deep image retrieval and manifold learning

Yannis Avrithis

Inria Rennes-Bretagne Atlantique

Paris, March 2018

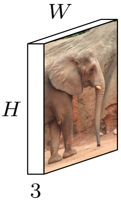
outline

feature pooling
manifold learning
fine-tuning
ranking on manifolds
fast spectral ranking
mining on manifolds

feature pooling

image ranking by CNN features

[Krizhevsky et al. 2012]

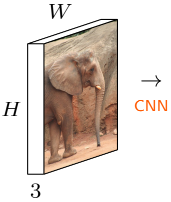


- 3-channel RGB input, 224×224
- AlexNet pre-trained on ImageNet for classification
- last fully connected layer (fc_6): global descriptor of dimension $k = 4096$

Krizhevsky, Sutskever, Hinton. NIPS 2012. Imagenet Classification with Deep Convolutional Neural Networks.

image ranking by CNN features

[Krizhevsky et al. 2012]

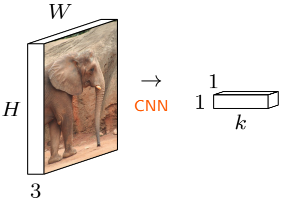


- 3-channel RGB input, 224×224
- AlexNet pre-trained on ImageNet for classification
- last fully connected layer (fc_6): global descriptor of dimension $k = 4096$

Krizhevsky, Sutskever, Hinton. NIPS 2012. Imagenet Classification with Deep Convolutional Neural Networks.

image ranking by CNN features

[Krizhevsky et al. 2012]



- 3-channel RGB input, 224×224
- AlexNet pre-trained on ImageNet for classification
- last fully connected layer (fc_6): global descriptor of dimension $k = 4096$

Krizhevsky, Sutskever, Hinton. NIPS 2012. Imagenet Classification with Deep Convolutional Neural Networks.

image ranking by CNN features

- query images
- nearest neighbors in ImageNet according to Euclidean distance

Krizhevsky, Sutskever, Hinton. NIPS 2012. Imagenet Classification with Deep Convolutional Neural Networks.

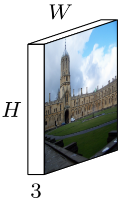
image ranking by CNN features

- query images
- nearest neighbors in ImageNet according to Euclidean distance

Krizhevsky, Sutskever, Hinton. NIPS 2012. Imagenet Classification with Deep Convolutional Neural Networks.

neural codes for image retrieval

[Babenko et al. 2014]

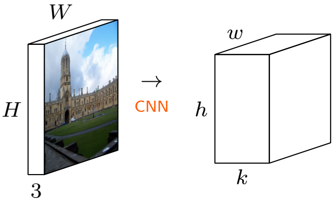


- 3-channel RGB input, 224×224
- AlexNet last pooling layer, global descriptor of dimension $w \times h \times k = 6 \times 6 \times 256 = 9216$
- alternatively: fully connected layers $\mathrm{fc}_6, \mathrm{fc}_7$, global descriptors of dimension $k' = 4096$ (best is fc_6)
- in each case: PCA-whitening, ℓ_2 normalization

Babenko, Slesarev, Chigorin, Lempitsky. ECCV 2014. Neural Codes for Image Retrieval.

neural codes for image retrieval

[Babenko et al. 2014]

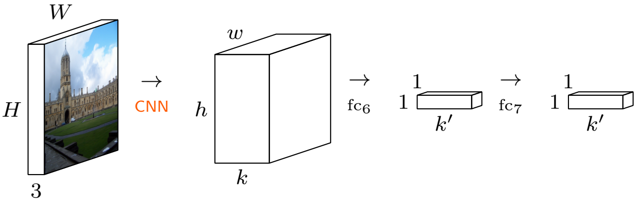


- 3-channel RGB input, 224×224
- AlexNet last pooling layer, global descriptor of dimension $w \times h \times k = 6 \times 6 \times 256 = 9216$
- alternatively: fully connected layers $\mathrm{fc}_6, \mathrm{fc}_7$, global descriptors of dimension $k' = 4096$ (best is fc_6)
- in each case: PCA-whitening, ℓ_2 normalization

Babenko, Slesarev, Chigorin, Lempitsky. ECCV 2014. Neural Codes for Image Retrieval.

neural codes for image retrieval

[Babenko et al. 2014]

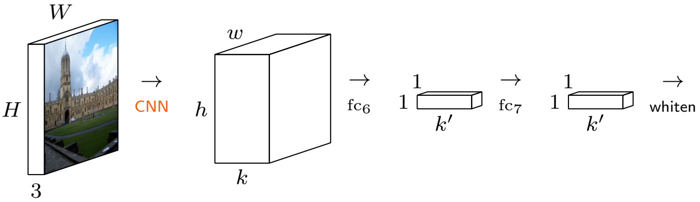


- 3-channel RGB input, 224×224
- AlexNet last pooling layer, global descriptor of dimension $w \times h \times k = 6 \times 6 \times 256 = 9216$
- alternatively: fully connected layers $\mathrm{fc}_6, \mathrm{fc}_7$, global descriptors of dimension $k' = 4096$ (best is fc_6)
- in each case: PCA-whitening, ℓ_2 normalization

Babenko, Slesarev, Chigorin, Lempitsky. ECCV 2014. Neural Codes for Image Retrieval.

neural codes for image retrieval

[Babenko et al. 2014]



- 3-channel RGB input, 224×224
- AlexNet last pooling layer, global descriptor of dimension $w \times h \times k = 6 \times 6 \times 256 = 9216$
- alternatively: fully connected layers $\mathrm{fc}_6, \mathrm{fc}_7$, global descriptors of dimension $k' = 4096$ (best is fc_6)
- in each case: PCA-whitening, ℓ_2 normalization

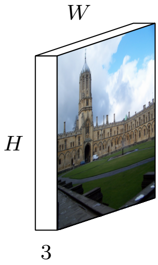
neural codes for image retrieval

- fine-tuning by softmax on 672 classes of 200k landmark photos
- outperforms VLAD and Fisher vectors on standard retrieval benchmarks, but still inferior to SIFT local descriptors

Babenko, Slesarev, Chigorin, Lempitsky. ECCV 2014. Neural Codes for Image Retrieval.

regional CNN features

[Razavian et al. 2015]

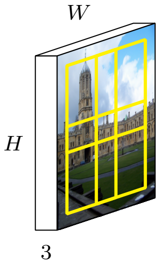


- 3-channel RGB input, largest square region extracted
- fixed multiscale overlapping regions, warped into $w \times h = 227 \times 227$
- each region yields a $w' \times h' \times k = 36 \times 36 \times 256$ dimensional feature at the last convolutional layer of AlexNet
- global spatial max-pooling
- ℓ_2-normalization, PCA-whitening of each descriptor

Razavian, Sullivan, Maki and Carlsson 2015. Visual Instance Retrieval with Deep Convolutional Networks.

regional CNN features

[Razavian et al. 2015]

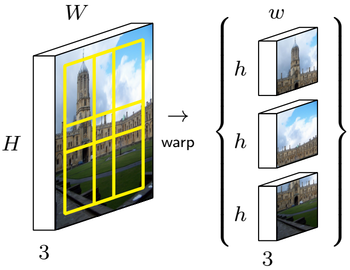


- 3-channel RGB input, largest square region extracted
- fixed multiscale overlapping regions, warped into $w \times h = 227 \times 227$
- each region yields a $w' \times h' \times k = 36 \times 36 \times 256$ dimensional feature at the last convolutional layer of AlexNet
- global spatial max-pooling
- ℓ_2-normalization, PCA-whitening of each descriptor

Razavian, Sullivan, Maki and Carlsson 2015. Visual Instance Retrieval with Deep Convolutional Networks.

regional CNN features

[Razavian et al. 2015]

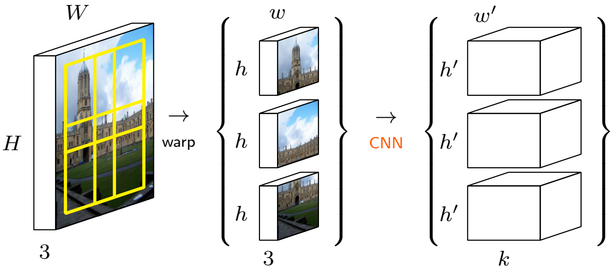


- 3-channel RGB input, largest square region extracted
- fixed multiscale overlapping regions, warped into $w \times h = 227 \times 227$
- each region yields a $w' \times h' \times k = 36 \times 36 \times 256$ dimensional feature at the last convolutional layer of AlexNet
- global spatial max-pooling
- ℓ_2-normalization, PCA-whitening of each descriptor

Razavian, Sullivan, Maki and Carlsson 2015. Visual Instance Retrieval with Deep Convolutional Networks.

regional CNN features

[Razavian et al. 2015]

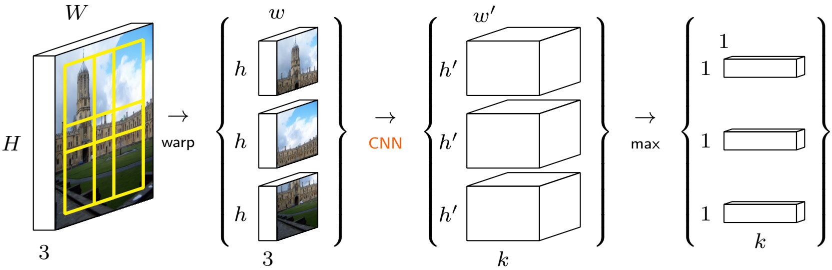


- 3-channel RGB input, largest square region extracted
- fixed multiscale overlapping regions, warped into $w \times h = 227 \times 227$
- each region yields a $w' \times h' \times k = 36 \times 36 \times 256$ dimensional feature at the last convolutional layer of AlexNet
- global spatial max-pooling
- ℓ_2-normalization, PCA-whitening of each descriptor

Razavian, Sullivan, Maki and Carlsson 2015. Visual Instance Retrieval with Deep Convolutional Networks.

regional CNN features

[Razavian et al. 2015]

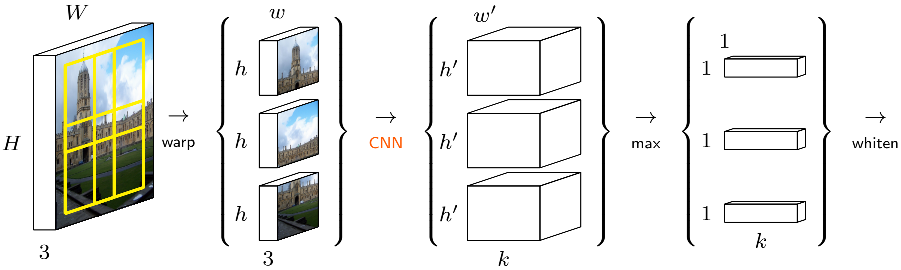


- 3-channel RGB input, largest square region extracted
- fixed multiscale overlapping regions, warped into $w \times h = 227 \times 227$
- each region yields a $w' \times h' \times k = 36 \times 36 \times 256$ dimensional feature at the last convolutional layer of AlexNet
- global spatial max-pooling
- ℓ_2-normalization, PCA-whitening of each descriptor

Razavian, Sullivan, Maki and Carlsson 2015. Visual Instance Retrieval with Deep Convolutional Networks.

regional CNN features

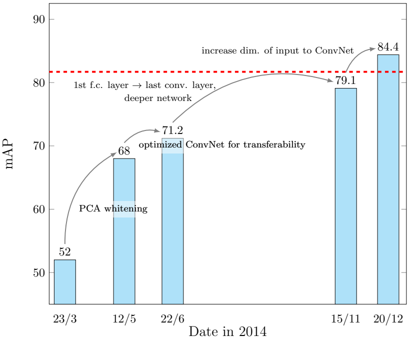
[Razavian et al. 2015]



- 3-channel RGB input, largest square region extracted
- fixed multiscale overlapping regions, warped into $w \times h = 227 \times 227$
- each region yields a $w' \times h' \times k = 36 \times 36 \times 256$ dimensional feature at the last convolutional layer of AlexNet
- global spatial max-pooling
- ℓ_2-normalization, PCA-whitening of each descriptor

Razavian, Sullivan, Maki and Carlsson 2015. Visual Instance Retrieval with Deep Convolutional Networks.

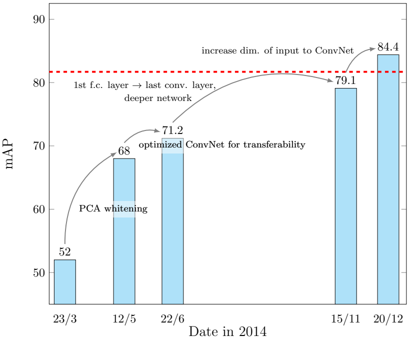
regional CNN features



- CNN visual representation jumps by more than 30% mAP to outperform standard SIFT pipeline in a few months
- however, this is based on multiple regional descriptors per image and exhaustive pairwise matching of all descriptors of query and all dataset images, which is not practical

Razavian, Sullivan, Maki and Carlsson 2015. Visual Instance Retrieval with Deep Convolutional Networks.

regional CNN features

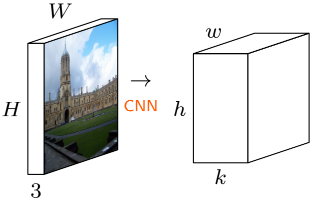


- CNN visual representation jumps by more than 30% mAP to outperform standard SIFT pipeline in a few months
- however, this is based on multiple regional descriptors per image and exhaustive pairwise matching of all descriptors of query and all dataset images, which is not practical

Razavian, Sullivan, Maki and Carlsson 2015. Visual Instance Retrieval with Deep Convolutional Networks.

regional max-pooling (R-MAC)

[Tolias et al. 2016]

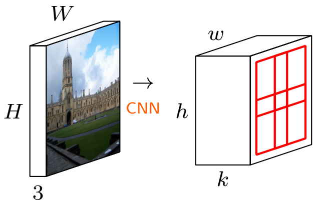


- VGG-16 last convolutional layer, $k = 512$
- fixed multiscale overlapping regions, spatial max-pooling
- ℓ_2-normalization, PCA-whitening, ℓ_2-normalization
- sum-pooling over all descriptors, ℓ_2-normalization

Tolias, Sicre and Jegou. ICLR 2016. Particular Object Retrieval with Integral Max-Pooling of CNN Activations.

regional max-pooling (R-MAC)

[Tolias et al. 2016]

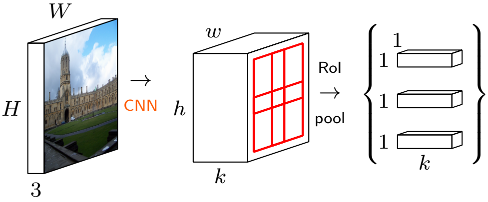


- VGG-16 last convolutional layer, $k = 512$
- fixed multiscale overlapping regions, spatial max-pooling
- ℓ_2-normalization, PCA-whitening, ℓ_2-normalization
- sum-pooling over all descriptors, ℓ_2-normalization

Tolias, Sicre and Jegou. ICLR 2016. Particular Object Retrieval with Integral Max-Pooling of CNN Activations.

regional max-pooling (R-MAC)

[Tolias et al. 2016]

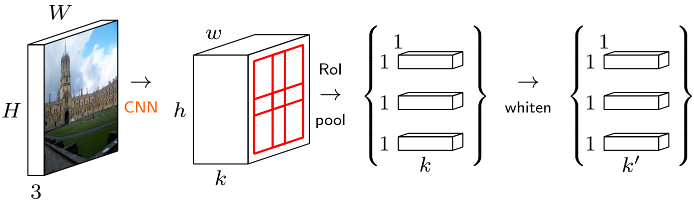


- VGG-16 last convolutional layer, $k = 512$
- fixed multiscale overlapping regions, spatial max-pooling
- ℓ_2-normalization, PCA-whitening, ℓ_2-normalization
- sum-pooling over all descriptors, ℓ_2-normalization

Tolias, Sicre and Jegou. ICLR 2016. Particular Object Retrieval with Integral Max-Pooling of CNN Activations.

regional max-pooling (R-MAC)

[Tolias et al. 2016]

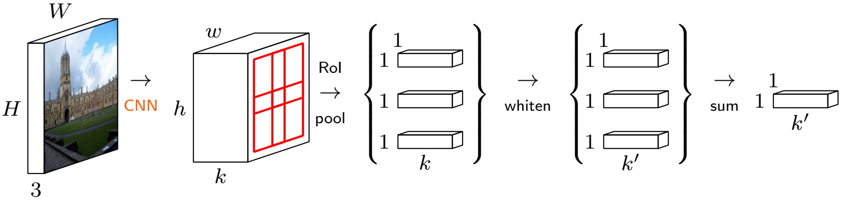


- VGG-16 last convolutional layer, $k = 512$
- fixed multiscale overlapping regions, spatial max-pooling
- ℓ_2-normalization, PCA-whitening, ℓ_2-normalization
- sum-pooling over all descriptors, ℓ_2-normalization

Tolias, Sicre and Jegou. ICLR 2016. Particular Object Retrieval with Integral Max-Pooling of CNN Activations.

