SIMILARITY AND DISTANCE METRIC LEARNING WITH APPLICATIONS TO COMPUTER VISION

An ECML/PKDD 2015 tutorial

Aurélien Bellet and **Matthieu Cord**

September 7, 2015

LTCI, Télécom ParisTech / CNRS, France
LIP6, Université Pierre et Marie Curie, France

Tutorial webpage: `http://goo.gl/0gqFIm`

1. Overview of metric learning (Aurélien, 2 hours)
2. Applications to computer vision (Matthieu, 1 hour)
3. Wrap-up and questions (15 minutes)

Part 1: Overview of metric learning

OUTLINE FOR THE FIRST PART

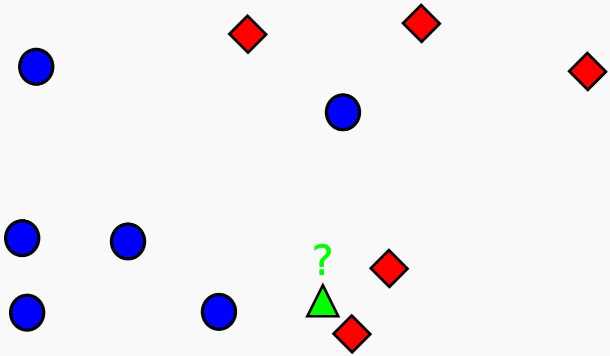
1. Introduction
2. Linear metric learning
3. Nonlinear extensions
4. Large-scale metric learning
5. Metric learning for structured data
6. Generalization guarantees

INTRODUCTION

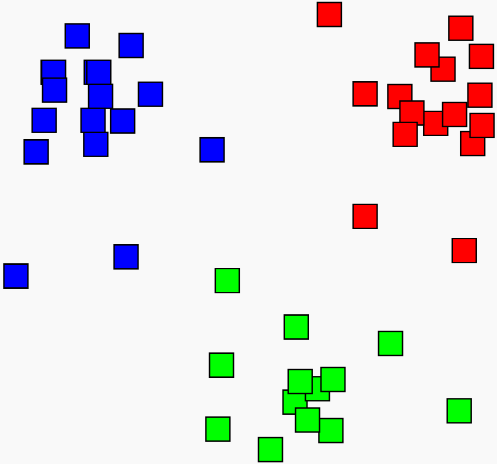
MOTIVATION

- Similarity / distance judgments are essential components of many human cognitive processes (see e.g., [Tversky, 1977])
    - Compare perceptual or conceptual representations
    - Perform recognition, categorization...
- Underlie most machine learning and data mining techniques

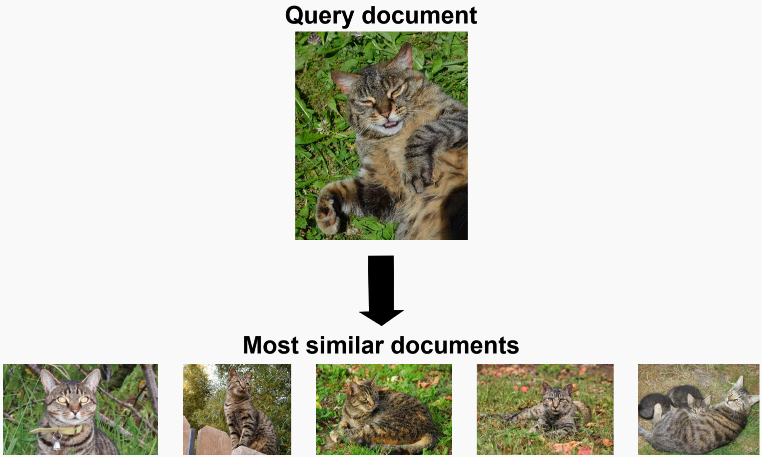
Nearest neighbor classification



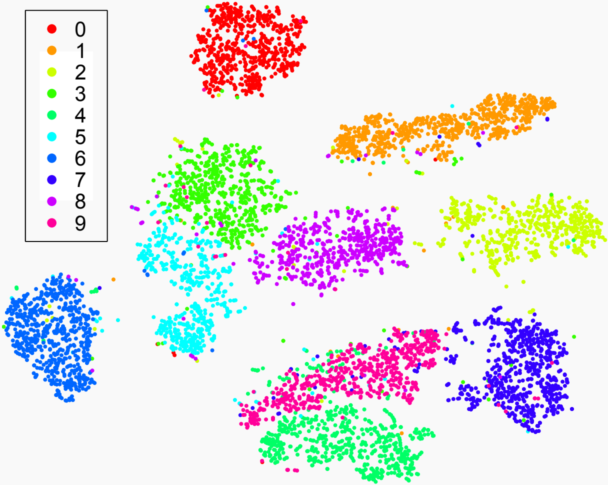
Clustering



Information retrieval



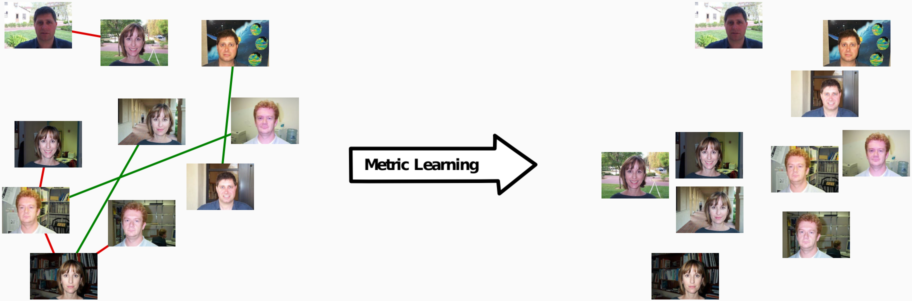
Data visualization



(image taken from [van der Maaten and Hinton, 2008])

- Choice of similarity is crucial to the performance
- Humans weight features differently depending on context [Nosofsky, 1986, Goldstone et al., 1997]
    - Facial recognition vs. determining facial expression
- Fundamental question: how to appropriately measure similarity or distance for a given task?
- Metric learning $\rightarrow$ infer this automatically from data
- Note: we will refer to *distance* or *similarity* indistinctly as *metric*

METRIC LEARNING IN A NUTSHELL



Basic recipe

1. Pick a parametric distance or similarity function
   - Say, a distance $D_M(x, x')$ function parameterized by M
2. Collect similarity judgments on data pairs/triplets
   - $\mathcal{S} = \{(x_i, x_j) : x_i \text{ and } x_j \text{ are similar}\}$
   - $\mathcal{D} = \{(x_i, x_j) : x_i \text{ and } x_j \text{ are dissimilar}\}$
   - $\mathcal{R} = \{(x_i, x_j, x_k) : x_i \text{ is more similar to } x_j \text{ than to } x_k\}$
3. Estimate parameters s.t. metric best agrees with judgments
   - Solve an optimization problem of the form

$$\hat{M} = \arg\min_M \left[\underbrace{\ell(M, \mathcal{S}, \mathcal{D}, \mathcal{R})}_{\text{loss function}} + \underbrace{\lambda reg(M)}_{\text{regularization}} \right]$$

SCOPE OF THE TUTORIAL

- Related topics (not covered)
    - Kernel learning: nonparametric, limited to transductive setting
    - Multiple kernel learning: combine predefined kernels
    - Dimensionality reduction: manifold learning, etc
- Prerequisites
    - None, really
    - Exposure to convex optimization will help

LINEAR METRIC LEARNING

- Mahalanobis (pseudo) distance:

$$D_M(x, x') = \sqrt{(x - x')^T M (x - x')}$$

  where $M \in \mathbb{S}_+^d$ is a symmetric PSD $d \times d$ matrix

- Equivalent to Euclidean distance after linear projection:

$$D_M(x, x') = \sqrt{(x - x')^T L^T L (x - x')} = \sqrt{(Lx - Lx')^T (Lx - Lx')}$$

- If M has rank $k \leq d$, $L \in \mathbb{R}^{k \times d}$ reduces data dimension
- For convenience, work with the squared distance

A first approach [Xing et al., 2002]

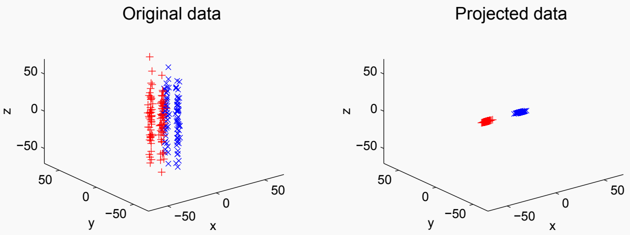
- Targeted task: clustering with side information

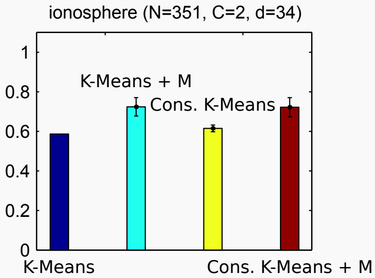
Formulation

$$\begin{aligned}\max_{M\in\mathbb{S}_+^d} \quad & \sum_{(x_i,x_j)\in\mathcal{D}} D_M(x_i,x_j)\\ \text{s.t.} \quad & \sum_{(x_i,x_j)\in\mathcal{S}} D_M^2(x_i,x_j)\leq 1\end{aligned}$$

- Convex in M and always feasible (take $M = \mathbf{0}$)
- Solved with projected gradient descent
- Time complexity of projection on $\mathbb{S}_+^d$ is $O(d^3)$
- Only look at sums of distances

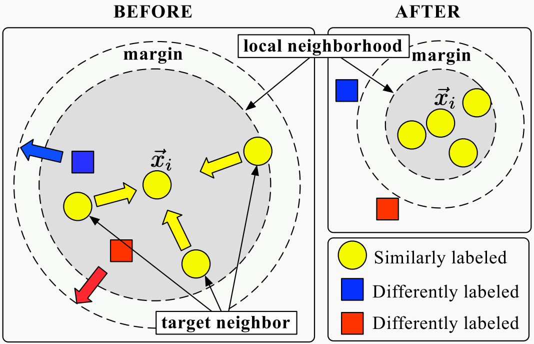
A first approach [Xing et al., 2002]





Large Margin Nearest Neighbor [Weinberger et al., 2005]

- Targeted task: k-NN classification
- Constraints derived from labeled data
  - $\mathcal{S} = \{(\mathbf{x}_i, \mathbf{x}_j) : y_i = y_j, \mathbf{x}_j$ belongs to k-neighborhood of $\mathbf{x}_i\}$
  - $\mathcal{R} = \{(\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k) : (\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{S}, y_i \neq y_k\}$



Large Margin Nearest Neighbor [Weinberger et al., 2005]

Formulation

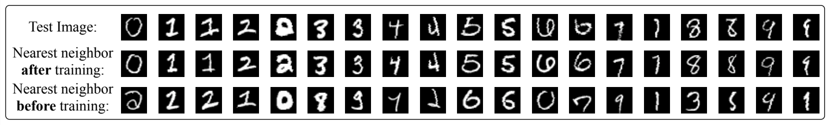
$$\min_{M \in \mathbb{S}_+^d, \boldsymbol{\xi} \geq 0} \quad (1-\mu) \sum_{(x_i, x_j) \in \mathcal{S}} D_M^2(x_i, x_j) \quad + \quad \mu \sum_{i,j,k} \xi_{ijk}$$

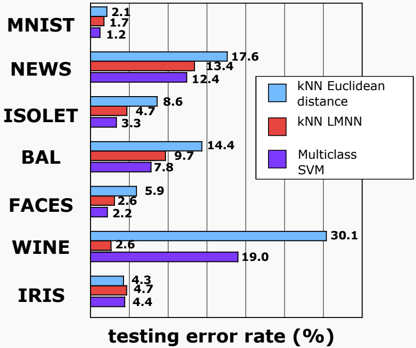
$$\text{s.t.} \quad D_M^2(x_i, x_k) - D_M^2(x_i, x_j) \geq 1 - \xi_{ijk} \quad \forall (x_i, x_j, x_k) \in \mathcal{R}$$

$\mu \in [0, 1]$ trade-off parameter

- Convex formulation, unlike NCA [Goldberger et al., 2004]
- Number of constraints in the order of kn^2
  - Solver based on projected gradient descent with working set
  - Simple alternative: only consider closest "impostors"
- Chicken and egg situation: which metric to build constraints?
- Possible overfitting in high dimensions

Large Margin Nearest Neighbor [Weinberger et al., 2005]





Algorithms for other tasks

- Learning to rank [McFee and Lanckriet, 2010, Lim and Lanckriet, 2014]
- Multi-task learning [Parameswaran and Weinberger, 2010]
- Transfer learning [Zhang and Yeung, 2010]
- Semi-supervised learning [Hoi et al., 2008]
- Domain adaptation [Kulis et al., 2011, Geng et al., 2011]

Interesting regularizers

- Add regularization term to prevent overfitting
- Simple choice: $\|\boldsymbol{M}\|_{\mathcal{F}}^2 = \sum_{i,j=1}^d M_{ij}^2$ (Frobenius norm)
    - Used in [Schultz and Joachims, 2003] and many others
- LogDet divergence (used in ITML [Davis et al., 2007])

$$
\begin{aligned}
D_{ld}(\boldsymbol{M}, \boldsymbol{M}_0) &= \operatorname{tr}(\boldsymbol{M}\boldsymbol{M}_0^{-1}) - \log\det(\boldsymbol{M}\boldsymbol{M}_0^{-1}) - d \\
&= \sum_{i,j} \frac{\sigma_i}{\theta_j}(\boldsymbol{v}_i^T\boldsymbol{u}_i)^2 - \sum_i \log\left(\frac{\sigma_i}{\theta_i}\right) - d
\end{aligned}
$$

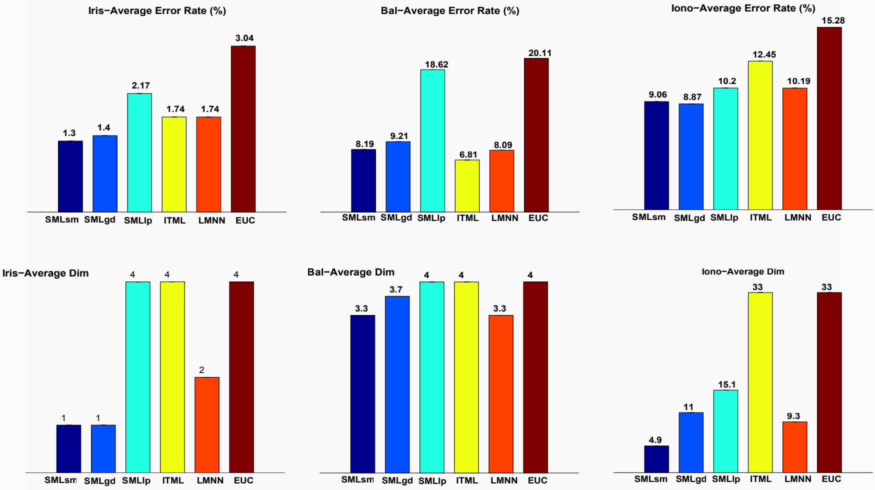
where $\boldsymbol{M} = \boldsymbol{V}\boldsymbol{\Sigma}\boldsymbol{V}^T$ and $\boldsymbol{M}_0 = \boldsymbol{U}\boldsymbol{\Theta}\boldsymbol{U}^T$ is PD

