37
Ours–(1)	35.17 ± 2.46	23.79 ± 1.31	16.64 ± 0.48	13.21 ± 0.61
Ours	32.40 ± 1.80	22.02 ± 0.88	15.66 ± 0.35	12.69 ± 0.29
VAT [23]†	-	-	-	11.36
Π model [20]†	-	-	-	12.36 ± 0.31
Temporal Ensemble [20]†	-	-	-	12.16 ± 0.24
MT [35]†	-	27.36 ± 1.30	15.73 ± 0.31	12.31 ± 0.28
MT [35]	27.45 ± 2.64	19.04 ± 0.51	14.35 ± 0.31	11.41 ± 0.25
MT + Ours	$\mathbf{24.02 \pm 2.44}$	$\mathbf{16.93 \pm 0.70}$	$\mathbf{13.22 \pm 0.29}$	$\mathbf{10.61 \pm 0.28}$

Iscen, Tolias, Avrithis and Chum. CVPR 2019. Label Propagation for Deep Semi-supervised Learning.

classification error on CIFAR100/miniImageNet

Dataset	CIFAR-100		Mini-ImageNet-*top1*	
Nb. labeled images	4000	10000	4000	10000
Fully supervised	55.43 ± 0.11	40.67 ± 0.49	74.78 ± 0.33	60.25 ± 0.29
Ours	46.20 ± 0.76	38.43 ± 1.88	$\mathbf{70.29 \pm 0.81}$	57.58 ± 1.47
MT [35]	45.36 ± 0.49	36.08 ± 0.51	72.51 ± 0.22	57.55 ± 1.11
MT + Ours	$\mathbf{43.73 \pm 0.20}$	$\mathbf{35.92 \pm 0.47}$	72.78 ± 0.15	$\mathbf{57.35 \pm 1.66}$

Iscen, Tolias, Avrithis and Chum. CVPR 2019. Label Propagation for Deep Semi-supervised Learning.

summary

- now that images are represented by a global descriptor or just a few regional descriptors, graph methods are more applicable than ever
- modeling the manifold explicitly allows unsupervised fine-tuning without labels, auxiliary systems (*e.g.* SIFT pipeline), or other information (*e.g.* temporal neighborhood in video)
- updating a graph while training and using it to provide "smooth" pseudo-labels boosts semi-supervised learning

summary

- now that images are represented by a global descriptor or just a few regional descriptors, graph methods are more applicable than ever
- modeling the manifold explicitly allows unsupervised fine-tuning without labels, auxiliary systems (*e.g.* SIFT pipeline), or other information (*e.g.* temporal neighborhood in video)
- updating a graph while training and using it to provide "smooth" pseudo-labels boosts semi-supervised learning

summary

- now that images are represented by a global descriptor or just a few regional descriptors, graph methods are more applicable than ever
- modeling the manifold explicitly allows unsupervised fine-tuning without labels, auxiliary systems (*e.g.* SIFT pipeline), or other information (*e.g.* temporal neighborhood in video)
- updating a graph while training and using it to provide "smooth" pseudo-labels boosts semi-supervised learning

thank you!