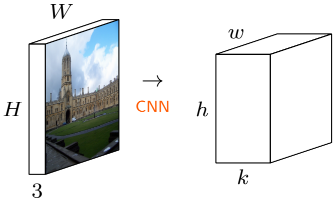
regional max-pooling (R-MAC)

[Tolias et al. 2016]



- VGG-16 last convolutional layer, $k = 512$
- fixed multiscale overlapping regions, spatial max-pooling
- ℓ_2-normalization, PCA-whitening, ℓ_2-normalization
- sum-pooling over all descriptors, ℓ_2-normalization

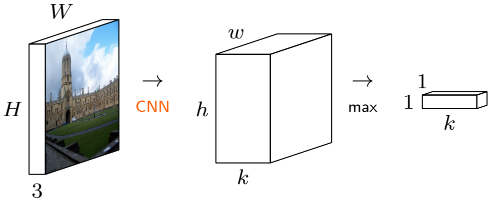
global max-pooling (MAC)



- VGG-16 last convolutional layer, $k = 512$
- global spatial max-pooling
- ℓ_2-normalization, PCA-whitening, ℓ_2-normalization
- MAC: maximum activation of convolutions

Tolias, Sicre and Jegou. ICLR 2016. Particular Object Retrieval with Integral Max-Pooling of CNN Activations.

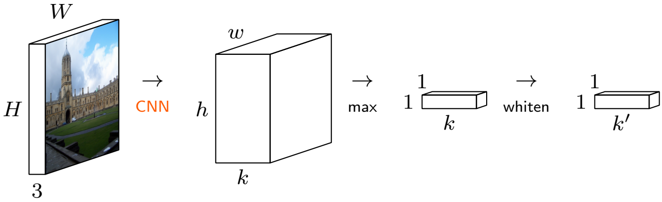
global max-pooling (MAC)



- VGG-16 last convolutional layer, $k = 512$
- global spatial max-pooling
- ℓ_2-normalization, PCA-whitening, ℓ_2-normalization
- MAC: maximum activation of convolutions

Tolias, Sicre and Jegou. ICLR 2016. Particular Object Retrieval with Integral Max-Pooling of CNN Activations.

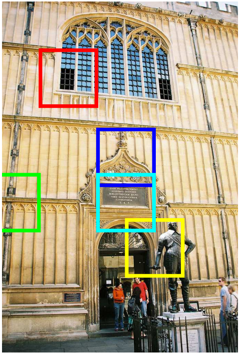
global max-pooling (MAC)

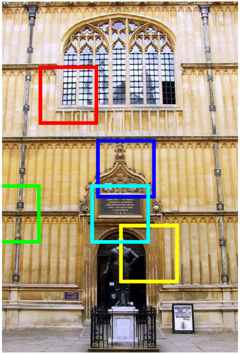


- VGG-16 last convolutional layer, $k = 512$
- global spatial max-pooling
- ℓ_2-normalization, PCA-whitening, ℓ_2-normalization
- MAC: maximum activation of convolutions

Tolias, Sicre and Jegou. ICLR 2016. Particular Object Retrieval with Integral Max-Pooling of CNN Activations.

global max-pooling: matching

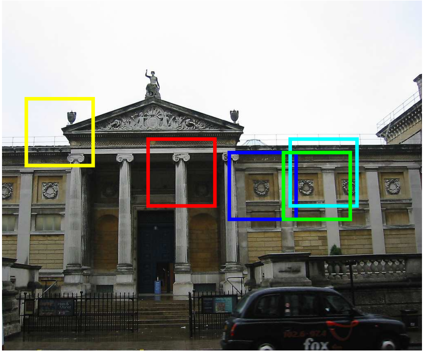


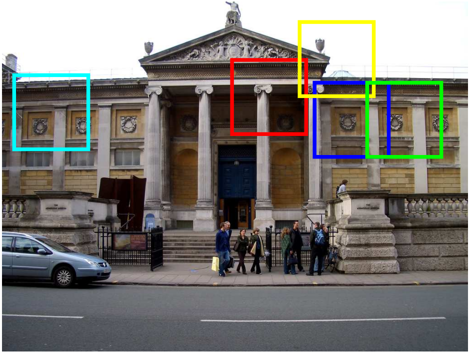


- receptive fields of 5 components of MAC vectors that contribute most to image similarity

Tolias, Sicre and Jégou. ICLR 2016. Particular Object Retrieval with Integral Max-Pooling of CNN Activations.

global max-pooling: matching





- receptive fields of 5 components of MAC vectors that contribute most to image similarity

Tolias, Sicre and Jégou. ICLR 2016. Particular Object Retrieval with Integral Max-Pooling of CNN Activations.

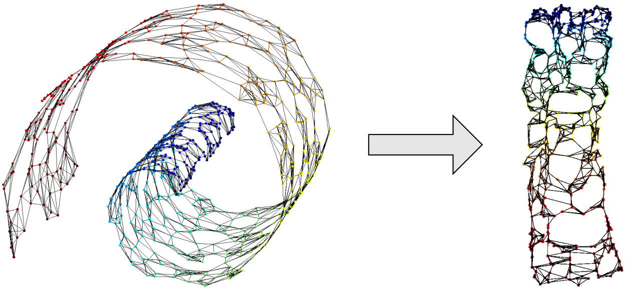
global max-pooling: matching

- receptive fields of 5 components of MAC vectors that contribute most to image similarity

Tolias, Sicre and Jégou. ICLR 2016. Particular Object Retrieval with Integral Max-Pooling of CNN Activations.

manifold learning

manifold learning



- *e.g.* Isomap: apply PCA to the geodesic (graph) distance matrix
- *e.g.* kernel PCA: apply PCA to the Gram matrix of a nonlinear kernel
- other topology-preserving methods are only focusing on distances to nearest neighbors
- many classic methods use eigenvalue decomposition and most do not learn and explicit mapping from the input to the embedding space

siamese architecture

[Chopra et al. 2005]

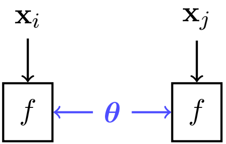
$\mathbf{x}_i$ $\mathbf{x}_j$

- an input sample is a pair $(\mathbf{x}_i, \mathbf{x}_j)$
- both $\mathbf{x}_i, \mathbf{x}_j$ go through the same function f with shared parameters θ
- loss ℓ_{ij} is measured on output pair $(\mathbf{y}_i, \mathbf{y}_j)$ and target t_{ij}

Chopra, Hadsell, Lecun, CVPR 2005. Learning a Similarity Metric Discriminatively, with Application to Face Verification.

siamese architecture

[Chopra et al. 2005]

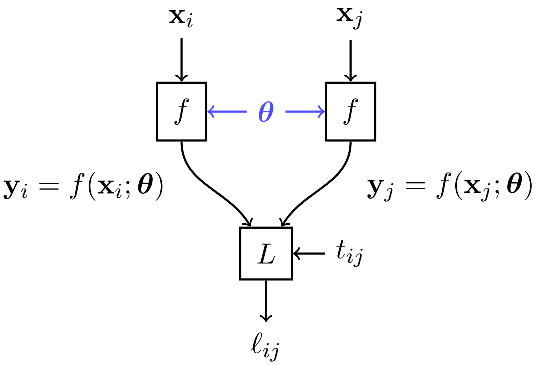


- an input sample is a pair $(\mathbf{x}_i, \mathbf{x}_j)$
- both $\mathbf{x}_i, \mathbf{x}_j$ go through the same function f with shared parameters $\boldsymbol{\theta}$
- loss ℓ_{ij} is measured on output pair $(\mathbf{y}_i, \mathbf{y}_j)$ and target t_{ij}

Chopra, Hadsell, Lecun, CVPR 2005. Learning a Similarity Metric Discriminatively, with Application to Face Verification.

siamese architecture

[Chopra et al. 2005]

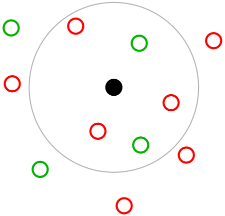


- an input sample is a pair $(\mathbf{x}_i, \mathbf{x}_j)$
- both $\mathbf{x}_i, \mathbf{x}_j$ go through the same function f with shared parameters $\boldsymbol{\theta}$
- loss ℓ_{ij} is measured on output pair $(\mathbf{y}_i, \mathbf{y}_j)$ and target t_{ij}

Chopra, Hadsell, Lecun, CVPR 2005. Learning a Similarity Metric Discriminatively, with Application to Face Verification.

contrastive loss

[Hadsell et al. 2006]



- input samples $\mathbf{x}_i$, output vectors $\mathbf{y}_i = f(\mathbf{x}_i; \boldsymbol{\theta})$
- target variables $t_{ij} = \mathbb{1}[\mathrm{sim}(\mathbf{x}_i, \mathbf{x}_j)]$
- contrastive loss is a function of distance $\|\mathbf{y}_i - \mathbf{y}_j\|$ only

$$\ell_{ij} = L((\mathbf{y}_i, \mathbf{y}_j), t_{ij}) = \ell(\|\mathbf{y}_i - \mathbf{y}_j\|, t_{ij})$$

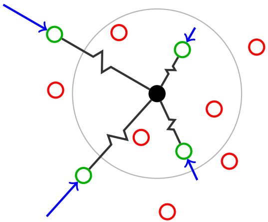
- similar samples are attracted

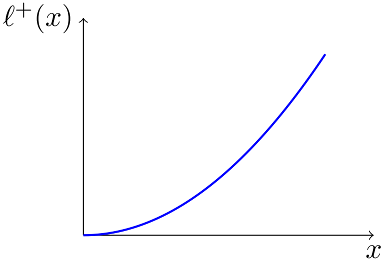
$$\ell(x, t) = t\ell^+(x) + (1-t)\ell^-(x) = tx^2 + (1-t)[m-x]_+^2$$

Hadsell, Chopra, Lecun. CVPR 2006. Dimensionality Reduction By Learning an Invariant Mapping.

contrastive loss

[Hadsell et al. 2006]





- input samples $\mathbf{x}_i$, output vectors $\mathbf{y}_i = f(\mathbf{x}_i; \boldsymbol{\theta})$
- target variables $t_{ij} = \mathbb{1}[\mathrm{sim}(\mathbf{x}_i, \mathbf{x}_j)]$
- contrastive loss is a function of distance $\|\mathbf{y}_i - \mathbf{y}_j\|$ only

$$\ell_{ij} = L((\mathbf{y}_i, \mathbf{y}_j), t_{ij}) = \ell(\|\mathbf{y}_i - \mathbf{y}_j\|, t_{ij})$$

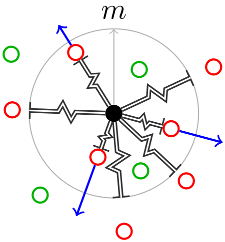
- similar samples are attracted

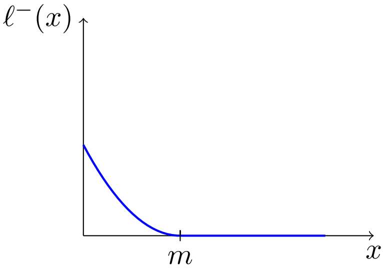
$$\ell(x,t) = \boxed{t\ell^+(x)} + (1-t)\ell^-(x) = \boxed{tx^2} + (1-t)[m-x]_+^2$$

Hadsell, Chopra, Lecun. CVPR 2006. Dimensionality Reduction By Learning an Invariant Mapping.

contrastive loss

[Hadsell et al. 2006]





- input samples $\mathbf{x}_i$, output vectors $\mathbf{y}_i = f(\mathbf{x}_i; \boldsymbol{\theta})$
- target variables $t_{ij} = \mathbb{1}[\mathrm{sim}(\mathbf{x}_i, \mathbf{x}_j)]$
- contrastive loss is a function of distance $\|\mathbf{y}_i - \mathbf{y}_j\|$ only

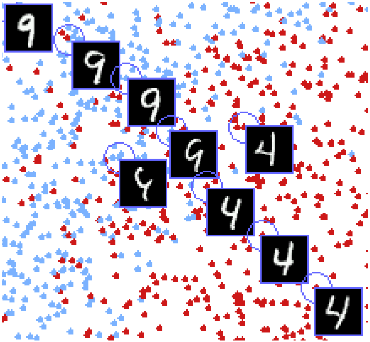
$$\ell_{ij} = L((\mathbf{y}_i, \mathbf{y}_j), t_{ij}) = \ell(\|\mathbf{y}_i - \mathbf{y}_j\|, t_{ij})$$

- dissimilar samples are repelled if closer than margin m

$$\ell(x,t) = t\ell^+(x) + (1-t)\boxed{\ell^-(x)} = tx^2 + (1-t)\boxed{[m-x]_+^2}$$

Hadsell, Chopra, Lecun. CVPR 2006. Dimensionality Reduction By Learning an Invariant Mapping.

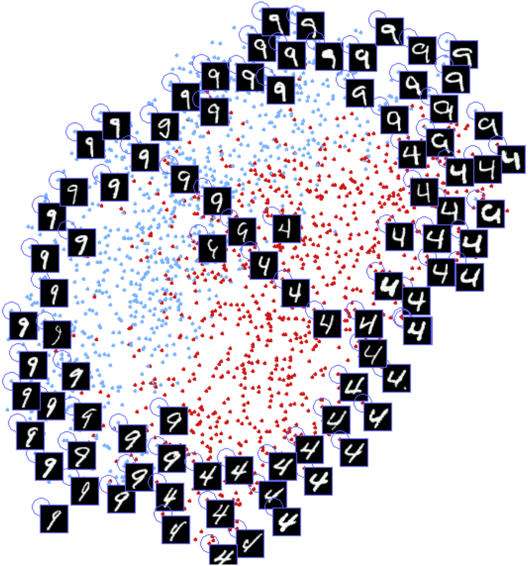
manifold learning: MNIST



- 3k samples of each of digits $4, 9$
- each sample similar to its 5 Euclidean nearest neighbors, and dissimilar to all other points
- 30k similar pairs, 18M dissimilar pairs

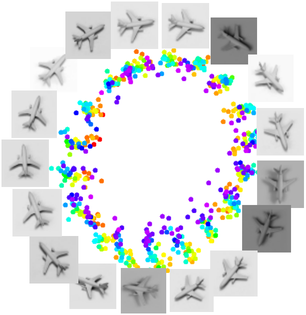
Hadsell, Chopra, Lecun. CVPR 2006. Dimensionality Reduction By Learning an Invariant Mapping.

manifold learning: MNIST



Hadsell, Chopra, Lecun. CVPR 2006. Dimensionality Reduction By Learning an Invariant Mapping.