- Remain close to good prior metric $\boldsymbol{M}_0$ (e.g., identity)
- Implicitly ensure that $\boldsymbol{M}$ is PD
- Convex in $\boldsymbol{M}$ (determinant of PD matrix is log-concave)
- Efficient Bregman projections in $O(d^2)$

Interesting regularizers

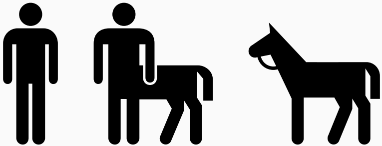
- Mixed $L_{2,1}$ norm: $\|\boldsymbol{M}\|_{2,1} = \sum_{i=1}^{d} \|\boldsymbol{M}_i\|_2$
  - Tends to zero-out entire columns $\rightarrow$ feature selection
  - Used in [Ying et al., 2009]
  - Convex but nonsmooth
  - Efficient proximal gradient algorithms (see e.g., [Bach et al., 2012])
- Trace (or nuclear) norm: $\|\boldsymbol{M}\|_* = \sum_{i=1}^{d} \sigma_i(\boldsymbol{M})$
  - Favors low-rank matrices $\rightarrow$ dimensionality reduction
  - Used in [McFee and Lanckriet, 2010]
  - Convex but nonsmooth
  - Efficient Frank-Wolfe algorithms [Jaggi, 2013]

$L_{2,1}$ norm illustration



(image taken from [Ying et al., 2009])

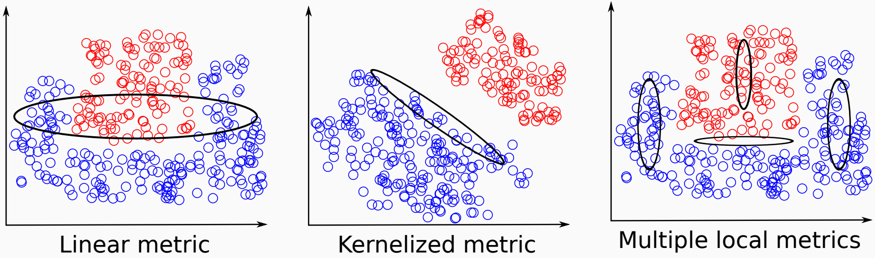
- Mahalanobis distance satisfies the distance axioms
  - Nonnegativity, symmetry, triangle inequality
  - Natural regularization, required by some applications
- In practice, these axioms may be violated
  - By human similarity judgments (see e.g., [Tversky and Gati, 1982])



  - By some good visual recognition systems [Scheirer et al., 2014]
- Alternative: learn bilinear similarity function $S_M(x, x') = x^T M x'$
  - See [Chechik et al., 2010, Bellet et al., 2012b, Cheng, 2013]
  - No PSD constraint on $M \rightarrow$ computational benefits
  - Theory of learning with arbitrary similarity functions [Balcan and Blum, 2006]

NONLINEAR EXTENSIONS

- So far, we have essentially been learning a linear projection
- Advantages
  - Convex formulations
  - Robustness to overfitting
- Drawback
  - Inability to capture nonlinear structure



Definition (Kernel function)

A symmetric function K is a kernel if there exists a mapping function $\phi : \mathcal{X} \to \mathbb{H}$ from the instance space $\mathcal{X}$ to a Hilbert space $\mathbb{H}$ such that K can be written as an inner product in $\mathbb{H}$:

$$K(x, x') = \langle \phi(x), \phi(x') \rangle .$$

Equivalently, K is a kernel if it is positive semi-definite (PSD), i.e.,

$$\sum_{i=1}^{n} \sum_{j=1}^{n} c_i c_j K(x_i, x_j) \geq 0$$

for all finite sequences of $x_1, \ldots, x_n \in \mathcal{X}$ and $c_1, \ldots, c_n \in \mathbb{R}$.

Kernel trick for metric learning

- Notations
  - Kernel $K(\boldsymbol{x}, \boldsymbol{x}') = \langle \phi(\boldsymbol{x}), \phi(\boldsymbol{x}') \rangle$, training data $\{\boldsymbol{x}_i\}_{i=1}^n$
  - $\boldsymbol{\phi}_i \overset{def}{=} \phi(\boldsymbol{x}_i) \in \mathbb{R}^D$, $\boldsymbol{\Phi} \overset{def}{=} [\boldsymbol{\phi}_1, \dots, \boldsymbol{\phi}_n] \in \mathbb{R}^{n \times D}$
- Mahalanobis distance in kernel space

$$D^2_{\boldsymbol{M}}(\boldsymbol{\phi}_i, \boldsymbol{\phi}_j) = (\boldsymbol{\phi}_i - \boldsymbol{\phi}_j)^T \boldsymbol{M} (\boldsymbol{\phi}_i - \boldsymbol{\phi}_j) = (\boldsymbol{\phi}_i - \boldsymbol{\phi}_j)^T \boldsymbol{L}^T \boldsymbol{L} (\boldsymbol{\phi}_i - \boldsymbol{\phi}_j)$$

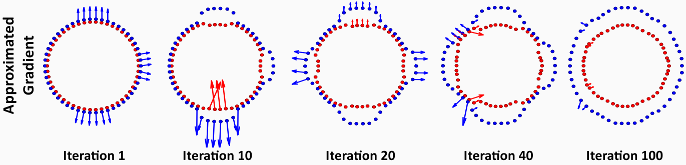
- Setting $\boldsymbol{L}^T = \boldsymbol{\Phi} \boldsymbol{U}^T$, where $\boldsymbol{U} \in \mathbb{R}^{D \times n}$, we get

$$D^2_{\boldsymbol{M}}(\phi(\boldsymbol{x}), \phi(\boldsymbol{x}')) = (\boldsymbol{k} - \boldsymbol{k}')^T \boldsymbol{M} (\boldsymbol{k} - \boldsymbol{k}')$$

  - $\boldsymbol{M} = \boldsymbol{U}^T \boldsymbol{U} \in \mathbb{R}^{n \times n}$, $\boldsymbol{k} = \boldsymbol{\Phi}^T \phi(\boldsymbol{x}) = [K(\boldsymbol{x}_1, \boldsymbol{x}), \dots, K(\boldsymbol{x}_n, \boldsymbol{x})]^T$
- Justified by a representer theorem [Chatpatanasiri et al., 2010]

Kernel trick for metric learning

- Similar trick as kernel SVM
  - Use a nonlinear kernel (e.g., Gaussian RBF)
  - Inexpensive computations through the kernel
  - Nonlinear metric learning while retaining convexity
- Need to learn $O(n^2)$ parameters
- Linear metric learning algorithm must be kernelized
  - Interface to data limited to inner products only
  - Several algorithms shown to be kernelizable
- General approach [Chatpatanasiri et al., 2010]:
  1. Kernel PCA: nonlinear projection to low-dimensional space
  2. Apply linear metric learning algorithm to projected data

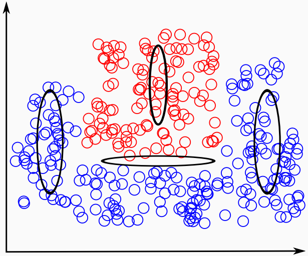


- More flexible approach: learn nonlinear mapping ϕ to optimize

$$D_{\phi}(x, x') = \|\phi(x) - \phi(x')\|_2$$

- Possible parameterizations for ϕ:
  - Regression trees [Kedem et al., 2012]
  - Deep neural nets [Chopra et al., 2005, Hu et al., 2014]
    $\rightarrow$ covered in second part of the tutorial
  - ...
- Nonconvex formulations

- Simple linear metrics perform well locally
- Idea: different metrics for different parts of the space
- Various issues
    - How to split the space?
    - How to avoid blowing up the number of parameters to learn?
    - How to make local metrics “mutually comparable”?
    - …



Multiple Metric LMNN [Weinberger and Saul, 2009]

- Group data into C clusters
- Learn a metric for each cluster in a coupled fashion

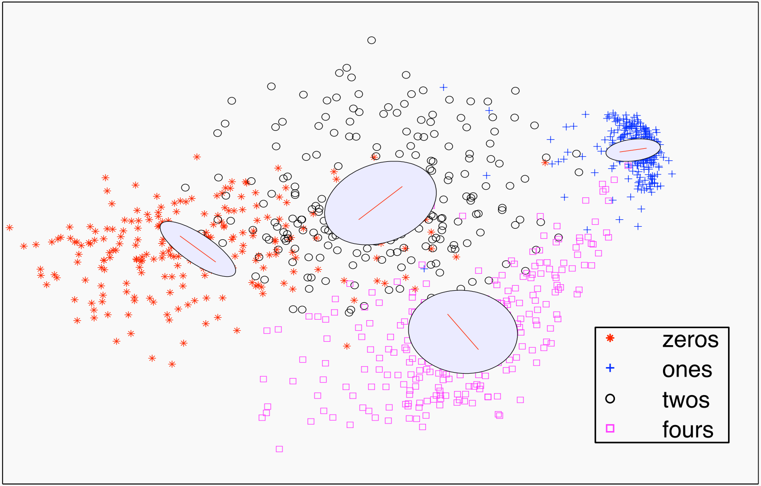
Formulation

$$\min_{\substack{M_1,\dots,M_C \\ \boldsymbol{\xi}\geq 0}} \quad (1-\mu)\sum_{(x_i,x_j)\in\mathcal{S}} D^2_{M_{C(x_j)}}(\mathbf{x}_i,\mathbf{x}_j) \quad + \quad \mu\sum_{i,j,k}\xi_{ijk}$$

$$\text{s.t.} \quad D^2_{M_{C(x_k)}}(\mathbf{x}_i,\mathbf{x}_k) - D^2_{M_{C(x_j)}}(\mathbf{x}_i,\mathbf{x}_j) \geq 1-\xi_{ijk} \quad \forall(\mathbf{x}_i,\mathbf{x}_j,\mathbf{x}_k)\in\mathcal{R}$$

- Remains convex
- Computationally more expensive than standard LMNN
- Subject to overfitting
  - Many parameters
  - Lack of smoothness in metric change

Multiple Metric LMNN [Weinberger and Saul, 2009]



Sparse Compositional Metric Learning [Shi et al., 2014]

- Learn a metric for each point in feature space
- Use the following parameterization

$$D_w^2(x, x') = (x - x')^T \left(\sum_{k=1}^{K} w_k(x) \mathbf{b}_k \mathbf{b}_k^T \right) (x - x'),$$

  - $\mathbf{b}_k \mathbf{b}_k^T$: rank-1 basis (generated from training data)
  - $w_k(\mathbf{x}) = (\mathbf{a}_k^T \mathbf{x} + c_k)^2$: weight of basis k
  - $\mathbf{A} \in \mathbb{R}^{d \times K}$ and $\mathbf{c} \in \mathbb{R}^K$: parameters to learn

Sparse Compositional Metric Learning [Shi et al., 2014]

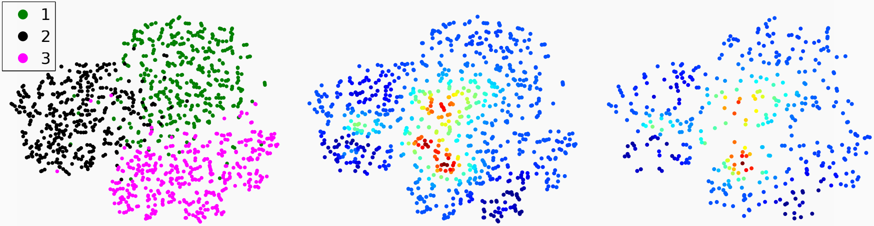
Formulation

$$\min_{\tilde{A}\in\mathbb{R}^{(d+1)\times K}} \sum_{(x_i,x_j,x_k)\in\mathcal{R}} \left[1 + D_w^2(x_i,x_j) - D_w^2(x_i,x_k)\right]_+ + \lambda\|\tilde{A}\|_{2,1}$$

- $\tilde{A}$: stacking A and c
- $[\cdot] = \max(0,\cdot)$: hinge loss

- Nonconvex problem
- Adapts to geometry of data
- More robust to overfitting
  - Limited number of parameters
  - Basis selection
  - Metric varies smoothly over feature space

Sparse Compositional Metric Learning [Shi et al., 2014]



(a) Class membership (b) Trained metrics (c) Test metrics

LARGE-SCALE METRIC LEARNING

- How to deal with large datasets?
  - Number of similarity judgments can grow as $O(n^2)$ or $O(n^3)$
- How to deal with high-dimensional data?
  - Cannot store $d \times d$ matrix
  - Cannot afford computational complexity in $O(d^2)$ or $O(d^3)$

Online learning

- Online algorithm
  - Receive *one* similarity judgment
  - Suffer loss based on current metric
  - Update metric and iterate
- Goal: minimize regret

$$\sum_{t=1}^{T} \ell_t(M_t) - \sum_{t=1}^{T} \ell_t(M^*) \leq f(T),$$

  - ℓ_t: loss suffered at time t
  - M_t: metric learned at time t
  - M^*: best metric in hindsight

Online learning

OASIS [Chechik et al., 2010]

- Set $M^0 = I$
- At step t, receive $(x_i, x_j, x_k) \in \mathcal{R}$ and update by solving

$$\begin{aligned} M^t = \quad & \arg\min_{M,\xi} \quad \frac{1}{2}\|M - M^{t-1}\|_{\mathcal{F}}^2 + C\xi \\ & \text{s.t.} \quad 1 - S_M(x_i, x_j) + S_M(x_i, x_k) \leq \xi \\ & \qquad \quad \xi \geq 0 \end{aligned}$$

  - $S_M(x, x') = x^T M x'$, C trade-off parameter

- Closed-form solution at each iteration
- Trained with 160M triplets in 3 days on 1 CPU

Stochastic and distributed optimization

- Assume metric learning problem of the form

$$\min_{M} \quad \frac{1}{|\mathcal{R}|} \sum_{(x_i,x_j,x_k)\in\mathcal{R}} \ell(M, x_i, x_j, x_k)$$