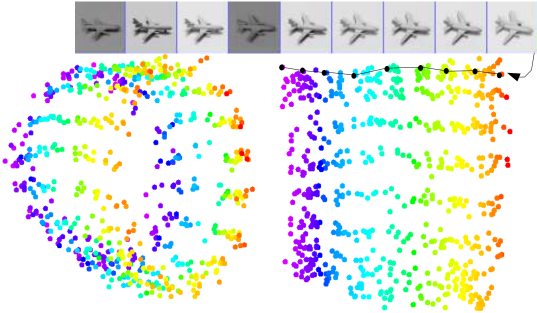
manifold learning: NORB



- 972 images of airplane class: 18 azimuths (every 20°), 9 elevations (in $[30^\circ, 70^\circ]$, every 5°), 6 lighting conditions
- samples similar if taken from contiguous azimuth or elevation, regardless of lighting
- 11k similar pairs, 206M dissimilar pairs
- cylindrer in 3d: azimuth on circumference, elevation on height

Hadsell, Chopra, Lecun. CVPR 2006. Dimensionality Reduction By Learning an Invariant Mapping.

manifold learning: NORB



- 972 images of airplane class: 18 azimuths (every $20°$), 9 elevations (in $[30°, 70°]$, every $5°$), 6 lighting conditions
- samples similar if taken from contiguous azimuth or elevation, regardless of lighting
- 11k similar pairs, 206M dissimilar pairs
- cylindrer in 3d: azimuth on circumference, elevation on height

Hadsell, Chopra, Lecun. CVPR 2006. Dimensionality Reduction By Learning an Invariant Mapping.

triplet architecture

[Wang et al. 2014]

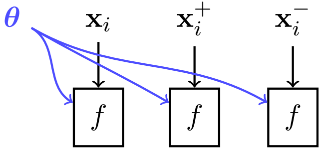
$\mathbf{x}_i \qquad \mathbf{x}_i^+ \qquad \mathbf{x}_i^-$

- an input sample is a triplet $(\mathbf{x}_i, \mathbf{x}_i^+, \mathbf{x}_i^-)$
- $\mathbf{x}_i, \mathbf{x}_i^+, \mathbf{x}_i^-$ go through the same function f with shared parameters $\boldsymbol{\theta}$
- loss ℓ_i measured on output triplet $(\mathbf{y}_i, \mathbf{y}_i^+, \mathbf{y}_i^-)$

Wang, Song, Leung, Rosenberg, Wang, Philbin, Chen, Wu. CVPR 2014. Learning Fine-Grained Image Similarity with Deep Ranking.

triplet architecture

[Wang et al. 2014]

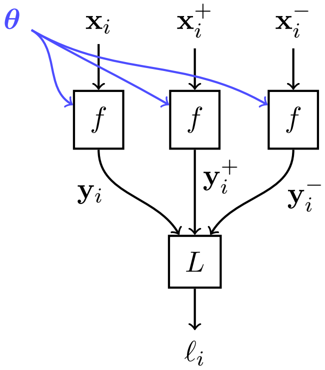


- an input sample is a triplet $(\mathbf{x}_i, \mathbf{x}_i^+, \mathbf{x}_i^-)$
- $\mathbf{x}_i, \mathbf{x}_i^+, \mathbf{x}_i^-$ go through the same function f with shared parameters $\boldsymbol{\theta}$
- loss ℓ_i measured on output triplet $(\mathbf{y}_i, \mathbf{y}_i^+, \mathbf{y}_i^-)$

Wang, Song, Leung, Rosenberg, Wang, Philbin, Chen, Wu. CVPR 2014. Learning Fine-Grained Image Similarity with Deep Ranking.

triplet architecture

[Wang et al. 2014]



- an input sample is a triplet $(\mathbf{x}_i, \mathbf{x}_i^+, \mathbf{x}_i^-)$
- $\mathbf{x}_i, \mathbf{x}_i^+, \mathbf{x}_i^-$ go through the same function f with shared parameters $\boldsymbol{\theta}$
- loss ℓ_i measured on output triplet $(\mathbf{y}_i, \mathbf{y}_i^+, \mathbf{y}_i^-)$

Wang, Song, Leung, Rosenberg, Wang, Philbin, Chen, Wu. CVPR 2014. Learning Fine-Grained Image Similarity with Deep Ranking.

triplet loss

- input "anchor" $\mathbf{x}_i$, output vector $\mathbf{y}_i = f(\mathbf{x}_i; \boldsymbol{\theta})$
- positive $\mathbf{y}_i^+ = f(\mathbf{x}_i^+; \boldsymbol{\theta})$, negative $\mathbf{y}_i^- = f(\mathbf{x}_i^-; \boldsymbol{\theta})$
- triplet loss is a function of distances $\|\mathbf{y}_i - \mathbf{y}_i^+\|, \|\mathbf{y}_i - \mathbf{y}_i^-\|$ only

$$\ell_i = L(\mathbf{y}_i, \mathbf{y}_i^+, \mathbf{y}_i^-) = \ell(\|\mathbf{y}_i - \mathbf{y}_i^+\|, \|\mathbf{y}_i - \mathbf{y}_i^-\|)$$

$$\ell(x^+, x^-) = \left[m + (x^+)^2 - (x^-)^2\right]_+$$

so distance $\|\mathbf{y}_i - \mathbf{y}_i^+\|$ should be less than $\|\mathbf{y}_i - \mathbf{y}_i^-\|$ by margin m

- by taking two pairs $(\mathbf{x}_i, \mathbf{x}_i^+)$ and $(\mathbf{x}_i, \mathbf{x}_i^-)$ at a time with targets $1, 0$ respectively, the contrastive loss can be written similarly

$$\ell(x^+, x^-) = (x^+)^2 + \left[m - x^-\right]_+^2$$

so distance $\|\mathbf{y}_i - \mathbf{y}_i^+\|$ should small and $\|\mathbf{y}_i - \mathbf{y}_i^-\|$ larger than m

Wang, Song, Leung, Rosenberg, Wang, Philbin, Chen, Wu. CVPR 2014. Learning Fine-Grained Image Similarity with Deep Ranking.

triplet loss

- input "anchor" $\mathbf{x}_i$, output vector $\mathbf{y}_i = f(\mathbf{x}_i; \boldsymbol{\theta})$
- positive $\mathbf{y}_i^+ = f(\mathbf{x}_i^+; \boldsymbol{\theta})$, negative $\mathbf{y}_i^- = f(\mathbf{x}_i^-; \boldsymbol{\theta})$
- triplet loss is a function of distances $\|\mathbf{y}_i - \mathbf{y}_i^+\|, \|\mathbf{y}_i - \mathbf{y}_i^-\|$ only

$$\ell_i = L(\mathbf{y}_i, \mathbf{y}_i^+, \mathbf{y}_i^-) = \ell(\|\mathbf{y}_i - \mathbf{y}_i^+\|, \|\mathbf{y}_i - \mathbf{y}_i^-\|)$$

$$\ell(x^+, x^-) = \left[m + (x^+)^2 - (x^-)^2\right]_+$$

so distance $\|\mathbf{y}_i - \mathbf{y}_i^+\|$ should be less than $\|\mathbf{y}_i - \mathbf{y}_i^-\|$ by margin m

- by taking two pairs $(\mathbf{x}_i, \mathbf{x}_i^+)$ and $(\mathbf{x}_i, \mathbf{x}_i^-)$ at a time with targets $1, 0$ respectively, the contrastive loss can be written similarly

$$\ell(x^+, x^-) = (x^+)^2 + \left[m - x^-\right]_+^2$$

so distance $\|\mathbf{y}_i - \mathbf{y}_i^+\|$ should small and $\|\mathbf{y}_i - \mathbf{y}_i^-\|$ larger than m

Wang, Song, Leung, Rosenberg, Wang, Philbin, Chen, Wu. CVPR 2014. Learning Fine-Grained Image Similarity with Deep Ranking.

unsupervised learning by context prediction

[Doersch et al. 2015]

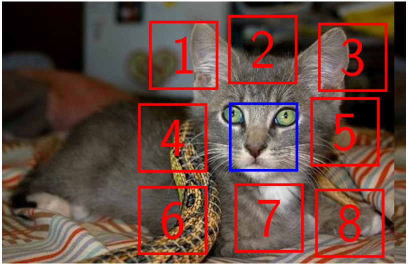
- sample random pairs of patches in one of eight spatial configurations
- patches are randomly jittered and do not overlap
- like solving a puzzle, learn to predict the relative position

$$f\left(\quad , \quad \right) = 3$$

Doersch, Gupta, Efros. ICCV 2015. Unsupervised Visual Representation Learning By Context Prediction.

unsupervised learning by context prediction

[Doersch et al. 2015]



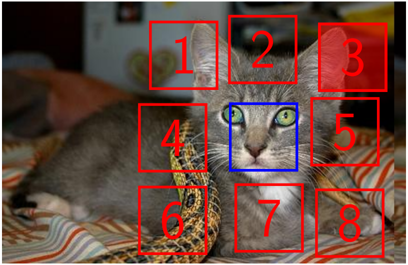
- sample random pairs of patches in one of eight spatial configurations
- patches are randomly jittered and do not overlap
- like solving a puzzle, learn to predict the relative position

$$f\left(\quad , \quad \right) = 3$$

Doersch, Gupta, Efros. ICCV 2015. Unsupervised Visual Representation Learning By Context Prediction.

unsupervised learning by context prediction

[Doersch et al. 2015]

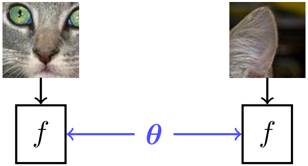


- sample random pairs of patches in one of eight spatial configurations
- patches are randomly jittered and do not overlap
- like solving a puzzle, learn to predict the relative position

$f\Bigg($  ,  $\Bigg) = 3$

Doersch, Gupta, Efros. ICCV 2015. Unsupervised Visual Representation Learning By Context Prediction.

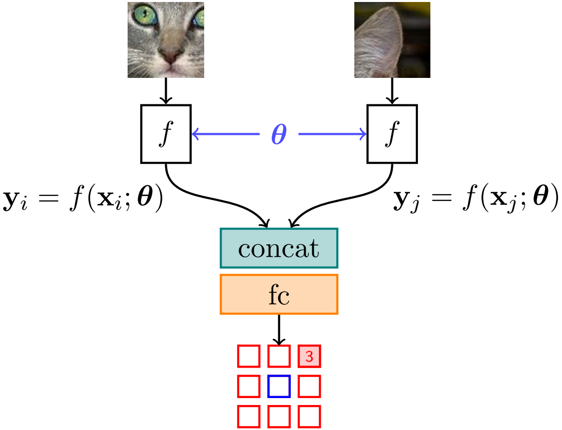
context prediction: architecture



- network f learned by siamese architecture
- representations are concatenated and followed by softmax classifier, where each spatial configuration is a class

Doersch, Gupta, Efros. ICCV 2015. Unsupervised Visual Representation Learning By Context Prediction.

context prediction: architecture



- network f learned by siamese architecture
- representations are concatenated and followed by softmax classifier, where each spatial configuration is a class

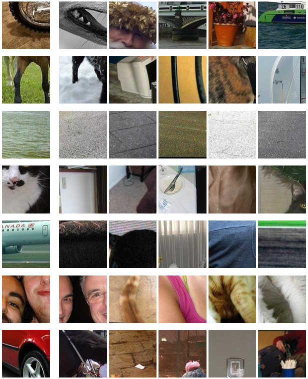
Doersch, Gupta, Efros. ICCV 2015. Unsupervised Visual Representation Learning By Context Prediction.

context prediction: examples

- input image
- nearest neighbors with randomly initialized network
- trained by supervised classification on ImageNet
- unsupervised training from scratch on the context prediction task

Doersch, Gupta, Efros. ICCV 2015. Unsupervised Visual Representation Learning By Context Prediction.

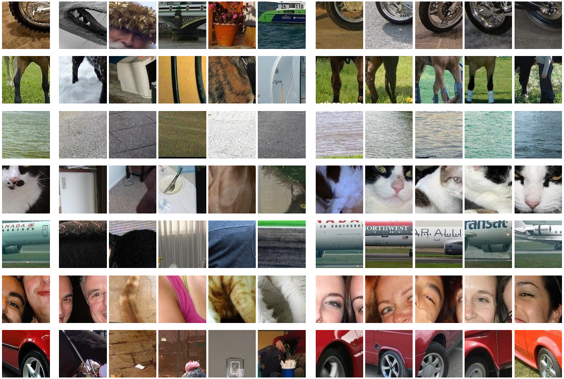
context prediction: examples



- input image
- nearest neighbors with randomly initialized network
- trained by supervised classification on ImageNet
- unsupervised training from scratch on the context prediction task

Doersch, Gupta, Efros. ICCV 2015. Unsupervised Visual Representation Learning By Context Prediction.

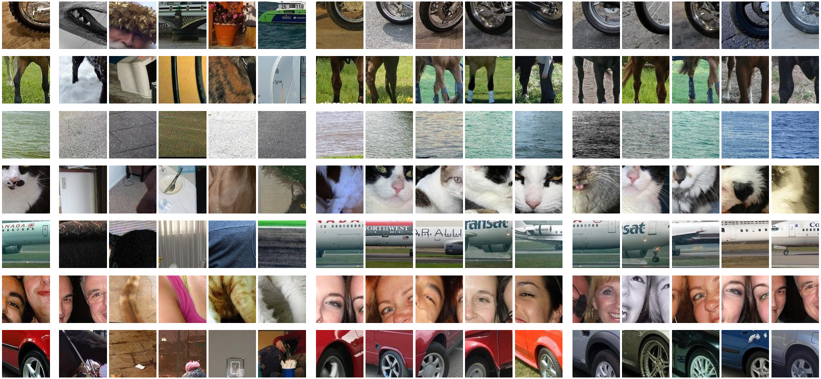
context prediction: examples



- input image
- nearest neighbors with randomly initialized network
- trained by supervised classification on ImageNet
- unsupervised training from scratch on the context prediction task

Doersch, Gupta, Efros. ICCV 2015. Unsupervised Visual Representation Learning By Context Prediction.

context prediction: examples

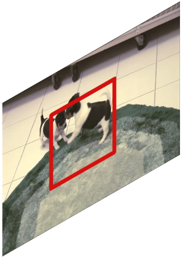


- input image
- nearest neighbors with randomly initialized network
- trained by supervised classification on ImageNet
- unsupervised training from scratch on the context prediction task

Doersch, Gupta, Efros. ICCV 2015. Unsupervised Visual Representation Learning By Context Prediction.

unsupervised learning on video: tracking

[Wang et al. 2015]

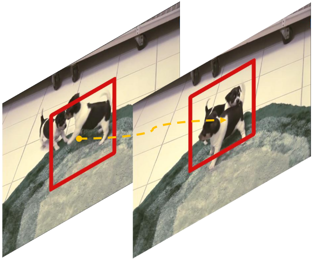


- estimate motion and find the region that contains most motion
- track this region for a number of frames
- generate a pair of matching patches on the first and last frames

Wang and Gupta. ICCV 2015. Unsupervised Learning of Visual Representations Using Videos.

unsupervised learning on video: tracking

[Wang et al. 2015]

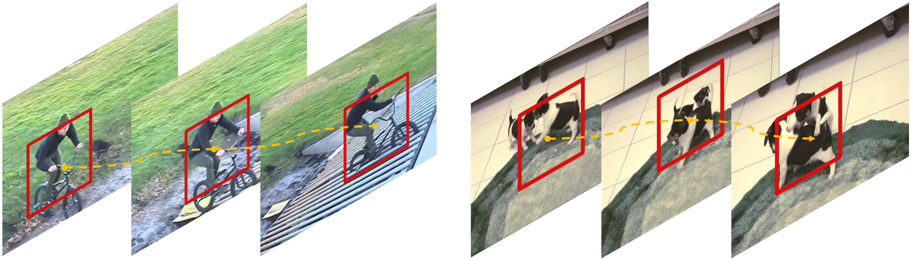


- estimate motion and find the region that contains most motion
- track this region for a number of frames
- generate a pair of matching patches on the first and last frames

Wang and Gupta. ICCV 2015. Unsupervised Learning of Visual Representations Using Videos.

unsupervised learning on video: tracking

[Wang et al. 2015]



- estimate motion and find the region that contains most motion
- track this region for a number of frames
- generate a pair of matching patches on the first and last frames

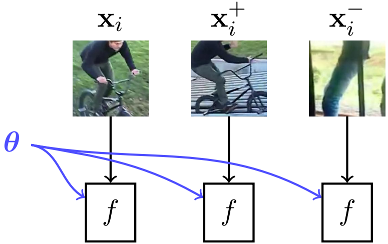
Wang and Gupta. ICCV 2015. Unsupervised Learning of Visual Representations Using Videos.

unsupervised learning on video: architecture

- input query $\mathbf{x}_i$ (first frame), tracked $\mathbf{x}_i^+$ (last frame), random $\mathbf{x}_i^-$
- $\mathbf{x}_i, \mathbf{x}_i^+, \mathbf{x}_i^-$ go through the same function f with shared parameters $\boldsymbol{\theta}$
- triplet loss ℓ_i measured on output triplet $(\mathbf{y}_i, \mathbf{y}_i^+, \mathbf{y}_i^-)$

Wang and Gupta. ICCV 2015. Unsupervised Learning of Visual Representations Using Videos.