- Can use Stochastic Gradient Descent
  - Use a random sample (mini-batch) to estimate gradient
  - Better than full gradient descent when n is large
- Can be combined with distributed optimization
  - Distribute triplets on workers
  - Each worker use a mini-batch to estimate gradient
  - Coordinator averages estimates and updates
- See [Xie and Xing, 2014, Clémençon et al., 2015]

Simple workarounds

- Learn a diagonal matrix
  - Used in [Xing et al., 2002, Schultz and Joachims, 2003]
  - Learn d parameters
  - Only a weighting of features...
- Learn metric after dimensionality reduction (e.g., PCA)
  - Used in many papers
  - Potential loss of information
  - Learned metric difficult to interpret

Matrix decompositions

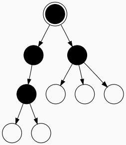
- Low-rank decomposition $\boldsymbol{M} = \boldsymbol{L}^T\boldsymbol{L}$ with $\boldsymbol{L} \in \mathbb{R}^{r\times d}$
    - Used in [Goldberger et al., 2004]
    - Learn $r \times d$ parameters
    - Generally nonconvex, must tune r
- Rank-1 decomposition $\boldsymbol{M} = \sum_{i=1}^{K} w_k \boldsymbol{b}_k \boldsymbol{b}_k^T$
    - Used in SCML [Shi et al., 2014]
    - Learn K parameters
    - Hard to generate good bases in high dimensions
- Special case: sparse data [Liu et al., 2015]
    - Decomposition as rank-1 4-sparse matrices
    - Greedy algorithm incorporating a single basis at each iteration
    - Computational cost independent of d

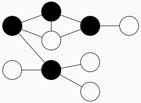
METRIC LEARNING FOR STRUCTURED DATA

MOTIVATION

- Each data instance is a structured object
  - Strings: words, DNA sequences
  - Trees: XML documents
  - Graphs: social network, molecules

ACGGCTT





- Metrics on structured data are convenient
  - Act as proxy to manipulate complex objects
  - Can use any metric-based algorithm

- Could represent each object by a feature vector
    - Idea behind many kernels for structured data
    - Could then apply standard metric learning techniques
    - Potential loss of structural information
- Instead, focus on edit distances
    - Directly operate on structured object
    - Variants for strings, trees, graphs
    - Natural parameterization by cost matrix

- Notations
  - Alphabet Σ: finite set of symbols
  - String x: finite sequence of symbols from Σ
  - $|x|$: length of string x
  - \$: empty string / symbol

Definition (Levenshtein distance)

The Levenshtein string edit distance between x and x' is the length of the shortest sequence of operations (called an *edit script*) turning x into x'. Possible operations are insertion, deletion and substitution of symbols.

- Computed in $O(|x| \cdot |x'|)$ time by Dynamic Programming (DP)

Parameterized version

- Use a nonnegative $(|\Sigma|+1)\times(|\Sigma|+1)$ matrix C
  - C_{ij}: cost of substituting symbol i with symbol j

Example 1: Levenshtein distance

C	\$	a	b
\$	0	1	1
a	1	0	1
b	1	1	0

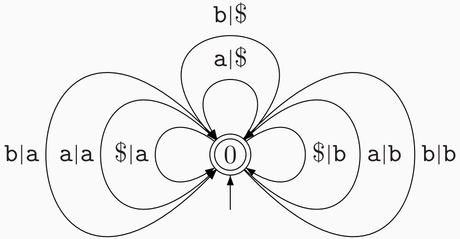
$\Longrightarrow$ edit distance between abb and aa is 2 (needs at least two operations)

Example 2: specific costs

C	\$	a	b
\$	0	2	10
a	2	0	4
b	10	4	0

$\Longrightarrow$ edit distance between abb and aa is 10 (a $\to$ \$, b $\to$ a, b $\to$ a)

- Interdependence issue
  - The optimal edit script depends on the costs
  - Updating the costs may change the optimal edit script
- Consider edit probability $p(x'|x)$ [Oncina and Sebban, 2006]
  - Cost matrix: probability distribution over operations
  - Corresponds to summing over all possible scripts
- Represent process by a stochastic memoryless transducer
- Maximize expected log-likelihood of positive pairs

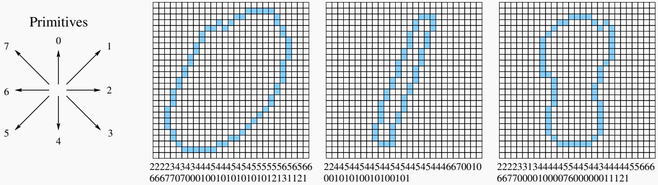


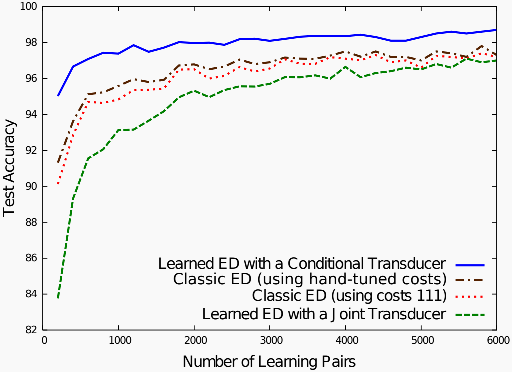
Iterative Expectation-Maximization algorithm [Oncina and Sebban, 2006]

- Expectation step
  - Given edit probabilities, compute frequency of each operation
  - Probabilistic version of the DP algorithm
- Maximization step
  - Given frequencies, update edit probabilities
  - Done by likelihood maximization under constraints

$$\forall u \in \Sigma, \sum_{v \in \Sigma \cup \{\$\}} c_{v|u} + \sum_{v \in \Sigma} c_{v|\$} = 1, \quad \text{with} \sum_{v \in \Sigma} c_{v|\$} + \underbrace{c(\#)}_{\text{exit prob.}} = 1,$$

Application to handwritten digit recognition [Oncina and Sebban, 2006]





Some remarks

- Advantages
    - Elegant probabilistic framework
    - Enables data generation
    - Generalization to trees [Bernard et al., 2008]
- Drawbacks
    - Convergence to local minimum
    - Costly: DP algorithm for each pair at each iteration
    - Cannot use negative pairs

GESL [Bellet et al., 2012a]

- Inspired from successful algorithms for non-structured data
    - Large-margin constraints
    - Convex optimization
- Requires key simplification: fix the edit script

$$e_C(\mathrm{x}, \mathrm{x}') = \sum_{u,v \in \Sigma \cup \{\$\}} C_{uv} \cdot \#_{uv}(\mathrm{x}, \mathrm{x}')$$

- $\#_{uv}(\mathrm{x}, \mathrm{x}')$: nb of times $u \to v$ appears in Levenshtein script
- e_C is a linear function of the costs

GESL [Bellet et al., 2012a]

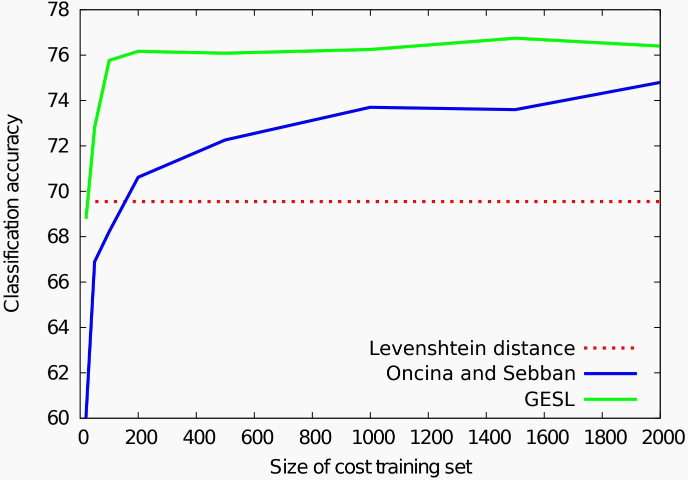
Formulation

$$\begin{aligned}
\min_{\mathbf{C}\geq 0, \boldsymbol{\xi}\geq 0, B_1\geq 0, B_2 \geq 0} \quad & \sum_{i,j} \xi_{ij} \quad + \quad \lambda \|\mathbf{C}\|^2_{\mathcal{F}} \\
\text{s.t.} \quad & e_{\mathbf{C}}(\mathrm{x}, \mathrm{x}') \geq B_1 - \xi_{ij} \qquad \forall (\mathrm{x}_i, \mathrm{x}_j) \in \mathcal{D} \\
& e_{\mathbf{C}}(\mathrm{x}, \mathrm{x}') \leq B_2 + \xi_{ij} \qquad \forall (\mathrm{x}_i, \mathrm{x}_j) \in \mathcal{S} \\
& B_1 - B_2 = \gamma
\end{aligned}$$

γ margin parameter

- Convex, less costly and use of negative pairs
- Straightforward adaptation to trees and graphs
- Less general than proper edit distance
  - Chicken and egg situation similar to LMNN

Application to word classification [Bellet et al., 2012a]



GENERALIZATION GUARANTEES

STATISTICAL VIEW OF SUPERVISED METRIC LEARNING

- Training data $T_n = \{\mathbf{z}_i = (\mathbf{x}_i, y_i)\}_{i=1}^n$
    - $\mathbf{z}_i \in \mathcal{Z} = \mathcal{X} \times \mathcal{Y}$
    - $\mathcal{Y}$ discrete label set
    - independent draws from unknown distribution μ over $\mathcal{Z}$
- Minimize the regularized empirical risk

$$R_n(M) = \frac{2}{n(n-1)} \sum_{1 \leq i < j \leq n} \ell(M, \mathbf{z}_i, \mathbf{z}_j) \quad + \quad \lambda reg(M)$$

- Hope to achieve small expected risk

$$R(M) = \mathop{\mathbb{E}}_{\mathbf{z},\mathbf{z}' \sim \mu} [\ell(M, \mathbf{z}, \mathbf{z}')]$$

- Note: this can be adapted to triplets

- Standard statistical learning theory: sum of i.i.d. terms
- Here $R_n(\boldsymbol{M})$ is a sum of dependent terms!
    - Each training point involved in several pairs
    - Corresponds to practical situation
- Need specific tools to go around this problem
    - Uniform stability
    - Algorithmic robustness

Definition ([Jin et al., 2009])

A metric learning algorithm has a uniform stability in κ/n, where κ is a positive constant, if

$$\forall(T_n, \boldsymbol{z}), \forall i, \quad \sup_{z_1,z_2} |\ell(M_{T_n}, \boldsymbol{z}_1, \boldsymbol{z}_2) - \ell(M_{T_n^{i,z}}, \boldsymbol{z}_1, \boldsymbol{z}_2)| \leq \frac{\kappa}{n}$$

- M_{T_n}: metric learned from T_n
- $T_n^{i,\boldsymbol{z}}$: set obtained by replacing $\boldsymbol{z}_i \in T_n$ by $\boldsymbol{z}$

- If $reg(M) = \|M\|_{\mathcal{F}}^2$, under mild conditions on ℓ, algorithm has uniform stability [Jin et al., 2009]
  - Applies for instance to GESL [Bellet et al., 2012a]
- Does not apply to other (sparse) regularizers

Generalization bound

Theorem ([Jin et al., 2009])

For any metric learning algorithm with uniform stability κ/n, with probability $1-\delta$ over the random sample T_n, we have:

$$R(\boldsymbol{M}_{T_n}) \leq R_n(\boldsymbol{M}_{T_n}) + \frac{2\kappa}{n} + (2\kappa + B)\sqrt{\frac{\ln(2/\delta)}{2n}}$$

B problem-dependent constant

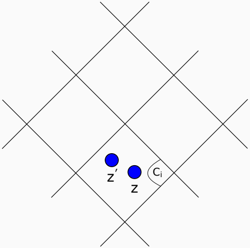
- Standard bound in $O(1/\sqrt{n})$

Definition ([Bellet and Habrard, 2015])

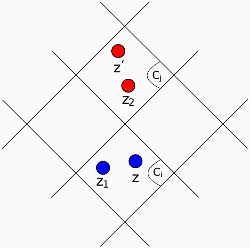
A metric learning algorithm is $(K, \epsilon(\cdot))$ robust for $K \in \mathbb{N}$ and $\epsilon : (\mathcal{Z} \times \mathcal{Z})^n \rightarrow \mathbb{R}$ if $\mathcal{Z}$ can be partitioned into K disjoints sets, denoted by $\{C_i\}_{i=1}^K$, such that the following holds for all T_n:

$$\forall (\boldsymbol{z}_1, \boldsymbol{z}_2) \in T_n^2, \forall \boldsymbol{z}, \boldsymbol{z}' \in \mathcal{Z}, \forall i, j \in [K], \text{ if } \boldsymbol{z}_1, \boldsymbol{z} \in C_i, \boldsymbol{z}_2, \boldsymbol{z}' \in C_j$$

$$|\ell(\boldsymbol{M}_{T_n}, \boldsymbol{z}_1, \boldsymbol{z}_2) - \ell(\boldsymbol{M}_{T_n}, \boldsymbol{z}, \boldsymbol{z}')| \leq \epsilon(T_n^2)$$



Classic robustness



Robustness for metric learning

Generalization bound

Theorem ([Bellet and Habrard, 2015])

If a metric learning algorithm is $(K, \epsilon(\cdot))$-robust, then for any $\delta > 0$, with probability at least $1 - \delta$ we have:

$$R(\boldsymbol{M}_{\mathcal{T}_n}) \leq R_n(\boldsymbol{M}_{\mathcal{T}_n}) + \epsilon(\mathcal{T}_n^2) + 2B\sqrt{\frac{2K \ln 2 + 2\ln(1/\delta)}{n}}$$

- Wide applicability
  - Mild assumptions on ℓ
  - Any norm regularizer: Frobenius, $L_{2,1}$, trace...
- Bounds are loose
  - $\epsilon(\mathcal{T}_n^2)$ can be as small as needed by increasing K
  - But K potentially very large and hard to estimate

- [Cao et al., 2012]
    - Relies on Rademacher complexity
    - Tight bounds for several matrix norms
- [Clémençon et al., 2015]
    - Approximation of empirical risk by sampling $O(n)$ pairs
    - Minimization of this incomplete risk preserves $O(1/\sqrt{n})$ rate
- [Bellet et al., 2012b]
    - Similarity learning for linear classification
    - Generalization bounds for classifier based on learned similarity
    - Builds upon theory developed in [Balcan and Blum, 2006]

- Short book published in 2015

  A. Bellet, A. Habrard and M. Sebban
  Metric Learning
  Morgan & Claypool Publishers

- Also see arXiv survey (last update in 2014, new update soon)

  A. Bellet, A. Habrard and M. Sebban
  A Survey on Metric Learning for Feature Vectors and Structured Data
  Technical report, arXiv:1306.6709

SUMMARY OF THE FIRST PART

- Good level of maturity
  - Various types of metrics
  - Many learning scenarios
  - Scalability
  - Theory
  - Code available for many methods
- Structured data not explored much
  - Lagging behind in many respects
  - Hardness of combinatorial problems
  - Taking structure into account is key

Questions?