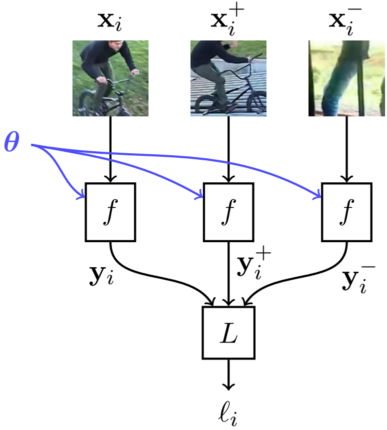
unsupervised learning on video: architecture



- input query $\mathbf{x}_i$ (first frame), tracked $\mathbf{x}_i^+$ (last frame), random $\mathbf{x}_i^-$
- $\mathbf{x}_i, \mathbf{x}_i^+, \mathbf{x}_i^-$ go through the same function f with shared parameters $\boldsymbol{\theta}$
- triplet loss ℓ_i measured on output triplet $(\mathbf{y}_i, \mathbf{y}_i^+, \mathbf{y}_i^-)$

Wang and Gupta. ICCV 2015. Unsupervised Learning of Visual Representations Using Videos.

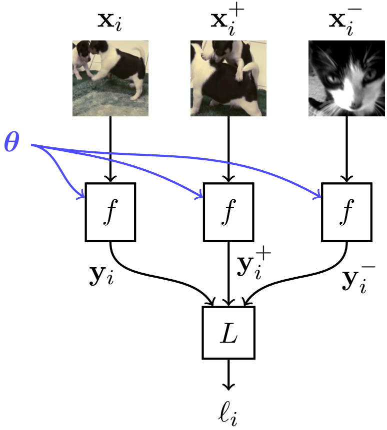
unsupervised learning on video: architecture



- input query $\mathbf{x}_i$ (first frame), tracked $\mathbf{x}_i^+$ (last frame), random $\mathbf{x}_i^-$
- $\mathbf{x}_i, \mathbf{x}_i^+, \mathbf{x}_i^-$ go through the same function f with shared parameters $\boldsymbol{\theta}$
- triplet loss ℓ_i measured on output triplet $(\mathbf{y}_i, \mathbf{y}_i^+, \mathbf{y}_i^-)$

Wang and Gupta. ICCV 2015. Unsupervised Learning of Visual Representations Using Videos.

unsupervised learning on video: architecture



- input query $\mathbf{x}_i$ (first frame), tracked $\mathbf{x}_i^+$ (last frame), random $\mathbf{x}_i^-$
- $\mathbf{x}_i, \mathbf{x}_i^+, \mathbf{x}_i^-$ go through the same function f with shared parameters $\boldsymbol{\theta}$
- triplet loss ℓ_i measured on output triplet $(\mathbf{y}_i, \mathbf{y}_i^+, \mathbf{y}_i^-)$

Wang and Gupta. ICCV 2015. Unsupervised Learning of Visual Representations Using Videos.

unsupervised learning on video: objective

$\Big\| f\Big($  $\Big) - f\Big($  $\Big)\Big\|^2 < \Big\| f\Big($  $\Big) - f\Big($  $\Big)\Big\|^2 - m$

$\Big\| f\Big($  $\Big) - f\Big($  $\Big)\Big\|^2 < \Big\| f\Big($  $\Big) - f\Big($  $\Big)\Big\|^2 - m$

- so, the objective is that squared distance $\|\mathbf{y}_i - \mathbf{y}_i^+\|^2$ is less than $\|\mathbf{y}_i - \mathbf{y}_i^-\|^2$ by margin m

Wang and Gupta. ICCV 2015. Unsupervised Learning of Visual Representations Using Videos.

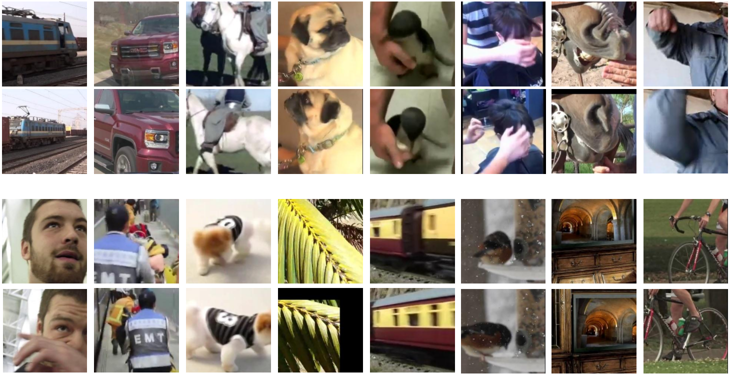
unsupervised learning on video: more examples

$\mathbf{x}_i$

$\mathbf{x}_i^+$

$\mathbf{x}_i$

$\mathbf{x}_i^+$

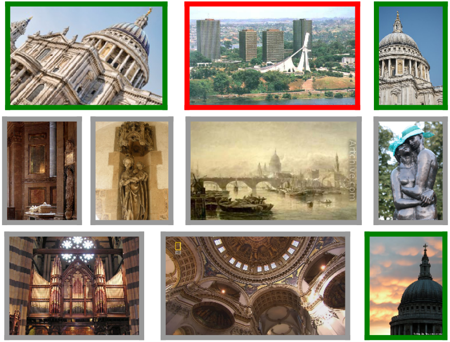


- input query $\mathbf{x}_i$ (first frame), tracked $\mathbf{x}_i^+$ (last frame)

fine-tuning

deep image retrieval: dataset cleaning

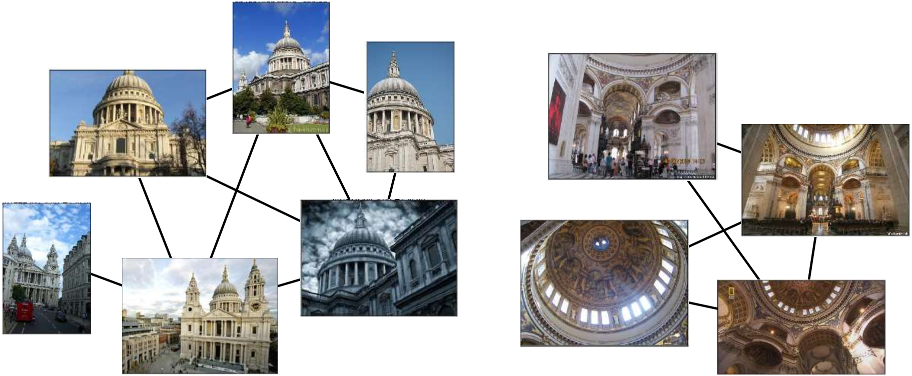
[Gordo et al. 2016]



- start from landmark dataset (192k images) and clean it (49k images)
- use it to fine-tune a network pre-trained on ImageNet for classification
- prototypical, non-prototypical and incorrect images per class
- only prototypical are kept to reduce intra-class variability

Gordo, Almazan, Revaud, Larlus. ECCV 2016. Deep Image Retrieval: Learning Global Representations for Image Search.

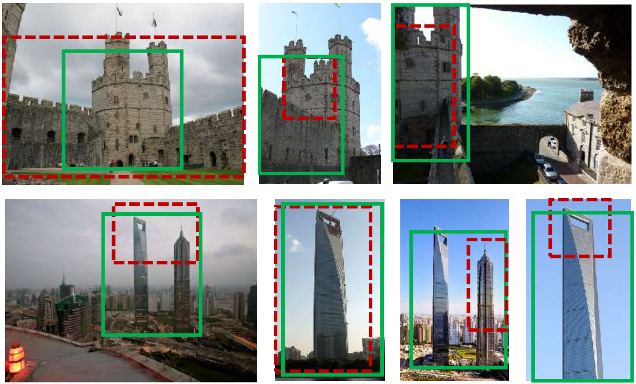
deep image retrieval: prototypical views



- pairwise match images per class by SIFT descriptors and fast spatial matching
- connect images into a graph and compute the connected components
- keep only the largest component

Gordo, Almazan, Revaud, Larlus. ECCV 2016. Deep Image Retrieval: Learning Global Representations for Image Search.

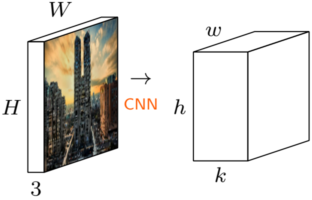
deep image retrieval: bounding boxes



- automatically find object bounding boxes
  - initialize with inlier features per image
  - update such that boxes are consistent over all matching pairs
- use bounding boxes to train a region proposal network

Gordo, Almazan, Revaud, Larlus. ECCV 2016. Deep Image Retrieval: Learning Global Representations for Image Search.

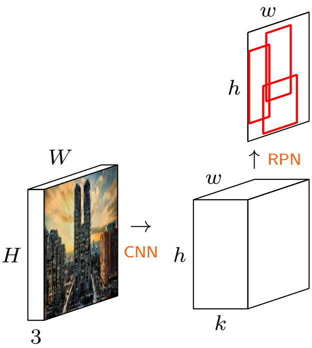
deep image retrieval: network, regions, pooling



- VGG-16 or ResNet-101 feature maps
- proposals detected on feature maps by RPN and max-pooled
- ℓ_2-normalization, PCA-whitening (FC layer), ℓ_2-normalization
- sum-pooling, ℓ_2-normalization (as in R-MAC)

Gordo, Almazan, Revaud, Larlus. ECCV 2016. Deep Image Retrieval: Learning Global Representations for Image Search.

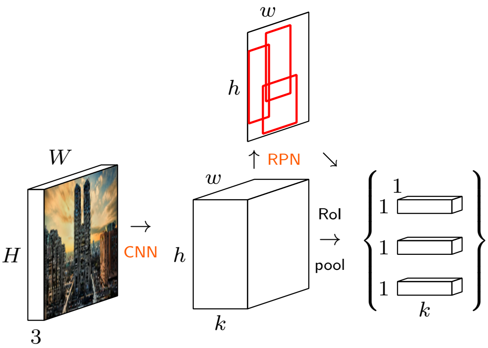
deep image retrieval: network, regions, pooling



- VGG-16 or ResNet-101 feature maps
- proposals detected on feature maps by RPN and max-pooled
- ℓ_2-normalization, PCA-whitening (FC layer), ℓ_2-normalization
- sum-pooling, ℓ_2-normalization (as in R-MAC)

Gordo, Almazan, Revaud, Larlus. ECCV 2016. Deep Image Retrieval: Learning Global Representations for Image Search.

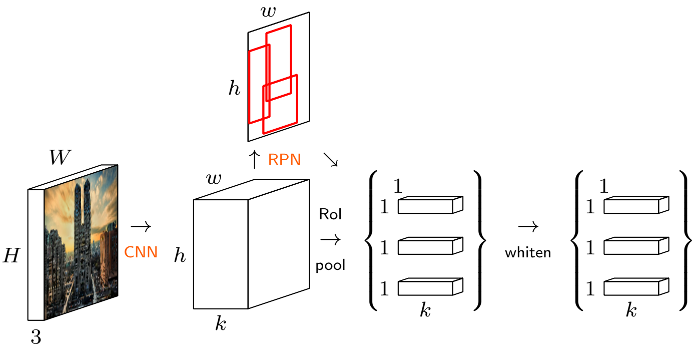
deep image retrieval: network, regions, pooling



- VGG-16 or ResNet-101 feature maps
- proposals detected on feature maps by RPN and max-pooled
- ℓ_2-normalization, PCA-whitening (FC layer), ℓ_2-normalization
- sum-pooling, ℓ_2-normalization (as in R-MAC)

Gordo, Almazan, Revaud, Larlus. ECCV 2016. Deep Image Retrieval: Learning Global Representations for Image Search.

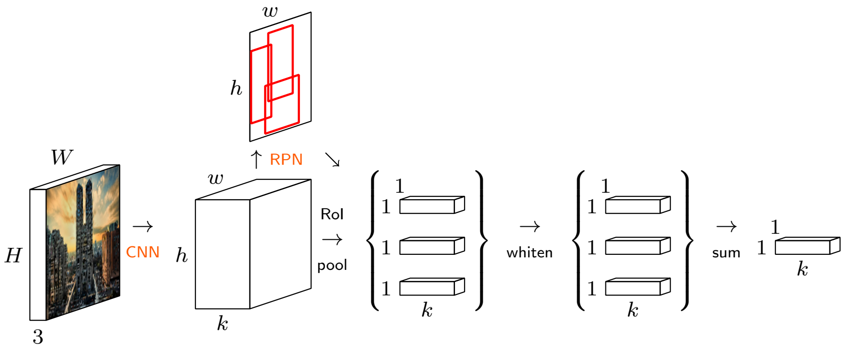
deep image retrieval: network, regions, pooling



- VGG-16 or ResNet-101 feature maps
- proposals detected on feature maps by RPN and max-pooled
- ℓ_2-normalization, PCA-whitening (FC layer), ℓ_2-normalization
- sum-pooling, ℓ_2-normalization (as in R-MAC)

Gordo, Almazan, Revaud, Larlus. ECCV 2016. Deep Image Retrieval: Learning Global Representations for Image Search.

deep image retrieval: network, regions, pooling



- VGG-16 or ResNet-101 feature maps
- proposals detected on feature maps by RPN and max-pooled
- ℓ_2-normalization, PCA-whitening (FC layer), ℓ_2-normalization
- sum-pooling, ℓ_2-normalization (as in R-MAC)

Gordo, Almazan, Revaud, Larlus. ECCV 2016. Deep Image Retrieval: Learning Global Representations for Image Search.

deep image retrieval: architecture

$\mathbf{x}_i$

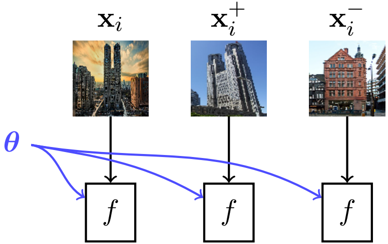
$\mathbf{x}_i^+$

$\mathbf{x}_i^-$

- query $\mathbf{x}_i$, relevant $\mathbf{x}_i^+$ (same building), irrelevant $\mathbf{x}_i^-$ (other building)
- $\mathbf{x}_i, \mathbf{x}_i^+, \mathbf{x}_i^-$ go through function f including features, RPN, pooling
- triplet loss ℓ_i measured on output $(\mathbf{y}_i, \mathbf{y}_i^+, \mathbf{y}_i^-)$

Gordo, Almazan, Revaud, Larlus. ECCV 2016. Deep Image Retrieval: Learning Global Representations for Image Search.