[Bach et al., 2012] Bach, F. R., Jenatton, R., Mairal, J., and Obozinski, G. (2012).
Optimization with Sparsity-Inducing Penalties.
Foundations and Trends in Machine Learning, 4(1):1–106.

[Balcan and Blum, 2006] Balcan, M.-F. and Blum, A. (2006).
On a Theory of Learning with Similarity Functions.
In ICML, pages 73–80.

[Bellet and Habrard, 2015] Bellet, A. and Habrard, A. (2015).
Robustness and Generalization for Metric Learning.
Neurocomputing, 151(1):259–267.

[Bellet et al., 2012a] Bellet, A., Habrard, A., and Sebban, M. (2012a).
Good edit similarity learning by loss minimization.
Machine Learning Journal, 89(1):5–35.

[Bellet et al., 2012b] Bellet, A., Habrard, A., and Sebban, M. (2012b).
Similarity Learning for Provably Accurate Sparse Linear Classification.
In ICML, pages 1871–1878.

[Bernard et al., 2008] Bernard, M., Boyer, L., Habrard, A., and Sebban, M. (2008).
Learning probabilistic models of tree edit distance.
Pattern Recognition, 41(8):2611–2629.

[Cao et al., 2012] Cao, Q., Guo, Z.-C., and Ying, Y. (2012).
Generalization Bounds for Metric and Similarity Learning.
Technical report, University of Exeter.

[Chatpatanasiri et al., 2010] Chatpatanasiri, R., Korsrilabutr, T., Tangchanachaianan, P., and Kijsirikul, B. (2010).
A new kernelization framework for Mahalanobis distance learning algorithms.
Neurocomputing, 73:1570–1579.

[Chechik et al., 2010] Chechik, G., Sharma, V., Shalit, U., and Bengio, S. (2010).
Large Scale Online Learning of Image Similarity Through Ranking.
Journal of Machine Learning Research, 11:1109–1135.

[Cheng, 2013] Cheng, L. (2013).
Riemannian Similarity Learning.
In ICML.

[Chopra et al., 2005] Chopra, S., Hadsell, R., and LeCun, Y. (2005).
Learning a Similarity Metric Discriminatively, with Application to Face Verification.
In CVPR, pages 539–546.

[Clémençon et al., 2015] Clémençon, S., Bellet, A., and Colin, I. (2015).
Scaling-up Empirical Risk Minimization: Optimization of Incomplete U-statistics.
Technical report, arXiv:1501.02629.

[Davis et al., 2007] Davis, J. V., Kulis, B., Jain, P., Sra, S., and Dhillon, I. S. (2007).
Information-theoretic metric learning.
In ICML, pages 209–216.

[Geng et al., 2011] Geng, B., Tao, D., and Xu, C. (2011).
DAML: Domain Adaptation Metric Learning.
IEEE Transactions on Image Processing, 20(10):2980–2989.

[Goldberger et al., 2004] Goldberger, J., Roweis, S., Hinton, G., and Salakhutdinov, R. (2004).
Neighbourhood Components Analysis.
In NIPS, pages 513–520.

[Goldstone et al., 1997] Goldstone, R. L., Medin, D. L., and Halberstadt, J. (1997).
Similarity in context.
Memory & Cognition, 25(2):237–255.

[Hoi et al., 2008] Hoi, S. C., Liu, W., and Chang, S.-F. (2008).
Semi-supervised distance metric learning for Collaborative Image Retrieval.
In CVPR.

[Hu et al., 2014] Hu, J., Lu, J., and Tan, Y.-P. (2014).
Discriminative Deep Metric Learning for Face Verification in the Wild.
In CVPR, pages 1875–1882.

[Jaggi, 2013] Jaggi, M. (2013).
Revisiting Frank-Wolfe: Projection-Free Sparse Convex Optimization.
In ICML.

[Jin et al., 2009] Jin, R., Wang, S., and Zhou, Y. (2009).
Regularized Distance Metric Learning: Theory and Algorithm.
In NIPS, pages 862–870.

[Kedem et al., 2012] Kedem, D., Tyree, S., Weinberger, K., Sha, F., and Lanckriet, G. (2012).
Non-linear Metric Learning.
In NIPS, pages 2582–2590.

[Kulis et al., 2011] Kulis, B., Saenko, K., and Darrell, T. (2011).
What you saw is not what you get: Domain adaptation using asymmetric kernel transforms.
In CVPR, pages 1785–1792.

[Lim and Lanckriet, 2014] Lim, D. and Lanckriet, G. R. (2014).
Efficient Learning of Mahalanobis Metrics for Ranking.
In ICML, pages 1980–1988.

[Liu et al., 2015] Liu, K., Bellet, A., and Sha, F. (2015).
Similarity Learning for High-Dimensional Sparse Data.
In AISTATS, pages 653–662.

[McFee and Lanckriet, 2010] McFee, B. and Lanckriet, G. R. G. (2010).
Metric Learning to Rank.
In ICML, pages 775–782.

[Nosofsky, 1986] Nosofsky, R. M. (1986).
Attention, similarity, and the identification–categorization relationship.
Journal of Experimental Psychology: General, 115(1):39–57.

[Oncina and Sebban, 2006] Oncina, J. and Sebban, M. (2006).
Learning Stochastic Edit Distance: application in handwritten character recognition.
Pattern Recognition, 39(9):1575–1587.

[Parameswaran and Weinberger, 2010] Parameswaran, S. and Weinberger, K. Q. (2010).
Large Margin Multi-Task Metric Learning.
In NIPS, pages 1867–1875.

[Scheirer et al., 2014] Scheirer, W. J., Wilber, M. J., Eckmann, M., and Boult, T. E. (2014).
Good recognition is non-metric.
Pattern Recognition, 47(8):2721–2731.

[Schultz and Joachims, 2003] Schultz, M. and Joachims, T. (2003).
Learning a Distance Metric from Relative Comparisons.
In NIPS.

[Shi et al., 2014] Shi, Y., Bellet, A., and Sha, F. (2014).
Sparse Compositional Metric Learning.
In AAAI, pages 2078–2084.

[Tversky, 1977] Tversky, A. (1977).
Features of similarity.
Psychological Review, 84(4):327–352.

[Tversky and Gati, 1982] Tversky, A. and Gati, I. (1982).
Similarity, separability, and the triangle inequality.
Psychological Review, 89(2):123–154.

[van der Maaten and Hinton, 2008] van der Maaten, L. and Hinton, G. (2008).
Visualizing Data using t-SNE.
Journal of Machine Learning Research, 9:2579–2605.

[Weinberger et al., 2005] Weinberger, K. Q., Blitzer, J., and Saul, L. K. (2005).
Distance Metric Learning for Large Margin Nearest Neighbor Classification.
In NIPS, pages 1473–1480.

[Weinberger and Saul, 2009] Weinberger, K. Q. and Saul, L. K. (2009).
Distance Metric Learning for Large Margin Nearest Neighbor Classification.
Journal of Machine Learning Research, 10:207–244.

[Xie and Xing, 2014] Xie, P. and Xing, E. (2014).
Large Scale Distributed Distance Metric Learning.
Technical report, arXiv:1412.5949.

[Xing et al., 2002] Xing, E. P., Ng, A. Y., Jordan, M. I., and Russell, S. J. (2002).
Distance Metric Learning with Application to Clustering with Side-Information.
In NIPS, pages 505–512.

[Ying et al., 2009] Ying, Y., Huang, K., and Campbell, C. (2009).
Sparse Metric Learning via Smooth Optimization.
In NIPS, pages 2214–2222.

[Zhang and Yeung, 2010] Zhang, Y. and Yeung, D.-Y. (2010).
Transfer metric learning by learning task relationships.
In KDD, pages 1199–1208.

ECML/PKDD
Porto, September 7, 2015

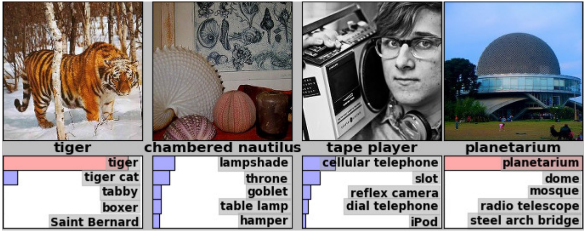
Similarity and Distance Metric Learning with Applications to Computer Vision
Part II

Matthieu Cord

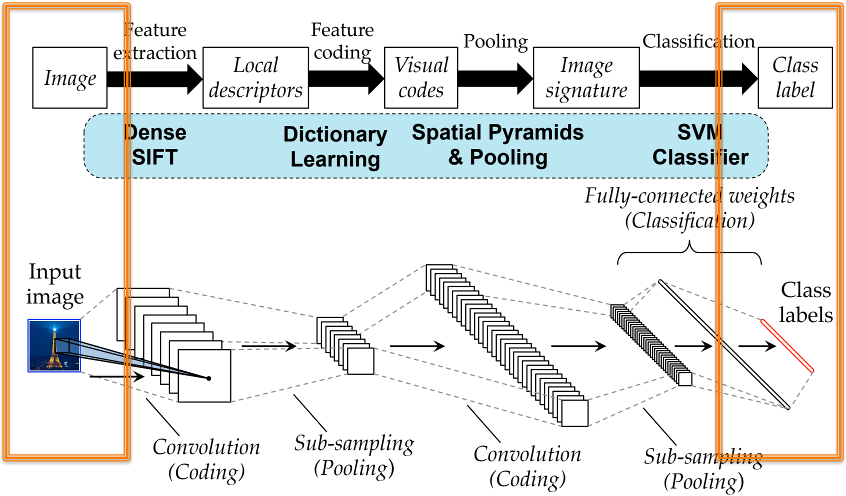
LIP6 - Computer Science Department
UPMC PARIS 6 - Sorbonne University

Introduction: Visual learning

- A lot of recent successful applications of Machine Learning to Visual Understanding
- Supervised classification on large dataset ImageNet [winner 2012]
  - 1M images
  - 1000 classes

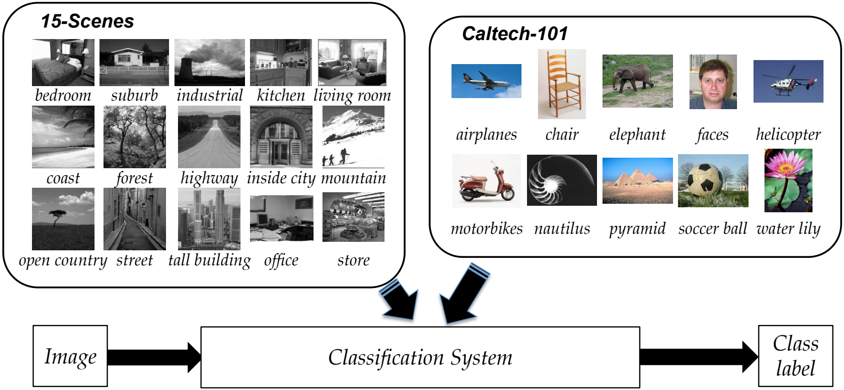


Introduction: Visual learning



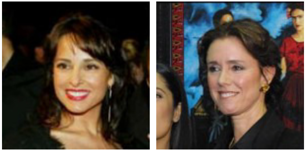
Introduction: Visual learning

- Data for training



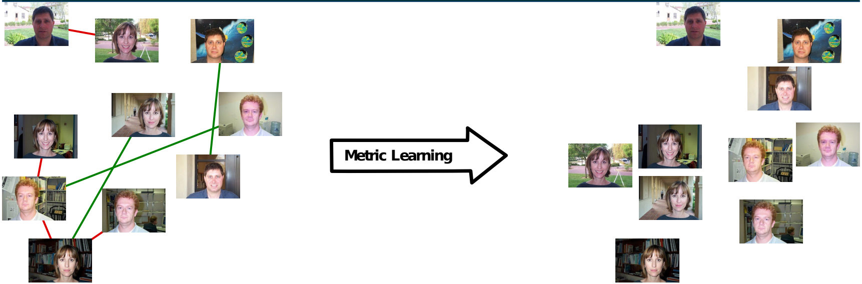
Introduction: Visual learning

- Beyond classification image+label
- Data for training : image pairs, triplets, …
  - Pairs+label YES/NO (LFW)



  - Class information

Introduction: Metric learning for CV



Metrics in Machine Learning and Computer Vision

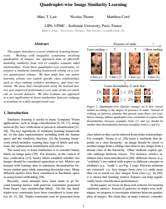
- Image dataset Clustering
- Information/Image retrieval
- kNN classification, Kernel methods

Commonly used metrics: Euclidean distance, chi2 for histograms, …

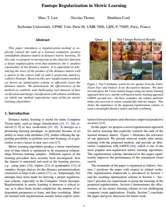
Outline of part II

1. Introduction
2. **Metric Learning in CV**
   - Data and Metric models
   - Learning schemes
   - Results
3. Computer Vision Applications
   - Relative attribute learning
   - Web page comparison

ICCV 2013


Quadruplet-wise Image Similarity Learning
Marc T. Law Nicolas Thome Matthieu Cord
LIP6, UPMC - Sorbonne University, Paris, France

CVPR 2014

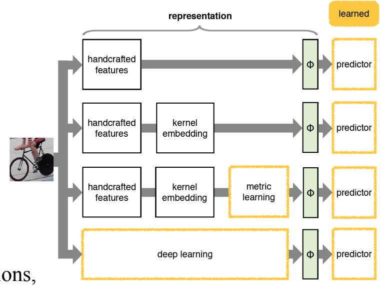

Fantope Regularization in Metric Learning
Marc T. Law Nicolas Thome Matthieu Cord
Sorbonne Universités, UPMC Univ Paris 06, UMR 7606, LIP6, F-75005, Paris, France

Metric Learning in CV

- Key ingredients of metric/similarity learning in CV:
  - Data representation including both:
    - Feature space
      - » Bag of visual word representation (BoW)
      - » Deep features, Gist …

      IMAGE REPRESENTATION ➔ VECTOR

    - Similarity function / Metric
  - Learning framework
    - training data, type of labels and relations,
    - Optimization formulation
    - Solvers



Credit: A. Vedaldi

Metric Learning in CV

- Similarity function / Metric:

Vector representations $\mathbf{x} \in \mathbb{R}^d$ (visual BoWs, deep, ...)