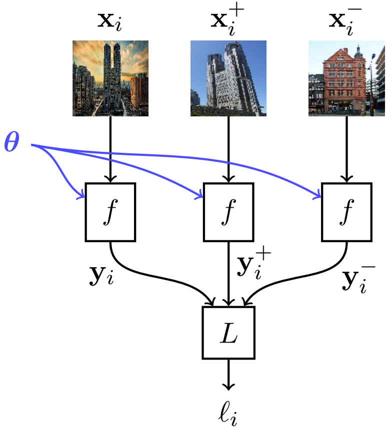
deep image retrieval: architecture



- query $\mathbf{x}_i$, relevant $\mathbf{x}_i^+$ (same building), irrelevant $\mathbf{x}_i^-$ (other building)
- $\mathbf{x}_i, \mathbf{x}_i^+, \mathbf{x}_i^-$ go through function f including features, RPN, pooling
- triplet loss ℓ_i measured on output $(\mathbf{y}_i, \mathbf{y}_i^+, \mathbf{y}_i^-)$

Gordo, Almazan, Revaud, Larlus. ECCV 2016. Deep Image Retrieval: Learning Global Representations for Image Search.

deep image retrieval: architecture

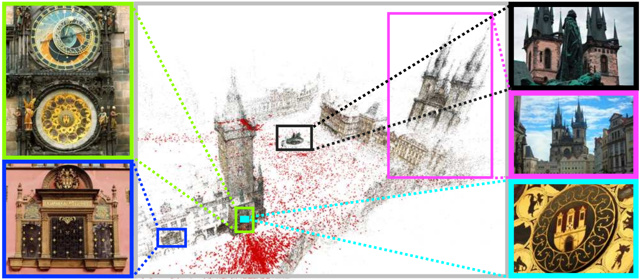


- query $\mathbf{x}_i$, relevant $\mathbf{x}_i^+$ (same building), irrelevant $\mathbf{x}_i^-$ (other building)
- $\mathbf{x}_i, \mathbf{x}_i^+, \mathbf{x}_i^-$ go through function f including features, RPN, pooling
- triplet loss ℓ_i measured on output $(\mathbf{y}_i, \mathbf{y}_i^+, \mathbf{y}_i^-)$

Gordo, Almazan, Revaud, Larlus. ECCV 2016. Deep Image Retrieval: Learning Global Representations for Image Search.

learning from bag-of-words: 3d reconstruction

[Radenovic et al. 2016]



- start from an independent dataset of 7.4M images, no class labels
- clustering, pairwise matching and reconstruction of 713 3d models containing 165k unique images
- 3d models playing the same role as classes in deep image retrieval
- again, fine-tune a network pre-trained on ImageNet for classification

Radenovic, Tolias, Chum. ECCV 2016. CNN Image Retrieval Learns From BoW: Unsupervised Fine-Tuning with Hard Examples.
Schönberger, Radenovic, Chum and Frahm. CVPR 2015. From Single Image Query to Detailed 3D Reconstruction.

learning from bag-of-words: positive pairs








- input query
- positive images found in same model by minimum MAC distance, maximum inliers, or drawn at random from images having at least a given number of inliers (more challenging)

Radenovic, Tolias, Chum. ECCV 2016. CNN Image Retrieval Learns From BoW: Unsupervised Fine-Tuning with Hard Examples.

learning from bag-of-words: positive pairs

- input query
- positive images found in same model by minimum MAC distance, maximum inliers, or drawn at random from images having at least a given number of inliers (more challenging)

Radenovic, Tolias, Chum. ECCV 2016. CNN Image Retrieval Learns From BoW: Unsupervised Fine-Tuning with Hard Examples.

learning from bag-of-words: positive pairs

- input query
- positive images found in same model by minimum MAC distance, maximum inliers, or drawn at random from images having at least a given number of inliers (more challenging)

Radenovic, Tolias, Chum. ECCV 2016. CNN Image Retrieval Learns From BoW: Unsupervised Fine-Tuning with Hard Examples.

learning from bag-of-words: positive pairs

- input query
- positive images found in same model by minimum MAC distance, maximum inliers, or drawn at random from images having at least a given number of inliers (more challenging)

Radenovic, Tolias, Chum. ECCV 2016. CNN Image Retrieval Learns From BoW: Unsupervised Fine-Tuning with Hard Examples.

learning from bag-of-words: negative pairs

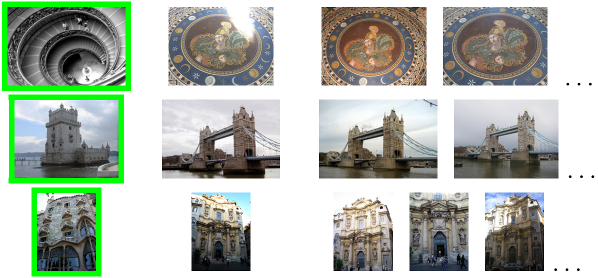
- input query
- negative images found in different models
- hard negatives are most similar to query, *i.e.* with minimum MAC distance
- hardest negative, nearest neighbors from all other models, or nearest neighbors, one per model (higher variability)

learning from bag-of-words: negative pairs

- input query
- negative images found in different models
- hard negatives are most similar to query, *i.e.* with minimum MAC distance
- hardest negative, nearest neighbors from all other models, or nearest neighbors, one per model (higher variability)

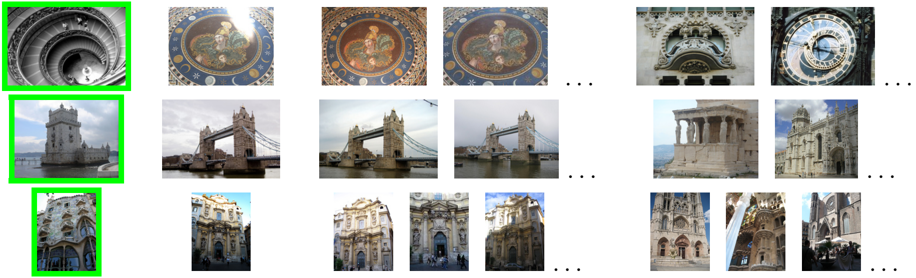
Radenovic, Tolias, Chum. ECCV 2016. CNN Image Retrieval Learns From BoW: Unsupervised Fine-Tuning with Hard Examples.

learning from bag-of-words: negative pairs



- input query
- negative images found in different models
- hard negatives are most similar to query, *i.e.* with minimum MAC distance
- hardest negative, nearest neighbors from all other models, or nearest neighbors, one per model (higher variability)

learning from bag-of-words: negative pairs



- input query
- negative images found in different models
- hard negatives are most similar to query, *i.e.* with minimum MAC distance
- hardest negative, nearest neighbors from all other models, or nearest neighbors, one per model (higher variability)

Radenovic, Tolias, Chum. ECCV 2016. CNN Image Retrieval Learns From BoW: Unsupervised Fine-Tuning with Hard Examples.

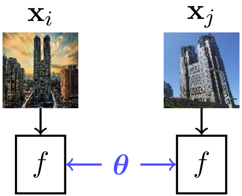
learning from bag-of-words: architecture

$\mathbf{x}_i$ $\mathbf{x}_j$

- input $(\mathbf{x}_i, \mathbf{x}_j)$ of relevant or irrelevant images
- both $\mathbf{x}_i, \mathbf{x}_j$ go through function f including features and MAC pooling
- contrastive loss ℓ_{ij} measured on output $(\mathbf{y}_i, \mathbf{y}_j)$ and target t_{ij}

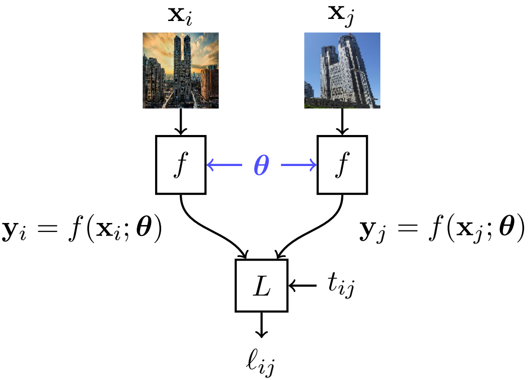
Radenovic, Tolias, Chum. ECCV 2016. CNN Image Retrieval Learns From BoW: Unsupervised Fine-Tuning with Hard Examples.

learning from bag-of-words: architecture



- input $(\mathbf{x}_i, \mathbf{x}_j)$ of relevant or irrelevant images
- both $\mathbf{x}_i, \mathbf{x}_j$ go through function f including features and MAC pooling
- contrastive loss ℓ_{ij} measured on output $(\mathbf{y}_i, \mathbf{y}_j)$ and target t_{ij}

learning from bag-of-words: architecture



- input $(\mathbf{x}_i, \mathbf{x}_j)$ of relevant or irrelevant images
- both $\mathbf{x}_i, \mathbf{x}_j$ go through function f including features and MAC pooling
- contrastive loss ℓ_{ij} measured on output $(\mathbf{y}_i, \mathbf{y}_j)$ and target t_{ij}

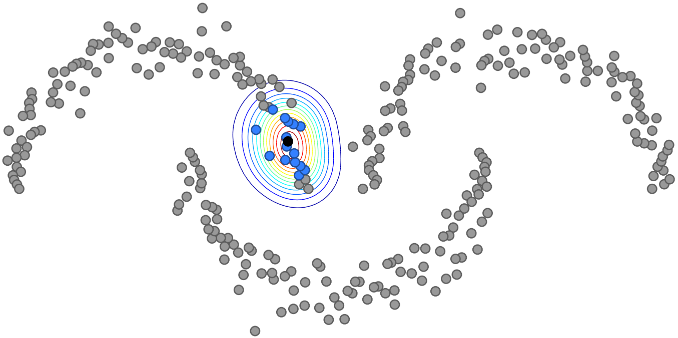
ranking on manifolds

ranking on manifolds: single query

- data points (•), query point (•), nearest neighbors (•)
- iteration × 30

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

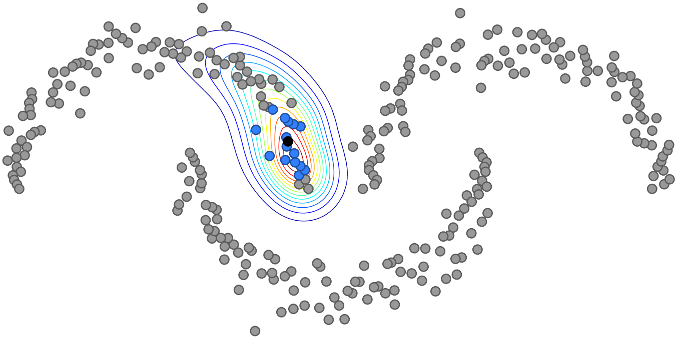
ranking on manifolds: single query



- data points (∘), query point (•), nearest neighbors (∘)
- iteration 0×30

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

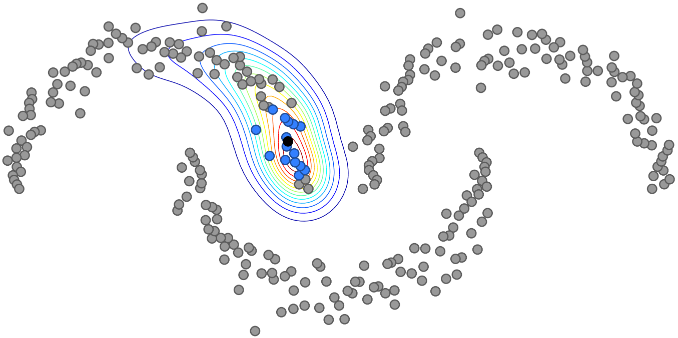
ranking on manifolds: single query



- data points (∘), query point (•), nearest neighbors (∘)
- iteration 1×30

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

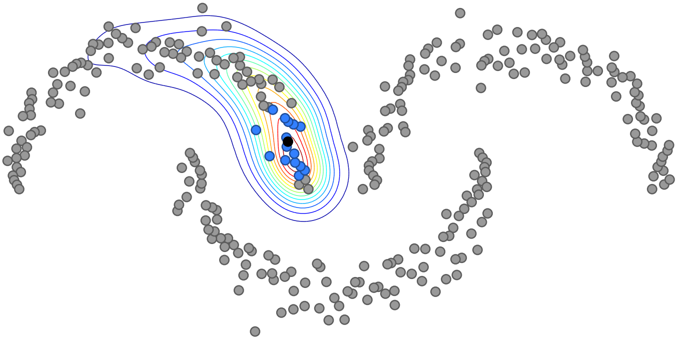
ranking on manifolds: single query



- data points (•), query point (•), nearest neighbors (•)
- iteration 2×30

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

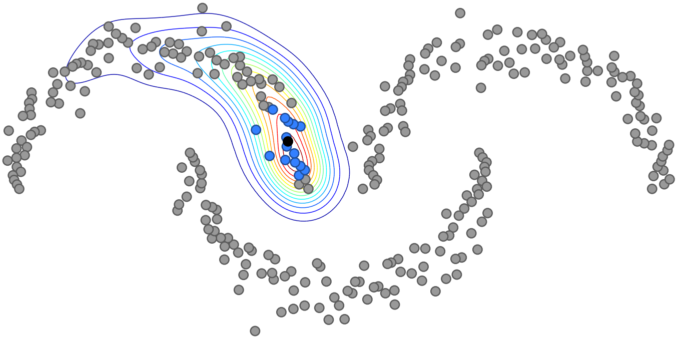
ranking on manifolds: single query



- data points (∘), query point (•), nearest neighbors (∘)
- iteration $3 \times {\color{orange}30}$

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

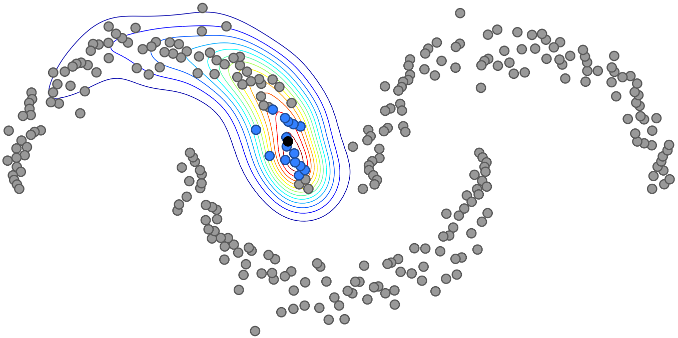
ranking on manifolds: single query



- data points (∘), query point (•), nearest neighbors (∘)
- iteration 4×30

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

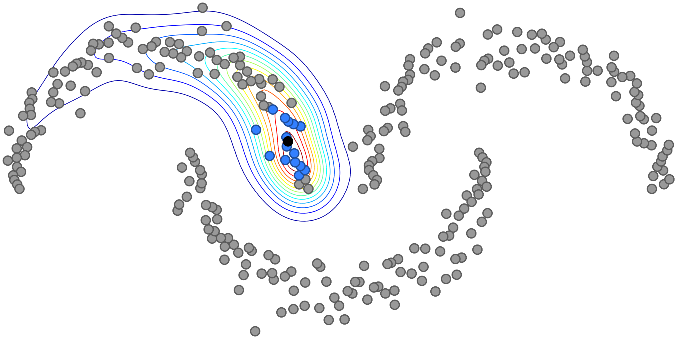
ranking on manifolds: single query



- data points (•), query point (•), nearest neighbors (•)
- iteration 5×30

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

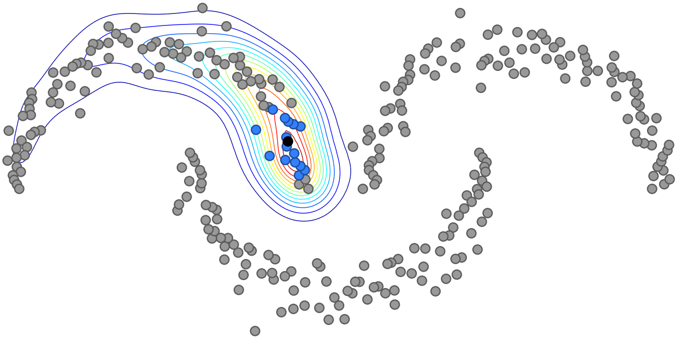
ranking on manifolds: single query



- data points (∘), query point (•), nearest neighbors (∘)
- iteration 6×30

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

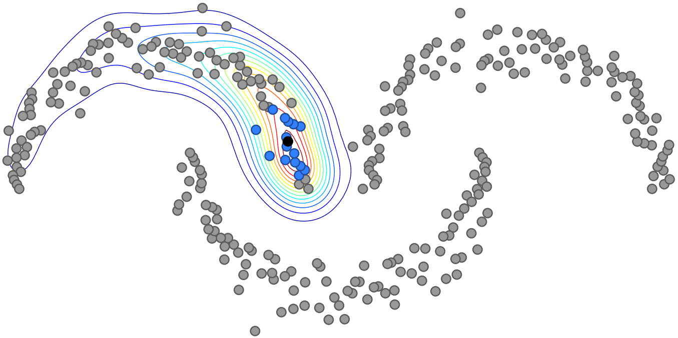
ranking on manifolds: single query



- data points (∘), query point (•), nearest neighbors (∘)
- iteration $7 \times \color{orange}{30}$