Widely used approach: **Mahalanobis-like Distance Metric Learning**

$$\mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^d, \mathbf{M} \in \mathbb{S}_+^d,\ D^2_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i - \mathbf{x}_j)^\top \mathbf{M} (\mathbf{x}_i - \mathbf{x}_j) \tag{1}$$

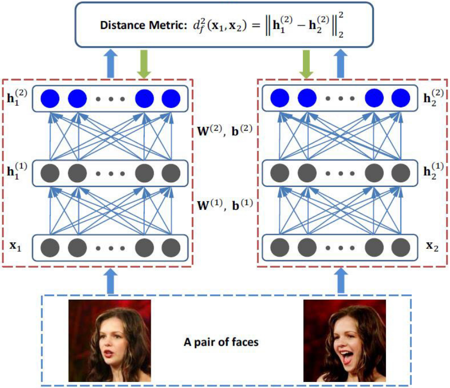
Since for all $\mathbf{M} \in \mathbb{S}_+^d$ with $\mathrm{rank}(\mathbf{M}) = e \leq d$, there exists $\mathbf{L} \in \mathbb{R}^{e \times d}$ such that $\mathbf{M} = \mathbf{L}^\top \mathbf{L}$:

$$\begin{aligned}\mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^d, \mathbf{M} \in \mathbb{S}_+^d,\ D^2_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j) &= (\mathbf{x}_i - \mathbf{x}_j)^\top \mathbf{L}^\top \mathbf{L} (\mathbf{x}_i - \mathbf{x}_j) \\ &= \|\mathbf{L}\mathbf{x}_i - \mathbf{L}\mathbf{x}_j\|_2^2\end{aligned} \tag{2}$$

- All M (or L) coefficients to be learned

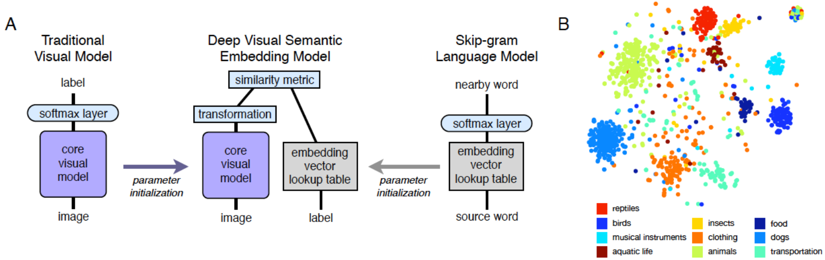
Metric Learning in CV

Non-linear extension: kernel vs deep [credit: Hu CVPR14]



Metric Learning in CV

- One step further: heterogeneous object deep embedding and metric learning



DeVISE system [google, NIPS 2013]

Outline

1. Introduction
2. **Metric Learning in CV**
   - Data and Metric models
   - **Learning schemes:**
     - **Constraints: Pairs, triplets …**
     - Objective function: regularization, optimization …
   - Results
3. Computer Vision Applications

Metric Learning in CV

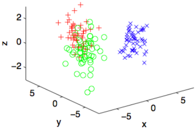
- PairWise Constraints for learning

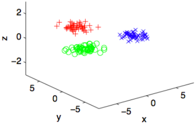
Metric Learning in CV

- Learning scheme for pairwise constraints:

  Xing et al: *Distance metric learning, ..., NIPS 2002* (cf. Part I)

$$\min_{\mathbf{M}\in\mathbb{S}_+^d} \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{S}} D^2_{\mathbf{M}}(\mathbf{x}_i,\mathbf{x}_j) \quad s.t. \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{D}} \sqrt{D^2_{\mathbf{M}}(\mathbf{x}_i,\mathbf{x}_j)} \geq 1$$



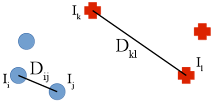


Metric Learning in CV

- What are the pairs in S and D ? All consistent ?

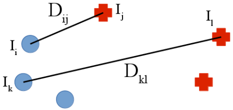
S: ok

very far :



- Mono-modality as underlying hypothesis

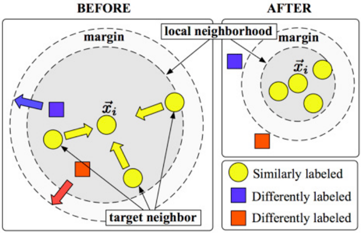
D: sometimes not far

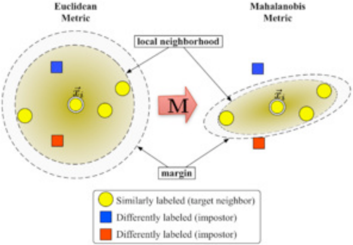


=> Important trick: getting training pairs using neighbor selection

Metric Learning in CV

- Triplet constraints for learning:
- The most used scheme: [Weinberger LMNN] (cf. Part I)





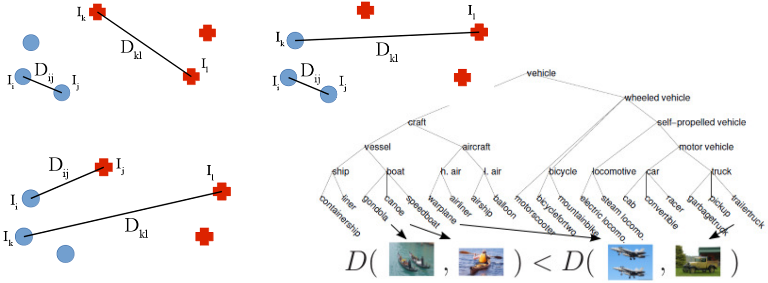
$$\min_{\mathbf{M}\in\mathbb{S}_+^d} \sum_{(\mathbf{x}_i,\mathbf{x}_i^+)\in\mathcal{S}} D^2_{\mathbf{M}}(\mathbf{x}_i,\mathbf{x}_i^+)$$

$$\text{s.t.} \quad \forall(\mathbf{x}_i,\mathbf{x}_i^+,\mathbf{x}_i^-)\in\mathcal{T}, D^2_{\mathbf{M}}(\mathbf{x}_i,\mathbf{x}_i^-) \geq \delta + D^2_{\mathbf{M}}(\mathbf{x}_i,\mathbf{x}_i^+)$$

Metric Learning in CV

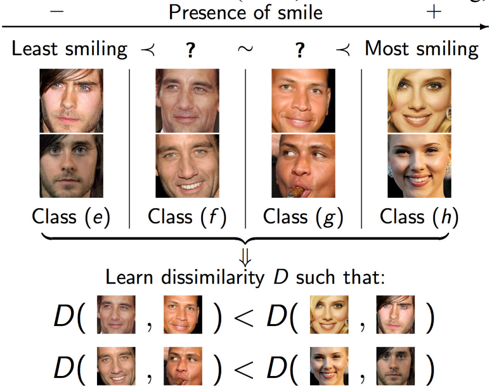
- Quadruplet-Wise constraints: [Law, Thome, Cord ICCV 2013]
  - Generalizing pairs-wise (and triplets), more flexible and expressive
  - Margin-based strategy, not always selecting all constraints

$$\forall q = (\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k, \mathbf{x}_l) \in \mathcal{N}, \; D^2(\mathbf{x}_i, \mathbf{x}_j) + \delta_q \leq D^2(\mathbf{x}_k, \mathbf{x}_l)$$



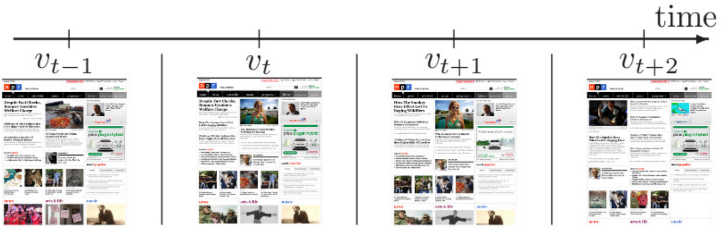
Metric Learning in CV

- Application 1: learning relative attributes
  - Supervision based on attributes (smile, masculine looking, …)



Web page/temporal info for ML

- Application 2:
  - Fully unsupervised ML, but temporal information available
  - Constraints by comparing screenshots of successive webpage versions



$$D(v_t, v_{t+1}) < D(v_{t-1}, v_{t+2})$$

Outline

1. Introduction
2. **Metric Learning in CV**
   - Data and Metric models
   - **Learning schemes:**
     - Constraints: Pairs, triplets …
     - **Objective function: regularization, optimization …**
   - Results
3. Computer Vision Applications
   - Relative attribute learning
   - Web page comparison

Metric Learning in CV

To summarize constraints with $D^2_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i - \mathbf{x}_j)^\top \mathbf{M}(\mathbf{x}_i - \mathbf{x}_j)$:

- **Pairs:**
  $\mathcal{N} = \mathcal{S} \cup \mathcal{D} \Longrightarrow \begin{cases} \forall(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{S} & D^2_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j) < 1 \\ \forall(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{D} & D^2_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j) > 1 \end{cases}$

- **Triplets:**
  $\mathcal{N} = \{(\mathbf{x}_i, \mathbf{x}_i^+, \mathbf{x}_i^-)\}_{i=1}^N \Longrightarrow \forall(\mathbf{x}_i, \mathbf{x}_i^+, \mathbf{x}_i^-) \in \mathcal{N},\ D^2_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_i^+) + \delta \leq D^2_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_i^-)$

- **Quadruplets:**
  $\mathcal{N} = \{q = (\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k, \mathbf{x}_l)\} \Longrightarrow \forall q \in \mathcal{N},\ D^2_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j) + \delta_q \leq D^2_{\mathbf{M}}(\mathbf{x}_k, \mathbf{x}_l)$

Optimization scheme:

$$\min_{\mathbf{M} \in \mathbb{S}^d_+} \mu R(\mathbf{M}) + \ell(\mathbf{M}, \mathcal{N})$$

With $R(\mathbf{M})$: regularizer and $\ell(\mathbf{M}, \mathcal{N})$ loss over set of constraints $\mathcal{N}$

Metric Learning in CV

(Large margin) **optimization**:

- Qwise optimization framework with hinge loss function

$$\min_{\mathbf{M}\in\mathbb{S}_+^d} \mu R(\mathbf{M}) + \sum_{q\in\mathcal{N}} \xi_q$$

$$\text{s.t. } \forall q = (\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k, \mathbf{x}_l) \in \mathcal{N}, D^2_{\mathbf{M}}(\mathbf{x}_k, \mathbf{x}_l) \geq D^2_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j) + \delta_q - \xi_q$$

$$\forall q \in \mathcal{N}, \xi_q \geq 0$$

  - $R(\mathbf{M})$: regularization term
  - μ : trade-off between fitting and regularization.

- Triplet optim:

$$\min_{\mathbf{M}\in\mathbb{S}_+^d} \sum_{(\mathbf{x}_i,\mathbf{x}_i^+)\in\mathcal{S}} D^2_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_i^+) + \sum_{(\mathbf{x}_i,\mathbf{x}_i^+,\mathbf{x}_i^-)\in\mathcal{T}} \xi_i$$

$$\text{s.t. } \forall(\mathbf{x}_i, \mathbf{x}_i^+, \mathbf{x}_i^-) \in \mathcal{T}, D^2_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_i^-) \geq 1 + D^2_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_i^+) - \xi_i$$

Metric Learning in CV

- How to define/choose the regularization R(M) in the objective function:

$$\min_{\mathbf{M}\in\mathbb{S}_{+}^{d}} \mu R(\mathbf{M}) + \ell(\mathbf{M}, \mathcal{N})$$

- Regularization term to express *prior*, to control complexity …

$$D_{\mathbf{M}}^{2}(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i - \mathbf{x}_j)^{\top}\mathbf{M}(\mathbf{x}_i - \mathbf{x}_j)$$

- For CV application, looking for Low rank solution:
  - Controlling overfitting
  - Sparsity of the singular values
  - Exploiting correlation between features
  - Fast/efficient solution

Metric Learning in CV

Formulation of $R(\mathbf{M})$ $\qquad D^2_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i - \mathbf{x}_j)^\top \mathbf{M}(\mathbf{x}_i - \mathbf{x}_j)$

- Frobenius norm $R(\mathbf{M}) = \|\mathbf{M}\|_F^2 = \sum M_{ij}^2$
  - does not promote low-rank solutions
  - useful when $\mathbf{M}$ is a diagonal matrix
- log det divergence: $D_{\ell d}(\mathbf{M}, \mathbf{M}_0) = \mathrm{tr}(\mathbf{M}\mathbf{M}_0^{-1}) - \log\det(\mathbf{M}\mathbf{M}_0^{-1}) - d$
- Sum of distances between similar examples (xing, LMNN)
- Nuclear norm regularization $R(\mathbf{M}) = \|\mathbf{M}\|_* = \mathrm{tr}(\mathbf{M})$:
  - rank NP-hard to optimize
  - convex envelope of rank($\mathbf{M}$) on the set $\{\mathbf{M} \in \mathbb{R}^{d\times d} : \|\mathbf{M}\| \leq 1\}$
  - ℓ_1 norm of vector of singular values $\sigma(\mathbf{M})$

Metric Learning in CV

- Fantope regularization [Law, Thome, Cord CVPR 2014]:

  - Explicit control of the rank of M

    By noting, $\forall \mathbf{M} \in \mathbb{S}_+^d, R(\mathbf{M})$: sum of the k smallest eigenvalues of $\mathbf{M}$

    $$R(\mathbf{M}) = 0 \iff \mathrm{rank}(\mathbf{M}) \leq d - k$$

  - Reformulation

    $$\min_{\mathbf{M} \in \mathbb{S}_+^d} \mu R(\mathbf{M}) + \ell(\mathbf{M}, \mathcal{N}) \Longrightarrow \min_{\mathbf{M} \in \mathbb{S}_+^d} \mu \langle \mathbf{W}, \mathbf{M} \rangle + \ell(\mathbf{M}, \mathcal{N})$$

with $\mathbf{W}$ rank-k projector on the eigenvectors of $\mathbf{M}$ with k smallest eigenvalues

Metric Learning in CV

Construction of W

- $\mathbf{M} = \mathbf{V_M}\mathrm{Diag}(\lambda(\mathbf{M}))\mathbf{V_M}^\top$ eigendecomposition of $\mathbf{M} \in \mathbb{S}_+^d$, $\mathbf{V_M}$ orthogonal matrix
- We construct $\mathbf{w} = (w_1, \ldots, w_d)^\top \in \mathbb{R}^d$:

$$w_i = \begin{cases} 0 \text{ if } 1 \leq i \leq d-k \text{ (the first } d-k \text{ elements)} \\ 1 \text{ if } d-k+1 \leq i \leq d \text{ (the last } k \text{ elements)} \end{cases}$$