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

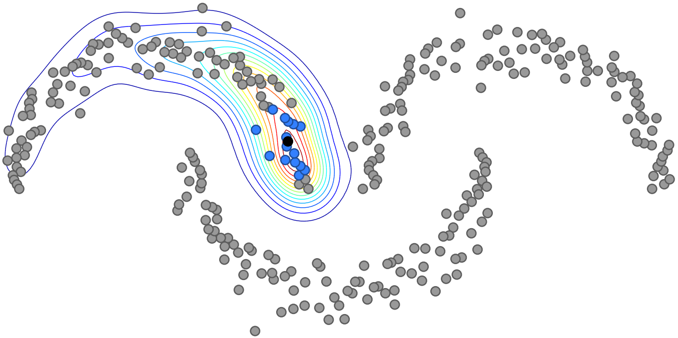
ranking on manifolds: single query



- data points (∘), query point (•), nearest neighbors (∘)
- iteration 8×30

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

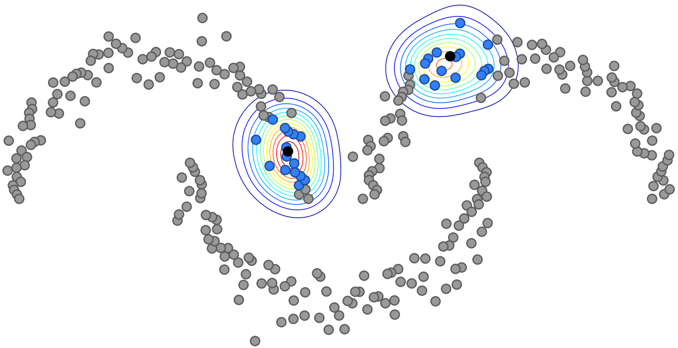
ranking on manifolds: single query



- data points (∘), query point (•), nearest neighbors (•)
- iteration 9×30

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

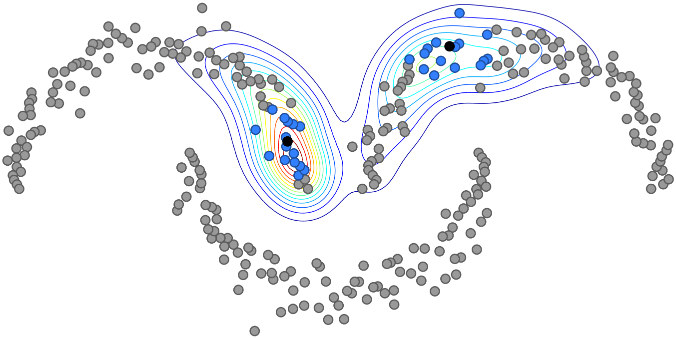
ranking on manifolds: multiple queries



- data points (∘), query points (•), nearest neighbors (∘)
- iteration 0×30

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

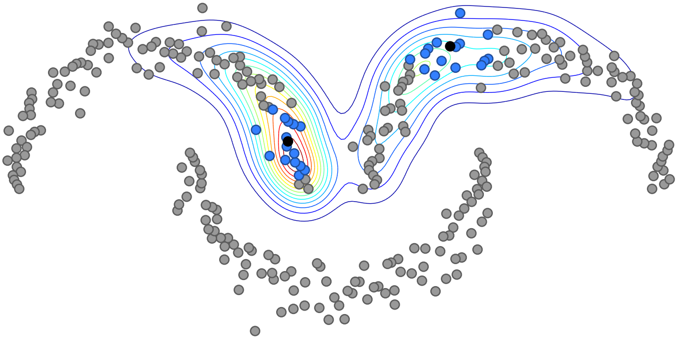
ranking on manifolds: multiple queries



- data points (∘), query points (•), nearest neighbors (∘)
- iteration 1×30

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

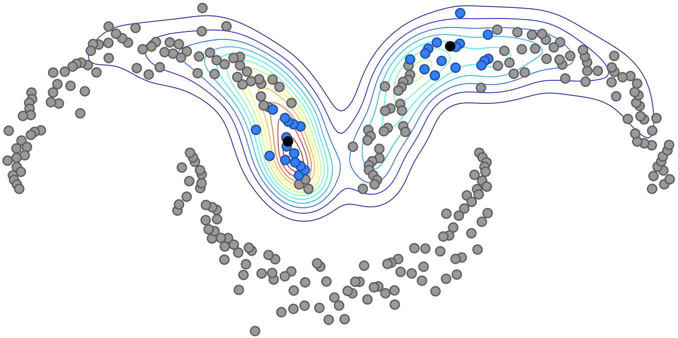
ranking on manifolds: multiple queries



- data points (∘), query points (•), nearest neighbors (∘)
- iteration 2×30

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

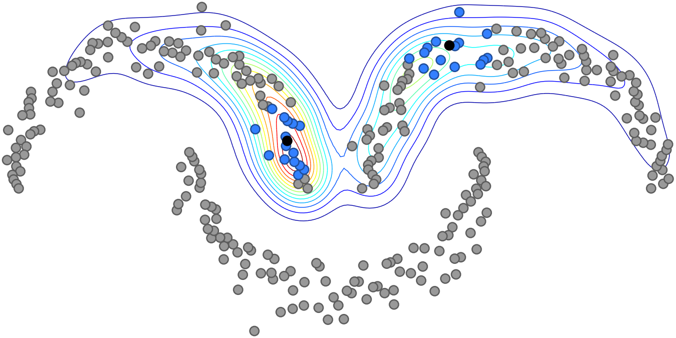
ranking on manifolds: multiple queries



- data points (∘), query points (•), nearest neighbors (∘)
- iteration 3×30

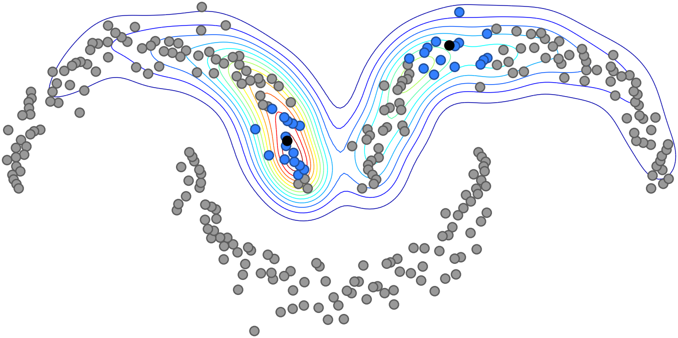
Zhou, Weston, Gretton, Bousquet and Schölkopf. NIPS 2003. Ranking on Data Manifolds.

ranking on manifolds: multiple queries



- data points (∘), query points (•), nearest neighbors (∘)
- iteration 4×30

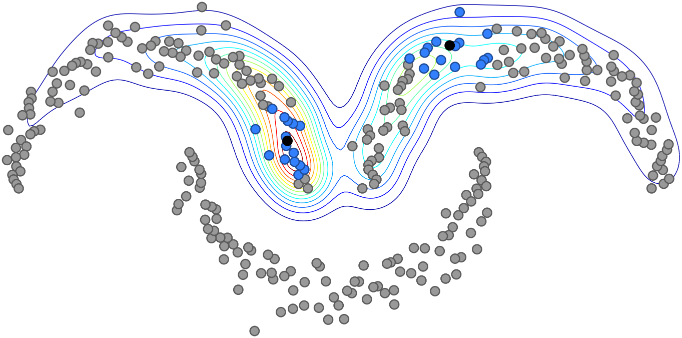
ranking on manifolds: multiple queries



- data points (∘), query points (•), nearest neighbors (∘)
- iteration 5×30

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

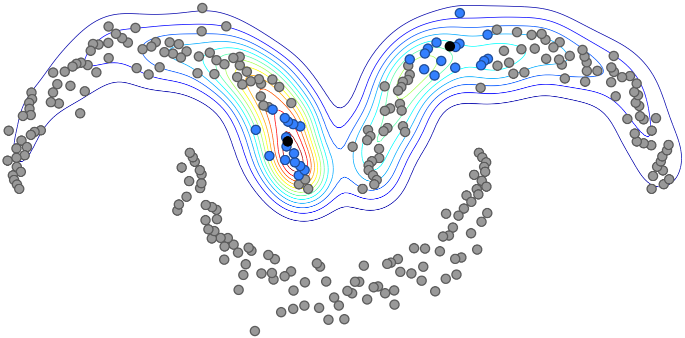
ranking on manifolds: multiple queries



- data points (∘), query points (•), nearest neighbors (∘)
- iteration $6 \times \color{orange}{30}$

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

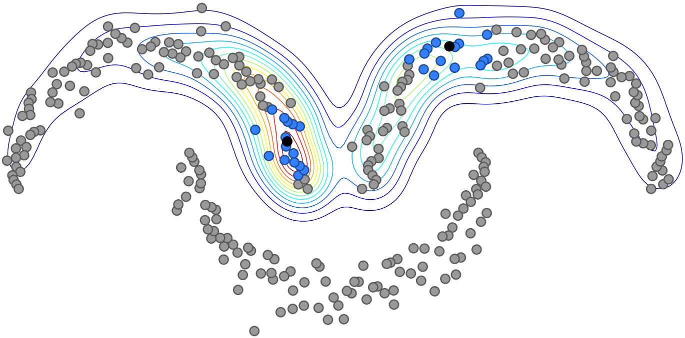
ranking on manifolds: multiple queries



- data points (∘), query points (•), nearest neighbors (∘)
- iteration 7×30

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

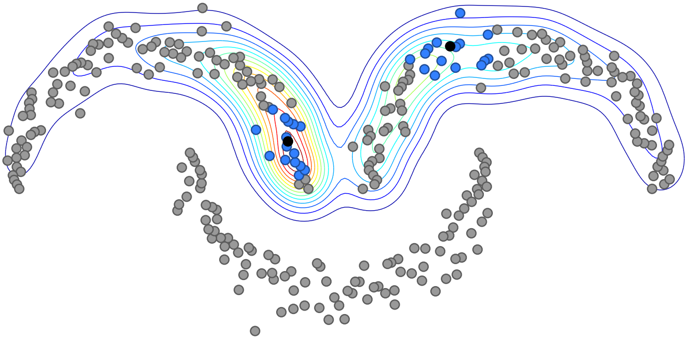
ranking on manifolds: multiple queries



- data points (∘), query points (•), nearest neighbors (∘)
- iteration 8×30

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

ranking on manifolds: multiple queries



- data points (∘), query points (•), nearest neighbors (∘)
- iteration 9×30

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 $D = \mathrm{diag}(W\mathbf{1})$ 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 $D = \mathrm{diag}(W\mathbf{1})$ 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 as solving a linear system

[Iscen et al. 2017]

- query: sparse vector $\mathbf{y} \in \mathbb{R}^n$ with nearest neighbor similarities

$$y_i = \sum_{\mathbf{q} \in Q} s(\mathbf{q}, \mathbf{x}_i) \times \mathbb{1}[\mathbf{x}_i \in \mathrm{NN}_X^k(\mathbf{q})]$$

where $Q, X \subset \mathbb{R}^d$ query/data points, $\mathbf{x}_i \in X$, s similarity function

- 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 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]

- query: sparse vector $\mathbf{y} \in \mathbb{R}^n$ with nearest neighbor similarities

$$y_i = \sum_{\mathbf{q} \in Q} s(\mathbf{q}, \mathbf{x}_i) \times \mathbb{1}[\mathbf{x}_i \in \mathrm{NN}_X^k(\mathbf{q})]$$

  where $Q, X \subset \mathbb{R}^d$ query/data points, $\mathbf{x}_i \in X$, s similarity function
- 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 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]

- query: sparse vector $\mathbf{y} \in \mathbb{R}^n$ with nearest neighbor similarities

$$y_i = \sum_{\mathbf{q} \in Q} s(\mathbf{q}, \mathbf{x}_i) \times \mathbb{1}[\mathbf{x}_i \in \mathrm{NN}_X^k(\mathbf{q})]$$

  where $Q, X \subset \mathbb{R}^d$ query/data points, $\mathbf{x}_i \in X$, s similarity function

- regularized Laplacian

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

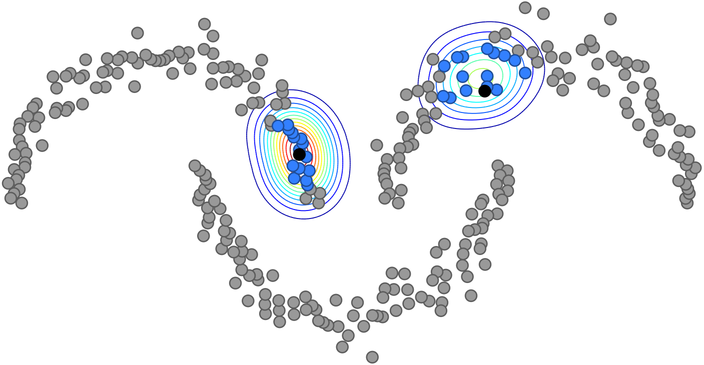
- solve linear system

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

  by conjugate gradient method

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

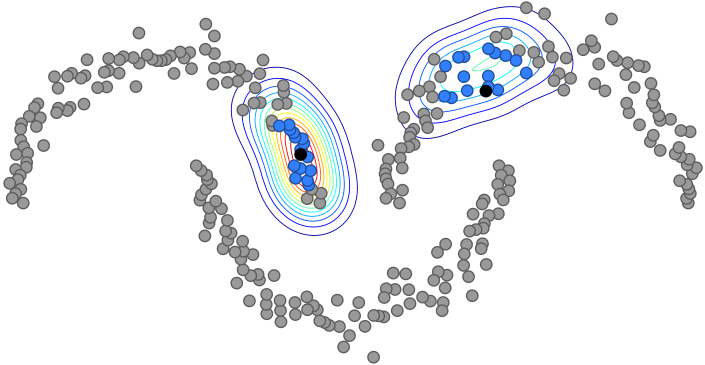
ranking by conjugate gradient



- 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.

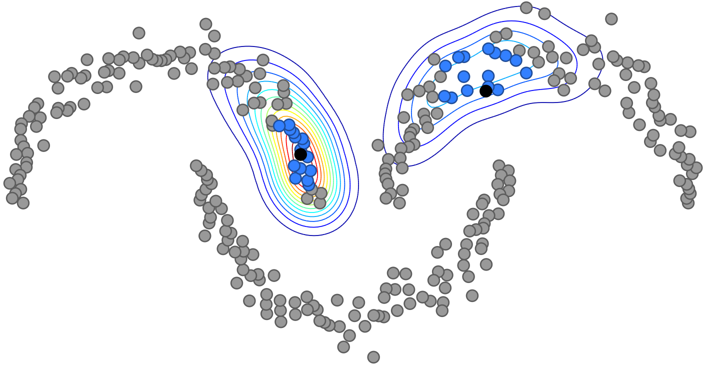
ranking by conjugate gradient



- 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.

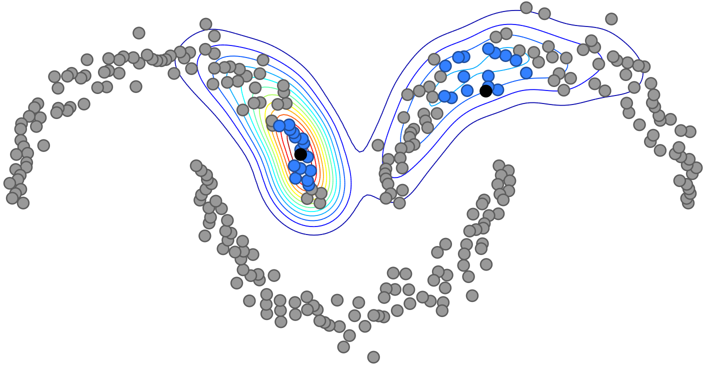
ranking by conjugate gradient



- 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.

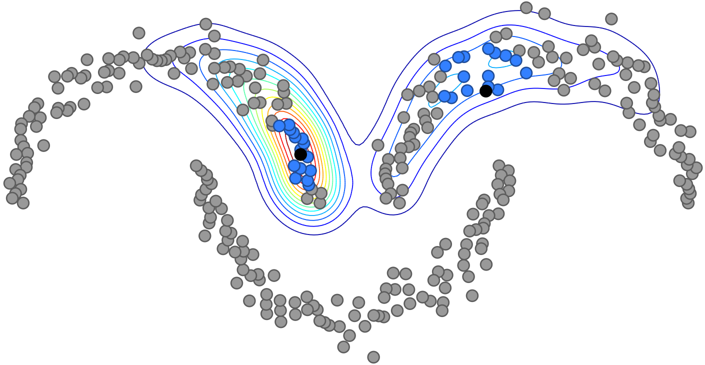
ranking by conjugate gradient



- 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.

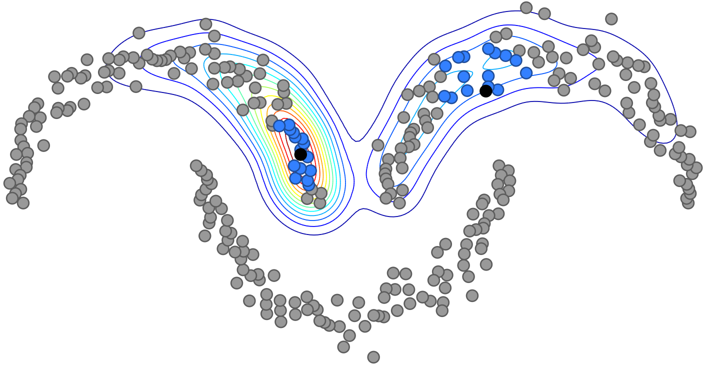
ranking by conjugate gradient



- 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.