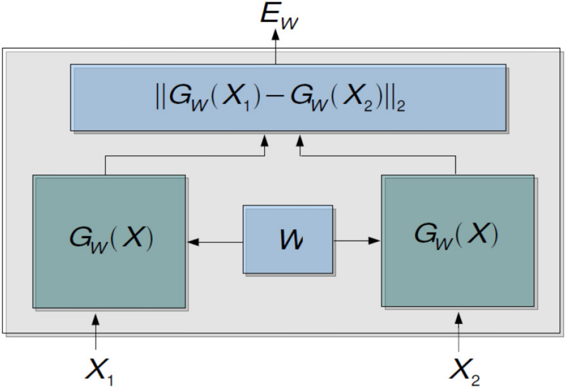
$$\mathbf{W} = \mathbf{V_M}\mathrm{Diag}(\mathbf{w})\mathbf{V_M}^\top \tag{1}$$

$$\min_{\mathbf{M}\in\mathbb{S}_+^d} \mu R(\mathbf{M}) + \ell(\mathbf{M}, \mathcal{N}) \Longrightarrow \min_{\mathbf{M}\in\mathbb{S}_+^d} \mu\langle \mathbf{W}, \mathbf{M}\rangle + \ell(\mathbf{M}, \mathcal{N}) \text{ s.t. } \mathbf{W} = \mathbf{V_M}\mathrm{Diag}(\mathbf{w})\mathbf{V_M}^\top$$

Algorithm: alternating optimization procedure

Metric Learning in CV

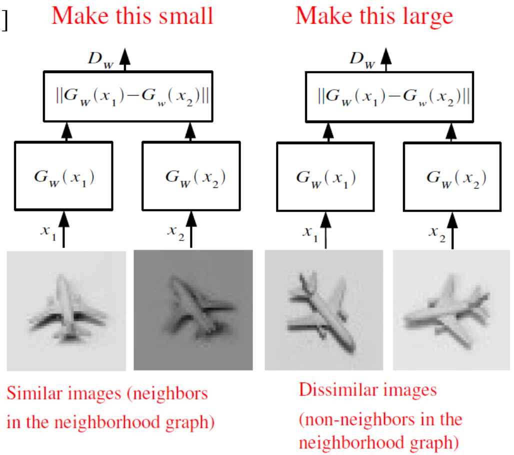
- Deep metric learning optimization
  - Siamese Architecture [LeCun NIPS 1993]



Metric Learning in CV

- Deep metric learning optimization

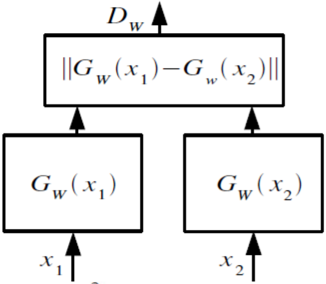
[credit: Y. LeCun 05]



Metric Learning in CV

- Deep metric learning optimization

[Y. LeCun CVPR 05,06] DrLIM scheme



Similar to LMNN procedure:

Y=0 for similar pairs

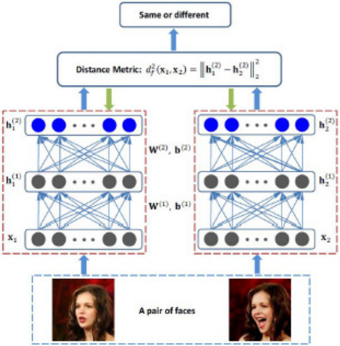
Y=1 for dissimilar pairs

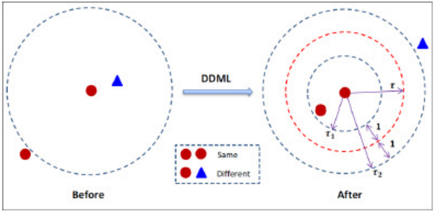
The exact loss function is

$$L(W, Y, \vec{X_1}, \vec{X_2}) = (1-Y)\frac{1}{2}(D_W)^2 + (Y)\frac{1}{2}\{max(0, m - D_W)\}^2$$

Metric Learning in CV

- Siamese Network for paiwise comparison: DDML approach
  [Credit: Hu CVPR 2014]





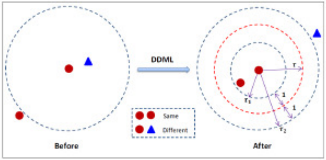
Intuitive illustration of the proposed DDML method

Metric Learning in CV

- DDML optimization [Hu CVPR 2014]:

$$d_f^2(x_i, x_j) < \tau - 1, l_{ij} = 1$$
$$d_f^2(x_i, x_j) > \tau + 1, l_{ij} = -1$$

$$\ell_{ij}\left(\tau - d_f^2(\mathbf{x}_i, \mathbf{x}_j)\right) > 1$$

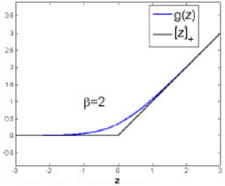


Intuitive illustration of the proposed DDML method

DDML as the following optimization problem:

$$\begin{aligned}\arg\min_f J &= J_1 + J_2 \\ &= \frac{1}{2}\sum_{i,j} g\Big(1 - \ell_{ij}\big(\tau - d_f^2(\mathbf{x}_i, \mathbf{x}_j)\big)\Big) \\ &+ \frac{\lambda}{2}\sum_{m=1}^{M}\left(\|\mathbf{W}^{(m)}\|_F^2 + \|\mathbf{b}^{(m)}\|_2^2\right)\end{aligned}$$

where $g(z) = \frac{1}{\beta}\log\left(1 + \exp(\beta z)\right)$ is the generalized logistic loss function [25], which is a smoothed approximation of the hinge loss function $[z]_+ = \max(z, 0)$



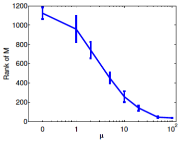
Outline

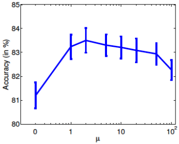
1. Introduction
2. **Metric Learning in CV**
   - Data and Metric models
   - Learning schemes:
   - **Results**
3. Computer Vision Applications

Results on face verification pb

2 images => same face ?
Labeled Faces in the Wild (LFW)-- 27 SIFT descriptors concatenated
10-fold Cross Validation (600 pairs per fold)

Method	Accuracy (in %)
ITML	76.2 ± 0.5
LDML	77.5 ± 0.5
PCCA	82.2 ± 0.4
Fantope	$\mathbf{83.5 \pm 0.5}$



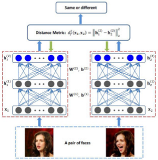


About 15% better with metric learning

Classical errors :

Results on face verification pb

Performances of deep DDML on LFW (more features): 90.68%



Recent extensions of deep archi (extra data, diff protocol):

Method	Accuracy (%)	No. of points	No. of images	Feature dimension
Joint Bayesian [8]	92.42 (o)	5	99,773	2000 × 4
ConvNet-RBM [31]	92.52 (o)	3	87,628	N/A
CMD+SLBP [17]	92.58 (u)	3	N/A	2302
Fisher vector faces [29]	93.03 (u)	9	N/A	128 × 2
Tom-vs-Pete classifiers [2]	93.30 (o+r)	95	20,639	5000
High-dim LBP [9]	95.17 (o)	27	99,773	2000
TL Joint Bayesian [6]	96.33 (o+u)	27	99,773	2000
DeepFace [32]	97.25 (o+u)	6 + 67	4,400,000 + 3,000,000	4096 × 4
DeepID on CelebFaces	**96.05** (o)	5	87,628	150
DeepID on CelebFaces+	**97.20** (o)	5	202,599	150
DeepID on CelebFaces+ & TL	**97.45** (o+u)	5	202,599	150

Results on face verification pb

DeepID2:

- Extension of classification and metric learning for LFW [Sun NIPS 2014]
- Deep learning face representation by joint Identification-Verification
- Score on LFW: 99.15%

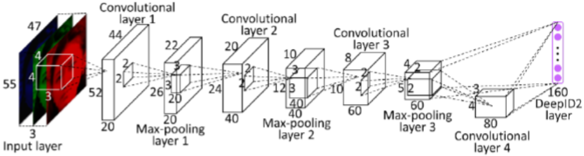
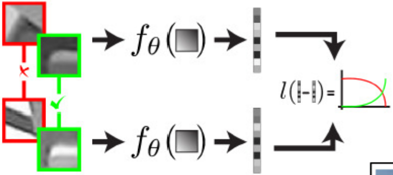


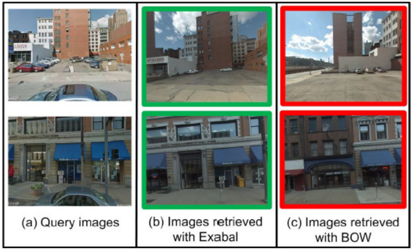
Figure 1: The ConvNet structure for DeepID2 extraction.

Other appli:
People verification

Results: feature learning

Robotics applis:
[Carlevaris-Bianco IROS 2014] from DrLIM scheme





Metric Learning for Geo-localization:
[LeBarz ICIP 2015] from LMNN scheme

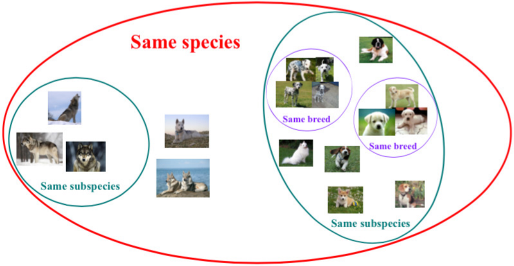
=> Many different contexts provide training data

Results: Hierarchical Classification

Rich relationships in taxonomies can be described with relative distances

Information richer that “is similar” or “is dissimilar”

Different levels of similarity



Learn dissimilarity D such that:

Taxonomy ML

- Qwise constraints sampling:
    1. Images in the same class more similar than images in sibling classes
    2. Images in sibling classes more similar than images in cousin classes
- $\mathbf{x}_i \in \mathbb{R}^d$: 1,000 dimensional SIFT BoW descriptor
- Diagonal PSD matrix framework: $\mathbf{w} \geq 0$
- **Convex Optimization Problem**:

$$\min_{\mathbf{w}} \mu \|\mathbf{w}\|_2^2 + \sum_{(p_i, p_j, p_k, p_l)} \ell(\mathbf{w}^\top [\Psi(p_k, p_l) - \Psi(p_i, p_j)])$$

with $\Psi(p_i, p_j) = (\mathbf{x}_i - \mathbf{x}_j) \circ (\mathbf{x}_i - \mathbf{x}_j)$ Hadamard product

Taxonomy ML

Subtree Dataset	[Verma 2012]	Qwise
Amphibian	41%	**43.5**%
Fish	39%	**41**%
Fruit	**23.5**%	21.1%
Furniture	46%	**48.8**%
Geological Formation	52.5%	**56.1**%
Musical Instrument	32.5%	**32.9**%
Reptile	22%	**23.0**%
Tool	**29.5**%	26.4%
Vehicle	27%	**34.7**%
Global Accuracy	34.8%	**36.4**%

Table 1: Standard classification accuracy for the various datasets.

- **9 datasets** from ImageNet, for each dataset: from 8 to 40 different classes, from 8,000 to 54,000 images for training

Outline

1. Introduction
2. Metric Learning
3. **Computer Vision Applications**
   - Relative attribute learning
   - Web page comparison

CV app: Scarlett and others

- Best Paper (Marr Prize) at ICCV 2011:

  Relative attributes,

  D. Parikh (TTI Chicago) and

  K. Grauman (Texas Univ)

To appear, Proceedings of the International Conference on Computer Vision (ICCV), 2011.

Relative Attributes

Devi Parikh
Toyota Technological Institute Chicago (TTIC)
dparikh@ttic.edu

Kristen Grauman
University of Texas at Austin
grauman@cs.utexas.edu

Abstract

Human-nameable visual "attributes" can benefit various recognition tasks. However, existing techniques restrict these properties to categorical labels (for example, a person is 'smiling' or not, a scene is 'dry' or not), and thus fail to capture more general semantic relationships. We propose to model relative attributes. *Given training data stating how object/scene categories relate according to different attributes, we learn a ranking function per attribute. The learned ranking functions predict the relative strength of each property in novel images. We then build a generative model over the joint space of attribute ranking outputs, and propose a novel form of zero-shot learning in which the supervisor* relates *the unseen object category to previously seen objects via attributes (for example, 'bears are furrier than giraffes'). We further show how the proposed relative attributes enable richer textual descriptions for new images, which in practice are more precise for human interpretation. We demonstrate the approach on datasets of faces and natural scenes, and show its clear advantages over traditional binary attribute prediction for these new tasks.*

(a) Smiling (b) ? (c) Not smiling
(d) Natural (e) ? (f) Manmade

Figure 1. Binary attributes are an artificially restrictive way to describe images. While it is clear that (a) is smiling, and (c) is not, the more informative and intuitive description for (b) is via *relative* attributes: he is smiling more than (a) but less than (c). Similarly, scene (e) is less natural than (d), but more so than (f). Our main idea is to model relative attributes via learned ranking functions, and then demonstrate their impact on novel forms of zero-shot learning and generating image descriptions.

1. Introduction

While traditional visual recognition approaches map low-level image features directly to object category labels, recent work proposes models using *visual attributes* [1–8]. Attributes are properties observable in images that have human-designated names (e.g., 'striped', 'four-legged'), and they are valuable as a new semantic cue in various problems. For example, researchers have shown their impact for strengthening facial verification [5], object recognition [6, 8, 16], generating descriptions of unfamiliar objects [1], and to facilitate "zero-shot" transfer learning [2], where one trains a classifier for an unseen object simply by specifying which attributes it has.

Problem: Most existing work focuses wholly on attributes as binary predicates indicating the presence (or absence) of a certain property in an image [1–8, 16]. This may suffice for part-based attributes (e.g., 'has a head') and some binary properties (e.g., 'spotted'). However, for a large variety of attributes, not only is this binary setting restrictive, but it is also unnatural. For instance, it is not clear if in Figure 1(b) Hugh Laurie is smiling or not; different people are likely to respond inconsistently in providing the presence or absence of the 'smiling' attribute for this image, or of the 'natural' attribute for Figure 1(e).

Indeed, we observe that *relative* visual properties are a semantically rich way by which humans describe and compare objects in the world. They are necessary, for instance, to refine an identifying description ("the 'rounder' pillow"; "the same except 'bluer'"), or to situate with respect to reference objects ("'brighter' than a candle; 'dimmer' than a flashlight"). Furthermore, they have potential to enhance active and interactive learning—for instance, offering a better guide for a visual search ("find me similar shoes, but 'shinier'." or "refine the retrieved images of downtown Chicago to those taken on 'sunnier' days").

Proposal: In this work, we propose to model *relative attributes*. As opposed to predicting the presence of an attribute, a relative attribute indicates the strength of an attribute in an image with respect to other images. For exam-

CV app: What are attributes?