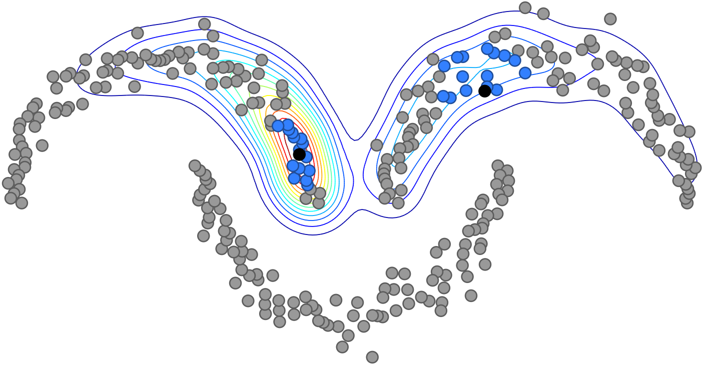
ranking by conjugate gradient



- 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.

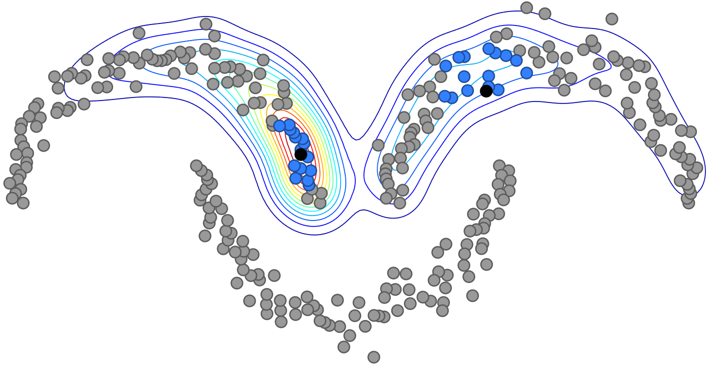
ranking by conjugate gradient



- 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.

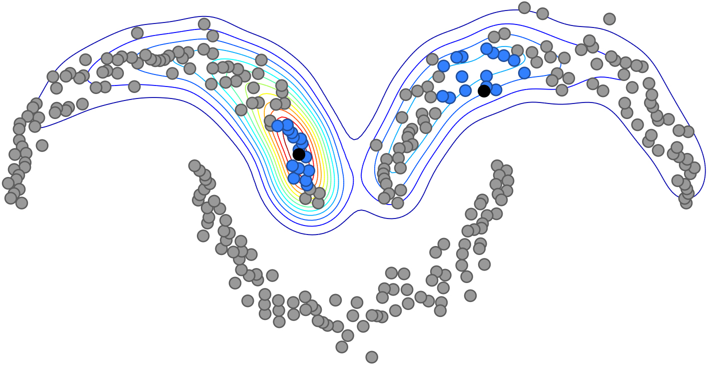
ranking by conjugate gradient



- 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.

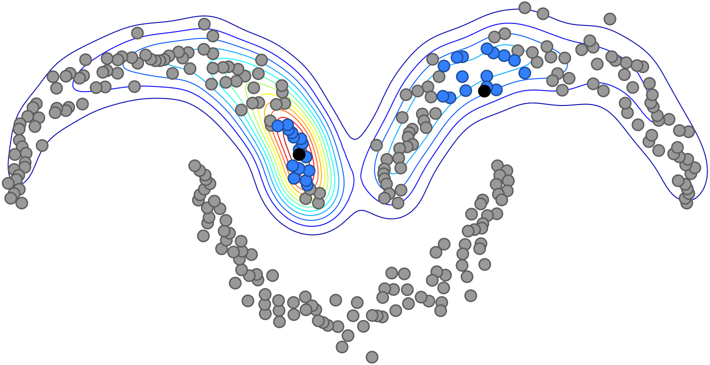
ranking by conjugate gradient



- 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



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

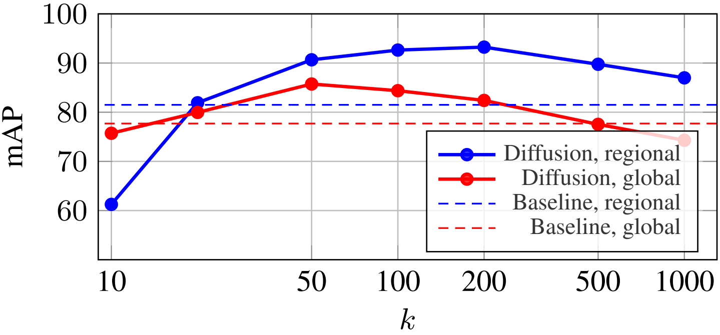
Iscen, Tolias, Avrithis, Furon and Chum. CVPR 2017. Efficient Diffusion on Region Manifolds: Recovering Small Objects With Compact CNN Representations.

ranking as solving a linear system

- represent image by global descriptor or multiple regional descriptors
- perform initial query based on Euclidean nearest neighbors
- re-rank by solving linear system
- ResNet-101 fine-tuned by BoW + R-MAC + re-ranking:
    - mAP 87.1 (95.8) on Oxford5k, 96.5 (96.9) on Paris6k
    - 1 (21) descriptors/image $\times$ 2048 dimensions

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

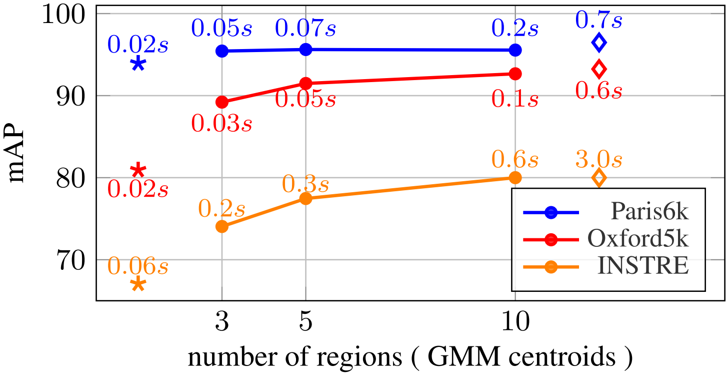
dependence on neighbors, k (Oxford5k)



"small patterns appear more frequently than entire images"

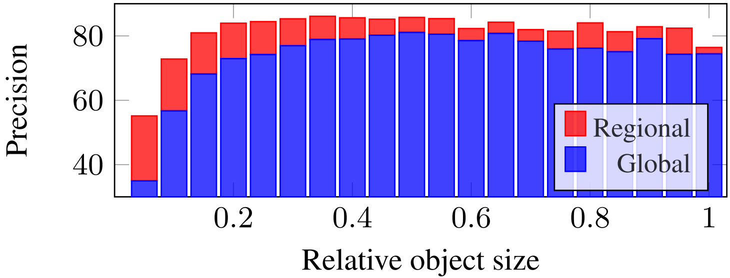
Iscen, Tolias, Avrithis, Furon and Chum. CVPR 2017. Efficient Diffusion on Region Manifolds: Recovering Small Objects With Compact CNN Representations.

global → regional



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

small objects (INSTRE)



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

fast spectral ranking

faster than CG?

- want to solve $\mathcal{L}_\alpha \mathbf{x} = \mathbf{y}$
- could invert $\mathcal{L}_\alpha$ offline, but it wouldn't be sparse
- could approximate $\mathcal{L}_\alpha^{-1}$ by $\Phi^\top \Phi$ where Φ is a (sparse) $r \times n$ matrix with $r \ll n$; then

$$\mathbf{x} \approx \Phi^\top \Phi \mathbf{y}$$

- but how to compute Φ without ever inverting $\mathcal{L}_\alpha$?
- still, there is no generalization; even α is given in advance

faster than CG?

- want to solve $\mathcal{L}_\alpha \mathbf{x} = \mathbf{y}$
- could invert $\mathcal{L}_\alpha$ offline, but it wouldn't be sparse
- could approximate $\mathcal{L}_\alpha^{-1}$ by $\Phi^\top \Phi$ where Φ is a (sparse) $r \times n$ matrix with $r \ll n$; then

$$\mathbf{x} \approx \Phi^\top \Phi \mathbf{y}$$

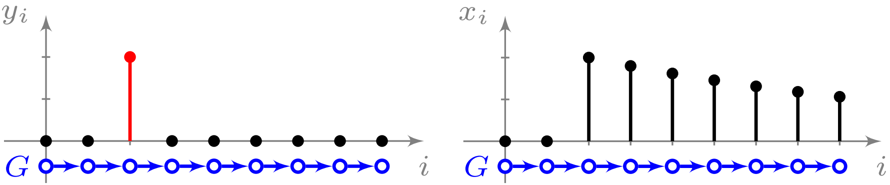
- but how to compute Φ without ever inverting $\mathcal{L}_\alpha$?
- still, there is no generalization; even α is given in advance

faster than CG?

- want to solve $\mathcal{L}_\alpha \mathbf{x} = \mathbf{y}$
- could invert $\mathcal{L}_\alpha$ offline, but it wouldn't be sparse
- could approximate $\mathcal{L}_\alpha^{-1}$ by $\Phi^\top \Phi$ where Φ is a (sparse) $r \times n$ matrix with $r \ll n$; then
$$\mathbf{x} \approx \Phi^\top \Phi \mathbf{y}$$
- but how to compute Φ without ever inverting $\mathcal{L}_\alpha$?
- still, there is no generalization; even α is given in advance

searching on manifolds as smoothing

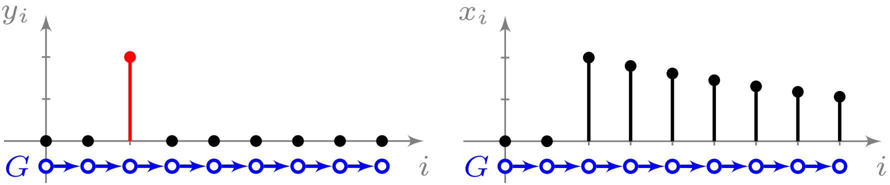
[Iscen et al. 2018]



- 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 $H(z) := (1-\alpha)\sum_{t=0}^{\infty}(az^{-1})^t = (1-\alpha)/(1-\alpha z^{-1})$

searching on manifolds as smoothing

[Iscen et al. 2018]

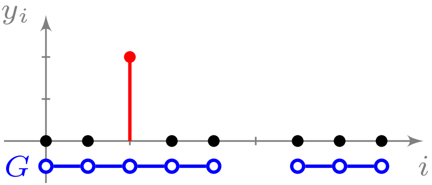


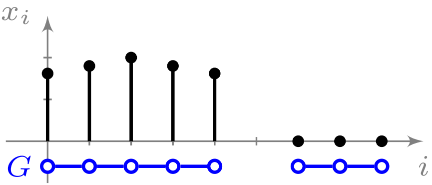
- exponential moving average filter
- output given by $x_i := (1-\alpha)\sum_{t=0}^{\infty} \alpha^t y_{i-t}$
- or by recurrence $x_i = \alpha x_{i-1} + (1-\alpha) y_i$
- impulse response $h_i = (1-\alpha)\alpha^i u_i$
- transfer function $H(z) := (1-\alpha)\sum_{t=0}^{\infty}(az^{-1})^t = (1-\alpha)/(1-\alpha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

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

searching 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.

searching 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 $h(\mathcal{W})$

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

searching 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 $h(\mathcal{W})$

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

searching on manifolds as smoothing

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

- eigenvalue decomposition of $\mathcal{W}$
- low-rank approximation
- (under conditions on h and Λ)
- dot-product search
- linear graph filter in frequency domain

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

searching on manifolds as smoothing

$$\mathbf{x} = h\left(U \; \Lambda \; U^\top \right) \mathbf{y}$$

- eigenvalue decomposition of $\mathcal{W}$
- low-rank approximation
- (under conditions on h and Λ)
- dot-product search
- linear graph filter in frequency domain

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

searching on manifolds as smoothing

$$\mathbf{x} \approx h\left(U \; \Lambda \; U^\top \right) \mathbf{y}$$

- eigenvalue decomposition of $\mathcal{W}$
- low-rank approximation
- (under conditions on h and Λ)
- dot-product search
- linear graph filter in frequency domain

searching on manifolds as smoothing

$$\mathbf{x} \approx U \, h\left(\Lambda \right) U^\top \mathbf{y}$$

- eigenvalue decomposition of $\mathcal{W}$
- low-rank approximation
- (under conditions on h and Λ)
- dot-product search
- linear graph filter in frequency domain

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

searching on manifolds as smoothing

$$\mathbf{x} \approx U \, h\Big(\underbrace{\Lambda}_{\text{diagonal}}\Big) \, U^\top \underbrace{\mathbf{y}}_{\text{sparse}}$$

- eigenvalue decomposition of $\mathcal{W}$
- low-rank approximation
- (under conditions on h and Λ)
- dot-product search
- linear graph filter in frequency domain

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

searching on manifolds as smoothing

$$\boxed{\mathbf{x}} \approx \boxed{\mathcal{F}^{-1}} \; h \left(\boxed{\Lambda} \right) \boxed{\mathcal{F}} \boxed{\mathbf{y}}$$

- eigenvalue decomposition of $\mathcal{W}$
- low-rank approximation
- (under conditions on h and Λ)
- dot-product search
- linear graph filter in frequency domain

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

interpretation: graph signal processing

- in discrete signal processing, a signal of period n is a vector $\mathbf{s} \in \mathbb{R}^n$ where $s_{\bar{i}} := s_{(i \bmod n)+1}$ for $i \in 1, \ldots, n$
- a shift of $\mathbf{s}$ is the mapping $s_{\bar{i}} \mapsto s_{\overline{i-1}}$; also represented by $\mathbf{s} \mapsto C_n \mathbf{s}$ where C_n is an $n \times n$ circulant zero-one matrix, *e.g.* for $n = 5$

$$C_5 = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

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

interpretation: graph signal processing

- in discrete signal processing, a signal of period n is a vector $\mathbf{s} \in \mathbb{R}^n$ where $s_{\bar{i}} := s_{(i \bmod n)+1}$ for $i \in 1, \ldots, n$
- a shift of $\mathbf{s}$ is the mapping $s_{\bar{i}} \mapsto s_{\overline{i-1}}$; also represented by $\mathbf{s} \mapsto C_n \mathbf{s}$ where C_n is an $n \times n$ circulant zero-one matrix, *e.g.* for $n = 5$

$$C_5 = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

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

interpretation: graph signal processing

- a linear, shift invariant, causal filter is the mapping $\mathbf{s} \mapsto H\mathbf{s}$ where

$$H := h(C_n) = \sum_{t=0}^{\infty} h_t C_n^t$$

- matrix C_n has the eigenvalue decomposition $U \Lambda U^\top$ where $U^\top$ is the $n \times n$ discrete Fourier transform matrix $\mathcal{F}$
- if the series $h(C_n)$ converges, filtering $\mathbf{s} \mapsto H\mathbf{s}$ is written as

$$\mathbf{s} \mapsto \mathcal{F}^{-1} h(\Lambda) \mathcal{F} \mathbf{s}$$

- graph signal processing generalizes the above by replacing C_n with a matrix determined by a graph

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

interpretation: graph signal processing

- a linear, shift invariant, causal filter is the mapping $\mathbf{s} \mapsto H\mathbf{s}$ where

$$H := h(C_n) = \sum_{t=0}^{\infty} h_t C_n^t$$

- matrix C_n has the eigenvalue decomposition $U \Lambda U^\top$ where $U^\top$ is the $n \times n$ discrete Fourier transform matrix $\mathcal{F}$
- if the series $h(C_n)$ converges, filtering $\mathbf{s} \mapsto H\mathbf{s}$ is written as

$$\mathbf{s} \mapsto \mathcal{F}^{-1} h(\Lambda) \mathcal{F} \mathbf{s}$$

- graph signal processing generalizes the above by replacing C_n with a matrix determined by a graph

interpretation: graph signal processing

- a linear, shift invariant, causal filter is the mapping $\mathbf{s} \mapsto H\mathbf{s}$ where

$$H := h(C_n) = \sum_{t=0}^{\infty} h_t C_n^t$$

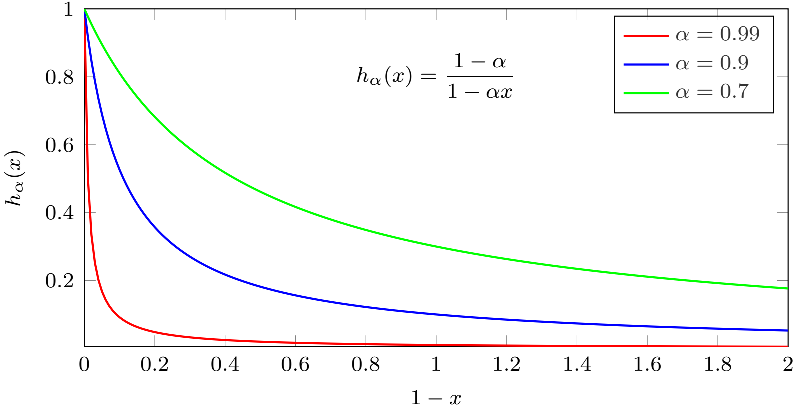
- matrix C_n has the eigenvalue decomposition $U \Lambda U^\top$ where $U^\top$ is the $n \times n$ discrete Fourier transform matrix $\mathcal{F}$
- if the series $h(C_n)$ converges, filtering $\mathbf{s} \mapsto H\mathbf{s}$ is written as

$$\mathbf{s} \mapsto \mathcal{F}^{-1} h(\Lambda) \mathcal{F} \mathbf{s}$$

- graph signal processing generalizes the above by replacing C_n with a matrix determined by a graph

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

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.

interpretation: random fields

- a Gaussian Markov random field (GMRF) with precision A and mean $\boldsymbol{\mu}$ can be parametrized as

$$p(\mathbf{x}) := \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, A^{-1}) \propto e^{-E(\mathbf{x}|\mathbf{b},A)}$$

  where $E(\mathbf{x}|\mathbf{b}, A) := \frac{1}{2}\mathbf{x}^\top A\mathbf{x} - \mathbf{b}^\top\mathbf{x}$ is a quadratic energy
- its expectation $\boldsymbol{\mu} = A^{-1}\mathbf{b}$ is the minimizer of this energy
- our solution $\mathbf{x}^* = \mathcal{L}_\alpha^{-1}\mathbf{y}$ is the expectation of a GMRF with energy

$$f_\alpha(\mathbf{x}) := E(\mathbf{x}|\mathbf{y}, \mathcal{L}_\alpha) = \frac{1}{2}\mathbf{x}^\top \mathcal{L}_\alpha \mathbf{x} - \mathbf{y}^\top\mathbf{x}$$

- if $\hat{\mathbf{x}} := D^{-1/2}\mathbf{x}$, this energy has the same minimizer as

$$\alpha \sum_{i,j} w_{ij} \left\|\hat{x}_i - \hat{x}_j\right\|^2 + (1-\alpha)\left\|\mathbf{x} - \mathbf{y}\right\|^2$$

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

interpretation: random fields

- a Gaussian Markov random field (GMRF) with precision A and mean $\boldsymbol{\mu}$ can be parametrized as

$$p(\mathbf{x}) := \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, A^{-1}) \propto e^{-E(\mathbf{x}|\mathbf{b},A)}$$

  where $E(\mathbf{x}|\mathbf{b}, A) := \frac{1}{2}\mathbf{x}^\top A\mathbf{x} - \mathbf{b}^\top\mathbf{x}$ is a quadratic energy
- its expectation $\boldsymbol{\mu} = A^{-1}\mathbf{b}$ is the minimizer of this energy
- our solution $\mathbf{x}^* = \mathcal{L}_\alpha^{-1}\mathbf{y}$ is the expectation of a GMRF with energy

$$f_\alpha(\mathbf{x}) := E(\mathbf{x}|\mathbf{y}, \mathcal{L}_\alpha) = \frac{1}{2}\mathbf{x}^\top\mathcal{L}_\alpha\mathbf{x} - \mathbf{y}^\top\mathbf{x}$$

- if $\hat{\mathbf{x}} := D^{-1/2}\mathbf{x}$, this energy has the same minimizer as

$$\alpha\sum_{i,j} w_{ij}\left\|\hat{x}_i - \hat{x}_j\right\|^2 + (1-\alpha)\left\|\mathbf{x} - \mathbf{y}\right\|^2$$

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

interpretation: random fields

- a Gaussian Markov random field (GMRF) with precision A and mean $\boldsymbol{\mu}$ can be parametrized as

$$p(\mathbf{x}) := \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, A^{-1}) \propto e^{-E(\mathbf{x}|\mathbf{b},A)}$$

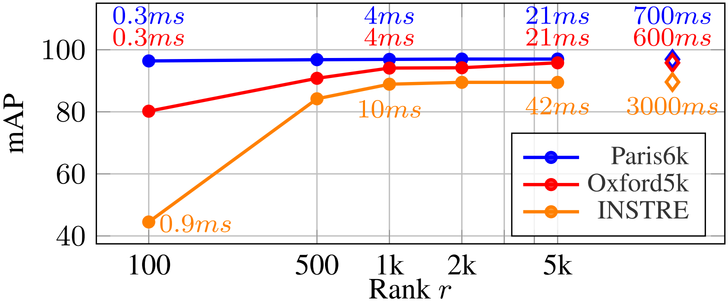
  where $E(\mathbf{x}|\mathbf{b}, A) := \frac{1}{2}\mathbf{x}^\top A\mathbf{x} - \mathbf{b}^\top\mathbf{x}$ is a quadratic energy
- its expectation $\boldsymbol{\mu} = A^{-1}\mathbf{b}$ is the minimizer of this energy
- our solution $\mathbf{x}^* = \mathcal{L}_\alpha^{-1}\mathbf{y}$ is the expectation of a GMRF with energy

$$f_\alpha(\mathbf{x}) := E(\mathbf{x}|\mathbf{y}, \mathcal{L}_\alpha) = \frac{1}{2}\mathbf{x}^\top\mathcal{L}_\alpha\mathbf{x} - \mathbf{y}^\top\mathbf{x}$$

- if $\hat{\mathbf{x}} := D^{-1/2}\mathbf{x}$, this energy has the same minimizer as

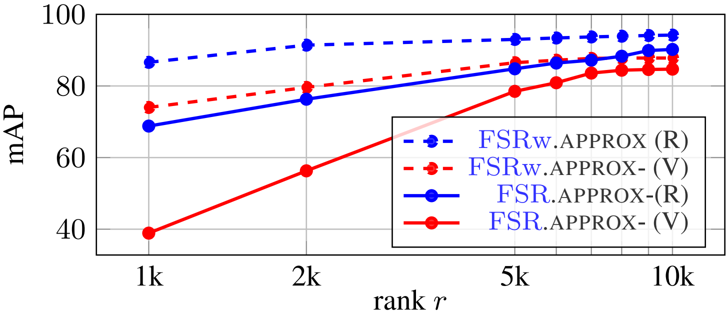
$$\alpha \sum_{i,j} w_{ij} \|\hat{x}_i - \hat{x}_j\|^2 + (1-\alpha)\|\mathbf{x} - \mathbf{y}\|^2$$

small scale (21 descriptors/image)



- in summary: same performance as CG, two orders of magnitude faster, but $3\times$ space needed

Oxford105k (5 descriptors/image)



- in summary: same performance as CG, two orders of magnitude faster, but $3\times$ space needed

hard examples?

- red: drift
- blue: incorrect annotations

mining on manifolds

embeddings for manifold similarity

- the $n \times n$ kernel matrix

$$K = h(\mathcal{W}) = U h(\Lambda) U^\top$$

expresses the pairwise manifold similarity of all data points

- if $h(x) > 0$ for $x \in \mathbb{R}$, which holds for h_α, then K is positive-definite and there is an $n \times n$ matrix Φ such that $K = \Phi^\top \Phi$
- a particular choice is

$$\Phi = h(\Lambda)^{1/2} U^\top$$

- if we choose a rank-r approximation instead, then Φ is $r \times n$ and defines a low-dimensional embedding onto $\mathbb{R}^r$
- so why not learn an embedding on a training set such that it generalizes manifold similarity to other data sets?

embeddings for manifold similarity

- the $n \times n$ kernel matrix

$$K = h(\mathcal{W}) = U h(\Lambda) U^\top$$

  expresses the pairwise manifold similarity of all data points
- if $h(x) > 0$ for $x \in \mathbb{R}$, which holds for h_α, then K is positive-definite and there is an $n \times n$ matrix Φ such that $K = \Phi^\top \Phi$
- a particular choice is

$$\Phi = h(\Lambda)^{1/2} U^\top$$

- if we choose a rank-r approximation instead, then Φ is $r \times n$ and defines a low-dimensional embedding onto $\mathbb{R}^r$
- so why not learn an embedding on a training set such that it generalizes manifold similarity to other data sets?

embeddings for manifold similarity

- the $n \times n$ kernel matrix

$$K = h(\mathcal{W}) = U h(\Lambda) U^\top$$

  expresses the pairwise manifold similarity of all data points
- if $h(x) > 0$ for $x \in \mathbb{R}$, which holds for h_α, then K is positive-definite and there is an $n \times n$ matrix Φ such that $K = \Phi^\top \Phi$
- a particular choice is

$$\Phi = h(\Lambda)^{1/2} U^\top$$

- if we choose a rank-r approximation instead, then Φ is $r \times n$ and defines a low-dimensional embedding onto $\mathbb{R}^r$
- so why not learn an embedding on a training set such that it generalizes manifold similarity to other data sets?

embeddings for manifold similarity

- the $n \times n$ kernel matrix

$$K = h(\mathcal{W}) = U h(\Lambda) U^\top$$

  expresses the pairwise manifold similarity of all data points
- if $h(x) > 0$ for $x \in \mathbb{R}$, which holds for h_α, then K is positive-definite and there is an $n \times n$ matrix Φ such that $K = \Phi^\top \Phi$
- a particular choice is

$$\Phi = h(\Lambda)^{1/2} U^\top$$

- if we choose a rank-r approximation instead, then Φ is $r \times n$ and defines a low-dimensional embedding onto $\mathbb{R}^r$
- so why not learn an embedding on a training set such that it generalizes manifold similarity to other data sets?

mining on manifolds

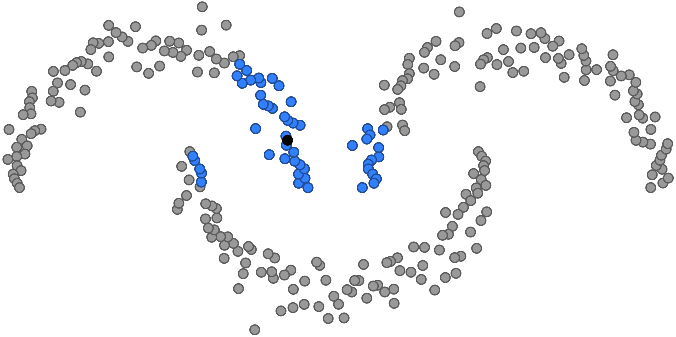
[Iscen et al. 2018]

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

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 $\mathbf{x}$ (•)
- Euclidean nearest neighbors $E(\mathbf{x})$ (∘)

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 $\mathbf{x}$ (•)
- manifold nearest neighbors $M(\mathbf{x})$ (∘)

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

mining on manifolds

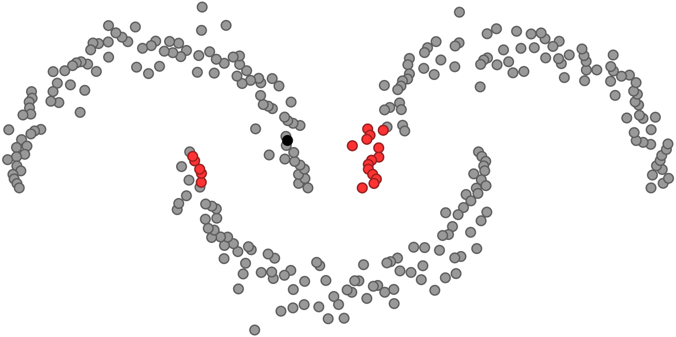
[Iscen et al. 2018]

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

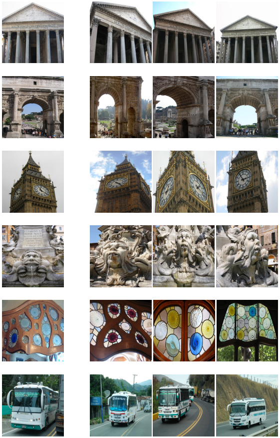
Iscen, Tolias, Avrithis and Chum. CVPR 2018. Mining on Manifolds: Metric Learning without Labels.

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 $X \setminus E(\mathbf{x})$

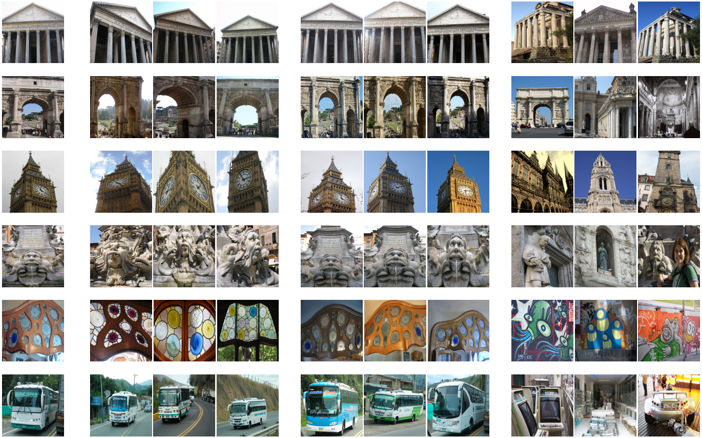
Iscen, Tolias, Avrithis and Chum. CVPR 2018. Mining on Manifolds: Metric Learning without Labels.

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.

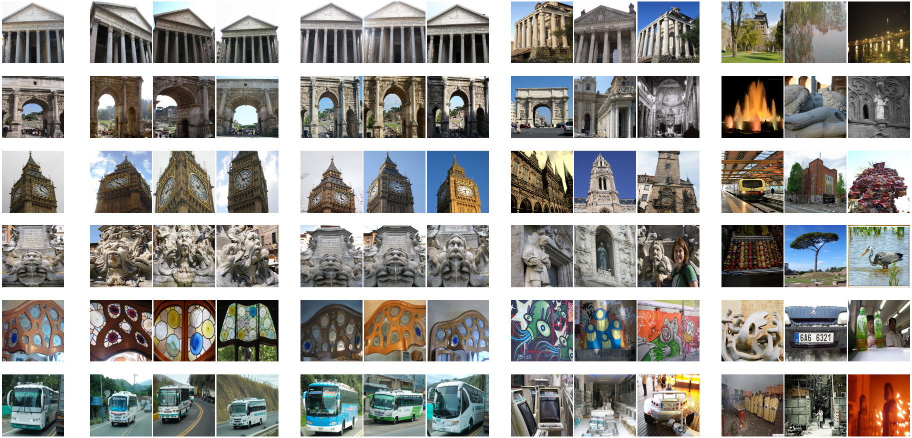
hard positive/negative examples



- 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

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

summary

- pooling CNN representations is best at last convolutional layers
- fine-tuning with constrastive or triplet loss allows transferring to a new domain and learning to rank as opposed to classify
- now that images are represented by a global descriptor or just a few regional descriptors, graph methods are more applicable than ever
- it turns out that query expansion is not just "post processing" but at the core of ranking on manifolds
- there is at least one low-dimensional embedding of manifold similarity, but is dataset-specific
- 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)

summary

- pooling CNN representations is best at last convolutional layers
- fine-tuning with constrastive or triplet loss allows transferring to a new domain and learning to rank as opposed to classify
- now that images are represented by a global descriptor or just a few regional descriptors, graph methods are more applicable than ever
- it turns out that query expansion is not just "post processing" but at the core of ranking on manifolds
- there is at least one low-dimensional embedding of manifold similarity, but is dataset-specific
- 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)

summary

- pooling CNN representations is best at last convolutional layers
- fine-tuning with constrastive or triplet loss allows transferring to a new domain and learning to rank as opposed to classify
- now that images are represented by a global descriptor or just a few regional descriptors, graph methods are more applicable than ever
- it turns out that query expansion is not just "post processing" but at the core of ranking on manifolds
- there is at least one low-dimensional embedding of manifold similarity, but is dataset-specific
- 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)