- Mid-level concepts
  - Higher than low-level features
  - Lower than high-level categories
- Shared across categories
- Human-understandable (semantic)
- Machine-detectable (visual)

otter

black: yes
white: no
brown: yes
stripes: no
water: yes
eats fish: yes

polar bear

black: no
white: yes
brown: no
stripes: no
water: yes
eats fish: yes

zebra

black: yes
white: yes
brown: no
stripes: yes
water: no
eats fish: no

Face Tracer Image Search (Kumar 08) “Smiling Asian Men With Glasses”

Slide credit: Devi Parikh

CV app: Attribute Models

$\boldsymbol{x}_i \rightarrow$ Real value

Density,
Smilling,
….

"I am 60% sure this person is smiling"
(Binary Classifier Confidence)

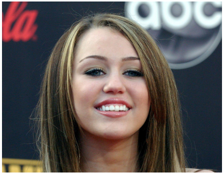
"This person is smiling 60%"
(Attribute Strength)

Slide credit: Devi Parikh

CV app: Relative Attributes

"Person A is smiling more than Person B"
[Relative Attribute, Parikh and Grauman ICCV 2011]

<
smiling

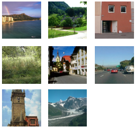


>
natural

Scarlett

- Training sets:

Attributes labeled at category level



	Binary	Relative
OSR	TI SHC OMF	
natural	0 0 0 0 1 1 1 1	$T \prec I \sim S \prec H \prec C \sim O \sim M \sim F$
open	0 0 0 1 1 1 1 0	$T \sim F \prec I \sim S \prec M \prec H \sim C \sim O$
perspective	1 1 1 1 0 0 0 0	$O \prec C \prec M \sim F \prec H \prec I \prec S \prec T$
large-objects	1 1 1 0 0 0 0 0	$F \prec O \sim M \prec I \sim S \prec H \sim C \prec T$
diagonal-plane	1 1 1 1 0 0 0 0	$F \prec O \sim M \prec C \prec I \sim S \prec H \prec T$
close-depth	1 1 1 1 0 0 0 1	$C \prec M \prec O \prec T \sim I \sim S \sim H \sim F$
PubFig	ACHJ MS VZ	
Masculine-looking	1 1 1 1 0 0 1 1	$S \prec M \prec Z \prec V \prec J \prec A \prec H \prec C$
White	0 1 1 1 1 1 1 1	$A \prec C \prec H \prec Z \prec J \prec S \prec M \prec V$
Young	0 0 0 0 1 1 0 1	$V \prec H \prec C \prec J \prec A \prec S \prec Z \prec M$
Smiling	1 1 1 0 1 1 0 1	$J \prec V \prec H \prec A \sim C \prec S \sim Z \prec M$
Chubby	1 0 0 0 0 0 0 0	$V \prec J \prec H \prec C \prec Z \prec M \prec S \prec A$
Visible-forehead	1 1 1 0 1 1 1 0	$J \prec Z \prec M \prec S \prec A \sim C \sim H \sim V$
Bushy-eyebrows	0 1 0 1 0 0 0 0	$M \prec S \prec Z \prec V \prec H \prec A \prec C \prec J$
Narrow-eyes	0 1 1 0 0 0 1 1	$M \prec J \prec S \prec A \prec H \prec C \prec V \prec Z$
Pointy-nose	0 0 1 0 0 0 0 1	$A \prec C \prec J \sim M \sim V \prec S \prec Z \prec H$
Big-lips	1 0 0 0 1 1 0 0	$H \prec J \prec V \prec Z \prec C \prec M \prec A \prec S$
Round-face	1 0 0 0 1 1 0 0	$H \prec V \prec J \prec C \prec Z \prec A \prec S \prec M$

Table 1. Binary and relative attribute assignments used in our experiments. Note that none of the relative orderings violate the binary memberships. The OSR dataset includes images from the following categories: coast (C), forest (F), highway (H), inside-city (I), mountain (M), open-country (O), street (S) and tall-building (T). The 8 attributes shown above are listed in [11] as the properties subjects used to organize the images.The PubFig dataset includes images of: Alex Rodriguez (A), Clive Owen (C), Hugh Laurie (H), Jared Leto (J), Miley Cyrus (M), Scarlett Johansson (S), Viggo Mortensen (V) and Zac Efron (Z). The 11 attributes shown above are a

CV app: Attribute Models

- Ranking functions for relative attributes

For each attribute a_m, open

Supervision = all pairs as:

	Binary	Relative
OSR	T I S H C O M F	
natural	0 0 0 0 1 1 1 1	$T \prec I \sim S \prec H \prec C \sim O \sim M \sim F$
open	0 0 0 1 1 1 1 0	$T \sim F \prec I \sim S \prec M \prec H \sim C \sim O$
perspective	1 1 1 1 0 0 0 0	$O \prec C \prec M \sim F \prec H \prec I \prec S \prec T$
large-objects	1 1 1 0 0 0 0 0	$F \prec O \sim M \prec I \sim S \prec H \sim C \prec T$
diagonal-plane	1 1 1 1 0 0 0 0	$F \prec O \sim M \prec C \prec I \sim S \prec H \prec T$
close-depth	1 1 1 1 0 0 0 1	$C \prec M \prec O \prec T \sim I \sim S \sim H \sim F$
PubFig	A C H J M S V Z	
Masculine-looking	1 1 1 1 0 0 1 1	$S \prec M \prec Z \prec V \prec J \prec A \prec H \prec C$
White	0 1 1 1 1 1 1 1	$A \prec C \prec H \prec Z \prec J \prec S \prec M \prec V$
Young	0 0 0 0 1 1 0 1	$V \prec H \prec C \prec J \prec A \prec S \prec Z \prec M$
Smiling	1 1 1 0 1 1 0 1	$J \prec V \prec H \prec A \sim C \prec S \sim Z \prec M$
Chubby	1 0 0 0 0 0 0 0	$V \prec J \prec H \prec C \prec Z \prec M \prec S \prec A$
Visible-forehead	1 1 1 0 1 1 1 0	$J \prec Z \prec M \prec S \prec A \sim C \sim H \sim V$
Bushy-eyebrows	0 1 0 1 0 0 0 0	$M \prec S \prec Z \prec V \prec H \prec A \prec C \prec J$
Narrow-eyes	0 1 1 0 0 0 1 1	$M \prec J \prec S \prec A \prec H \prec C \prec V \prec Z$
Pointy-nose	0 0 1 0 0 0 0 1	$A \prec C \prec J \sim M \sim V \prec S \prec Z \prec H$
Big-lips	1 0 0 0 1 1 0 0	$H \prec J \prec V \prec Z \prec C \prec M \prec A \prec S$
Round-face	1 0 0 0 1 1 0 0	$H \prec V \prec J \prec C \prec Z \prec A \prec S \prec M$

$O_m : \{($  $\succ$  $), \ldots\},$

$S_m : \{\{$  $\sim$  $\}, \ldots\}$

CV app: pairwise ranking

- Coarse labeling at category level => noisy pair sampling

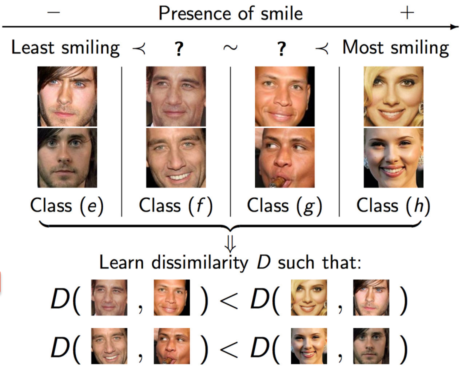
	Binary	Relative
OSR	T I S H C O M F	
natural	0 0 0 0 1 1 1 1	$T \prec I \sim S \prec H \prec C \sim O \sim M \sim F$
open	0 0 0 1 1 1 1 0	$T \sim F \prec I \sim S \prec M \prec H \sim C \sim O$
perspective	1 1 1 1 0 0 0 0	$O \prec C \prec M \sim F \prec H \prec I \prec S \prec T$
large-objects	1 1 1 0 0 0 0 0	$F \prec O \sim M \prec I \sim S \prec H \sim C \prec T$
diagonal-plane	1 1 1 1 0 0 0 0	$F \prec O \sim M \prec C \prec I \sim S \prec H \prec T$
close-depth	1 1 1 1 0 0 0 1	$C \prec M \prec O \prec T \sim I \sim S \sim H \sim F$
PubFig	A C H J M S V Z	
Masculine-looking	1 1 1 1 0 0 1 1	$S \prec M \prec Z \prec V \prec J \prec A \prec H \prec C$
White	0 1 1 1 1 1 1 1	$A \prec C \prec H \prec Z \prec J \prec S \prec M \prec V$
Young	0 0 0 0 1 1 0 1	$V \prec H \prec C \prec J \prec A \prec S \prec Z \prec M$
Smiling	1 1 1 0 1 1 0 1	$J \prec V \prec H \prec A \sim C \prec S \sim Z \prec M$
Chubby	1 0 0 0 0 0 0 0	$V \prec J \prec H \prec C \prec Z \prec M \prec S \prec A$
Visible-forehead	1 1 1 0 1 1 1 0	$J \prec Z \prec M \prec S \prec A \sim C \sim H \sim V$
Bushy-eyebrows	0 1 0 1 0 0 0 0	$M \prec S \prec Z \prec V \prec H \prec A \prec C \prec J$
Narrow-eyes	0 1 1 0 0 0 1 1	$M \prec J \prec S \prec A \prec H \prec C \prec V \prec Z$
Pointy-nose	0 0 1 0 0 0 0 1	$A \prec C \prec J \sim M \sim V \prec S \prec Z \prec H$
Big-lips	1 0 0 0 1 1 0 0	$H \prec J \prec V \prec Z \prec C \prec M \prec A \prec S$
Round-face	1 0 0 0 1 1 0 0	$H \prec V \prec J \prec C \prec Z \prec A \prec S \prec M$

Scarlett Johansson *vs* Miley Cyrus

- Quadruplet to minimize this artefact

CV app: Quadruplet-wise ML

	Binary	Relative
OSR	T I S H C O M F	
natural	0 0 0 0 1 1 1 1	T≺I∼S≺H≺C∼O∼M∼F
open	0 0 0 1 1 1 1 0	T∼F≺I∼S≺M≺H∼C∼O
perspective	1 1 1 1 0 0 0 0	O≺C≺M∼F≺H≺I≺S≺T
large-objects	1 1 1 0 0 0 0 0	F≺O∼M≺I∼S≺H∼C≺T
diagonal-plane	1 1 1 1 0 0 0 0	F≺O∼M≺C≺I∼S≺H≺T
close-depth	1 1 1 1 0 0 0 1	C≺M≺O≺T∼I∼S∼H∼F
PubFig	A C H J M S V Z	
Masculine-looking	1 1 1 1 0 0 1 1	S≺M≺Z≺V≺J≺A≺H≺C
White	0 1 1 1 1 1 1 1	A≺C≺H≺Z≺J≺S≺M≺V
Young	0 0 0 0 1 1 0 1	V≺H≺C≺J≺A≺S≺Z≺M
Smiling	1 1 1 0 1 1 0 1	J≺V≺H≺A∼C≺S∼Z≺M
Chubby	1 0 0 0 0 0 0 0	V≺J≺H≺C≺Z≺M≺S≺A
Visible-forehead	1 1 1 0 1 1 1 0	J≺Z≺M≺S≺A∼C∼H∼V
Bushy-eyebrows	0 1 0 1 0 0 0 0	M≺S≺Z≺V≺H≺A≺C≺J
Narrow-eyes	0 1 1 0 0 0 1 1	M≺J≺S≺A≺H≺C≺V≺Z
Pointy-nose	0 0 1 0 0 0 0 1	A≺C≺J∼M∼V≺S≺Z≺H
Big-lips	1 0 0 0 1 1 0 0	H≺J≺V≺Z≺C≺M≺A≺S
Round-face	1 0 0 0 1 1 0 0	H≺V≺J≺C≺Z≺A≺S≺M



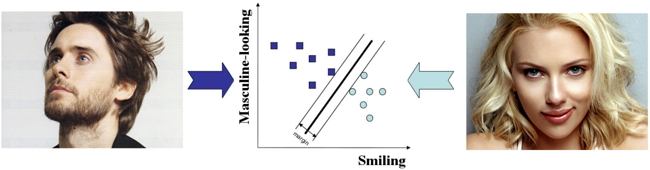
- Relative attributes => (Dis)similarity Learning under Qwise constraints

Relative attribute learning

- Learning a feature space

$$\begin{aligned} D^2_{\mathbf{M}}(p_i, p_j) &= \Phi(p_i, p_j)^\top \mathbf{M} \Phi(p_i, p_j) \\ &= (\mathbf{x}_i - \mathbf{x}_j)^\top \mathbf{L}^\top \mathbf{L} (\mathbf{x}_i - \mathbf{x}_j) \end{aligned}$$

- Corresponds to learn a linear transformation parameterized by $\mathbf{L} \in \mathbb{R}^{M \times d}$ such that $\mathbf{h}_i = \mathbf{L}\mathbf{x}_i$ where the m-th row of $\mathbf{L}$ is $\mathbf{w}_m^\top$
- Application to Actor retrieval and classification:



Relative attribute learning

$$\min_{\mathbf{w}} \mu \|\mathbf{w}\|_2^2 + \sum_{(p_i, p_j, p_k, p_l)} \ell \left(\mathbf{w}^\top \left[\Psi(p_k, p_l) - \Psi(p_i, p_j) \right] \right)$$

- $\mathbf{x}_i \in \mathbb{R}^d$: GIST (+ color) descriptor
- $\Psi(p_i, p_j) = \mathbf{x}_i - \mathbf{x}_j$
- Relative attributes a_m for $m \in \{1, \ldots, M\}$: smiling, masculine-looking young...
- Learning a $\mathbf{w}_m$ for each attribute a_m using Qwise optimization
- Resulting in learning a linear transformation parameterized by $\mathbf{L} \in \mathbb{R}^{M \times d}$

$$\mathbf{L} = \begin{bmatrix} w_{1,1} & \ldots & w_{1,d} \\ \vdots & \vdots & \vdots \\ w_{M,1} & \ldots & w_{M,d} \end{bmatrix} = \begin{bmatrix} \mathbf{w}_1^\top \\ \vdots \\ \mathbf{w}_M^\top \end{bmatrix}, \ \mathbf{w}_m^\top : m\text{-th row}$$

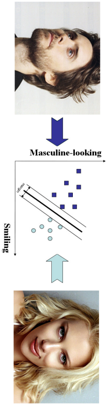
Relative attribute experiments

- Outdoor Scene Recognition OSR [Oliva 01]
- 8 classes, ~2700 images, GIST
- 6 attributes: open, natural …

- Public Figures Faces PubFig [Kumar 09]
- 8 classes, ~800 images, GIST +color
- 11 attributes: smiling, shubby …

Relative attribute experiments

- Baselines
  - RA Relative attribute method (Parikh and Grauman)
    - annotations on class relationships with pairwise constraints
  - LMNN Linear transformation learned
    - class membership information used only unlike RA
  - RA + LMNN: Combination of the first two baselines
    1. Relative attribute annotations to learn attribute space
    2. Metric in attribute space with LMNN
- Qwise Method:
  - Qwise constraints generated as pairwise
  - Qwise output alone or combined Qwise + LMNN

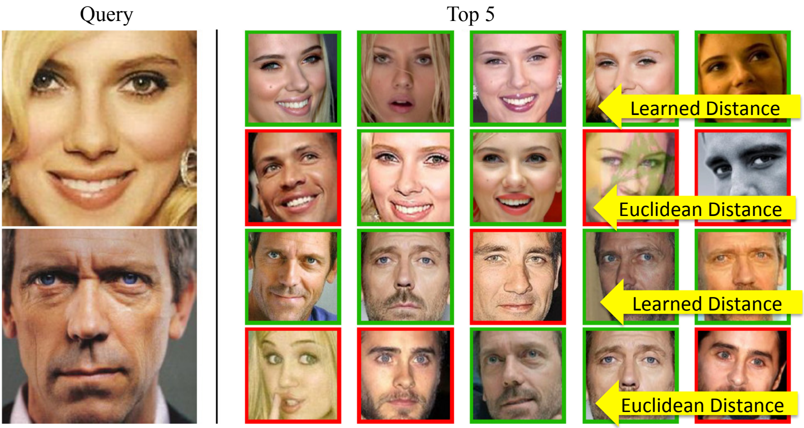


Relative attribute experiments

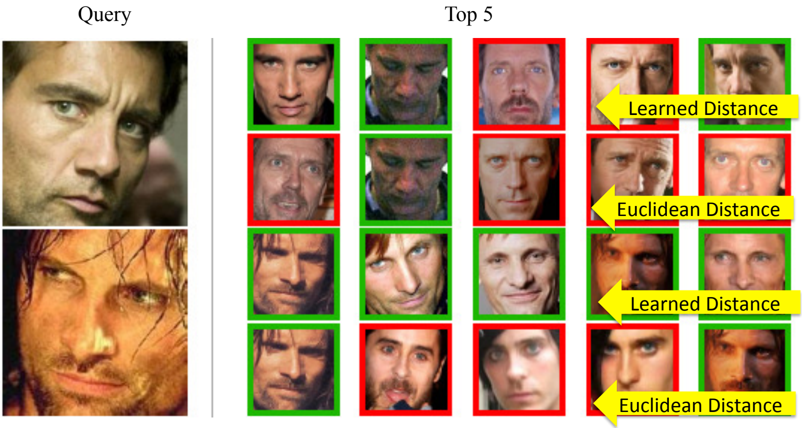
	OSR	Pubfig
Parikh's code	$71.3 \pm 1.9\%$	$71.3 \pm 2.0\%$
LMNN-G	$70.7 \pm 1.9\%$	$69.9 \pm 2.0\%$
LMNN	$71.2 \pm 2.0\%$	$71.5 \pm 1.6\%$
RA + LMNN	$71.8 \pm 1.7\%$	$74.2 \pm 1.9\%$
Qwise	$74.1 \pm 2.1\%$	$74.5 \pm 1.3\%$
Qwise + LMNN-G	$\mathbf{74.6} \pm \mathbf{1.7}\%$	$76.5 \pm 1.2\%$
Qwise + LMNN	$74.3 \pm 1.9\%$	$\mathbf{77.6} \pm \mathbf{2.0}\%$

Table 1: Test classification accuracies on the OSR and Pubfig datasets for different methods.

Relative attribute experiments



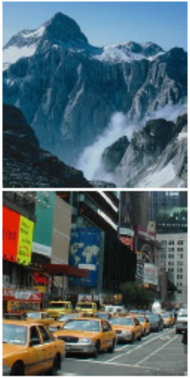
Relative attribute experiments



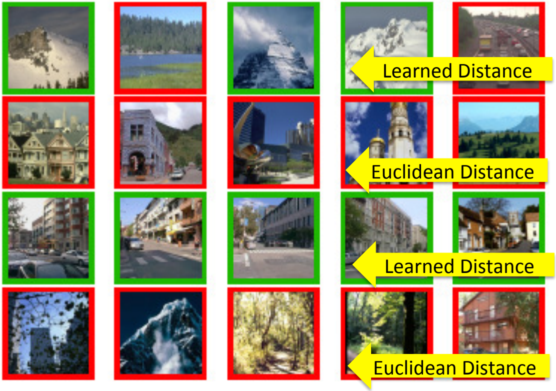
Relative attribute experiments

Relative attribute experiments

Query



Top 5

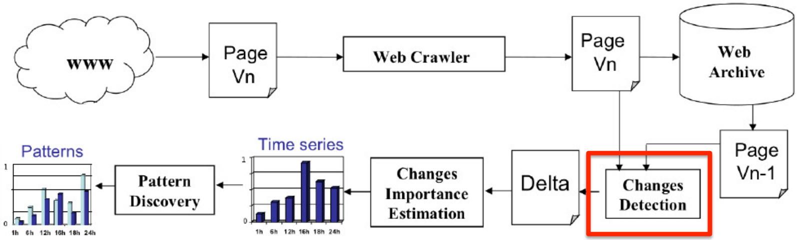


Outline

1. Introduction
2. Metric Learning
3. Computer Vision Applications
   - Relative attribute learning
   - **Web page comparison**

Web page ML

- Context:
  - For Web crawling purpose, useful to understand the change behavior of websites over time



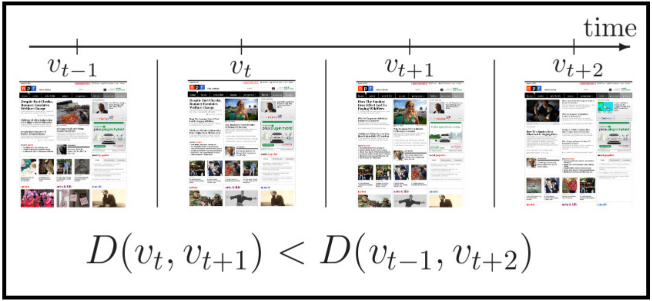
  - Significant changes between successive versions of a same webpage => revisit the page
- Web page comparison
  - Learning Web page metric and significant webpage regions

Web page ML

- Focus on news websites
  - Advertisements or menus not significant
  - News content significant
- Find a metric able to properly identify **significant** changes between webpage versions
- Localize changes inside pages:
  - semantic spatial structure
  - significant to capture

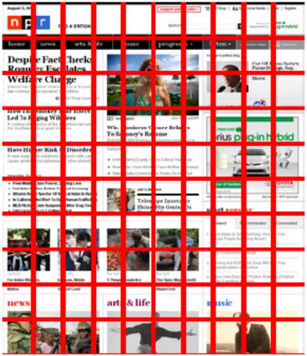
Web page ML

- Temporal info. to get Pair/Triplet/Qwise Constraints:
  - Adjacent screenshots in a temporal sequence of a web site are more likely to be semantically similar than distant frames
  - Fully unsupervised ML (just using temporal information available)
  - Constraints by comparing screenshots of successive webpage versions:



Web page ML

- Descriptors: classical image descriptors over a spatial m-by-m image grid
- **Ψ** is a m-by-m vector of Euclidean distance between blocks
- Diagonal PSD matrix: **w** represents block weights
- Optimization over **w**
  - Learning of spatial weights of webpage regions using temporal relationships
  - Discovering important change regions
  - Ignoring menus and advertisements

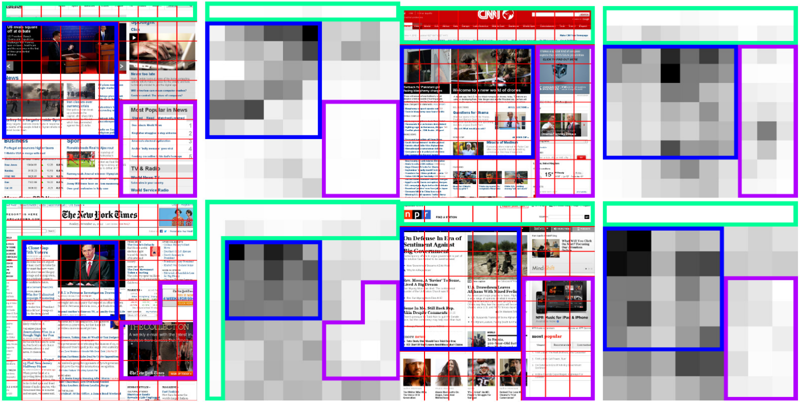


Web page ML

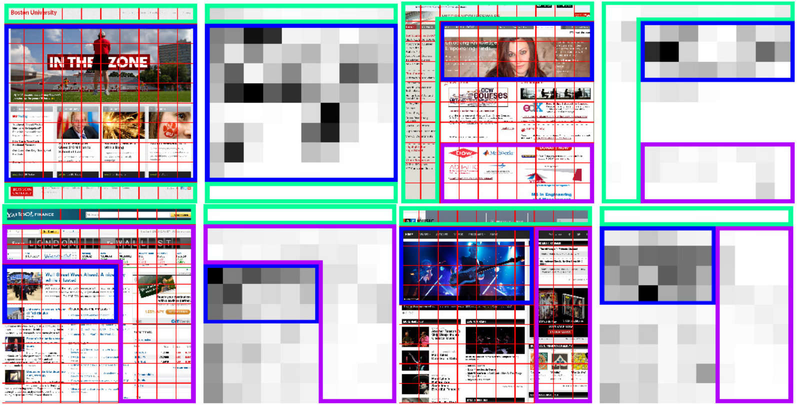
- Evaluation and Comparison [Law PhD 2015]
  - Crawling 50 days Several sites CNN, NPR, BBC, …
  - Manual change detection (news updates) for GT on 5 days
  - Baselines: Euclidean Dist, LMNN
  - GIST on 10x10
  - Mean Average Precision on succ. Web page Metric scores

Site	CNN			NPR			New York Times			BBC		
Eval.	AP_S	AP_D	MAP	AP_S	AP_D	MAP	AP_S	AP_D	MAP	AP_S	AP_D	MAP
Eucl. Dist.	68.1 ±0.6	85.9 ±0.6	77.0 ±0.5	96.3 ±0.2	89.5 ±0.5	92.9 ±0.3	69.8 ±0.9	79.5 ±0.4	74.6 ±0.5	91.1 ±0.3	76.7 ±0.6	83.9 ±0.4
LMNN	78.8 ±1.9	91.7 ±1.7	85.2 ±1.8	98.0 ±0.6	92.5 ±1.1	95.2 ±0.9	83.2 ±1.4	89.1 ±2.7	86.1 ±2.0	92.5 ±0.4	**80.1 ±1.0**	**86.3 ±0.6**
Qwise	**82.7 ±4.1**	**94.6 ±1.8**	**88.6 ±2.9**	**98.6 ±0.2**	**94.3 ±0.6**	**96.5 ±0.4**	**85.5 ±5.4**	**92.3 ±4.1**	**88.9 ±4.6**	**92.8 ±0.4**	79.3 ±1.3	86.1 ±0.8

Web page ML



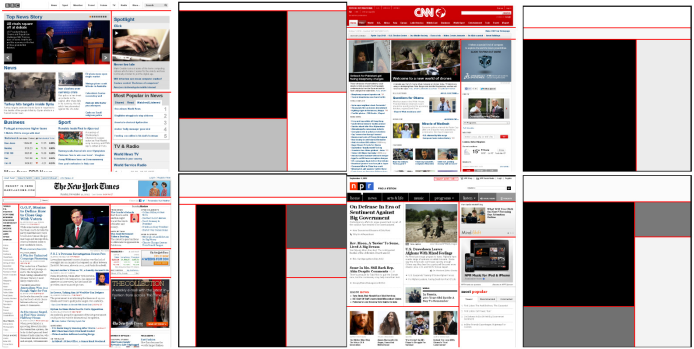
Web page ML



- Not connected to the structural layout of the Web page

Web page ML

- Detect significant changes using the source code of pages (Segmentation) + Qwise



Key issues in Metric Learning for CV

- Modeling: Data representation, type of metric (linear, non lin., local)
  - Connection to deep : deep features + metric learn on top
- Learning Paradigm: unsupervised, semi-supervised, transfer**, type of constraints**
  - Temporal/spatial relationships [LeCun ICCV 2015]
  - Class/Structure relationships => rich context to learn metrics or semantic embedding
- Optimization issues: Global/local solution, Convexity, Scalability, …
- Learning joint embedding

General conclusion of this tutorial

- Ongoing and open topics
  - Adapting metrics to changing data
    - Lifelong learning, etc
  - Unsupervised metric learning
    - What is a good metric for clustering?
    - Denoising / Robustness to invariance
  - Learning richer metrics
    - Different degrees of similarity
    - Several co-existing notions of similarity
  - Relation to representation learning

References

Team ref. on related subjects (*many Codes on project web pages or available on demand*):
- C. LeBarz *et al* Exemplar based metric learning for robust visual localization, ICIP 2015
- M.T. Law, N. Thome, M. Cord. Fantope Regularization in Metric Learning, CVPR 2014
- M.T. Law, N. Thome, M. Cord. Quadruplet-wise Image Similarity Learning, ICCV 2013
- M.T. Law, N. Thome, S. Gancarski, M. Cord. Structural and Visual Comparisons for Web Page Archiving, ACM DocEng, 2012
- S. Avila, N. Thome, M. Cord, E. Valle, A. Araujo, Pooling in Image Representation: the Visual Codeword Point of View, CVIU 2013
- H. Goh, N. Thome, M. Cord, JH. Lim, Top-Down Regularization of Deep Belief Networks, NIPS 2013

Others:
- R. Hadsell, S. Chopra, Y. LeCun, Dimensionality Reduction by Learning an Invariant Mapping, CVPR 2006
- S. Chopra, R. Hadsell, Y. LeCun, Learning a Similarity Metric Discriminatively, CVPR 2005
- *J. Hu, J. Lu, Y. Tan,* Discriminative Deep Metric Learning for Face Verification in the Wild, CVPR 2014
- Y. Sun, Y. Chen, X. Wang, X. Tang, Deep Learning Face Representation by Joint Identification-Verification, NIPS 2014
- N. Carlevaris-Bianco, RM. Eustice, Learning visual feature descriptors for dynamic lighting conditions, IROS 2014
- N. Verma et al, Learning Hierarchical Similarity Metrics, CVPR 2012
- K.Q. Weinberger, L.K. Saul, Distance metric learning for large margin nearest neighbor classication, JMLR 2009
- R. Goroshin, J. Bruna, Y. LeCun, Unsupervised Learning of Spatiotemporally Coherent Metrics, ICCV 2015