METRIC LEARNING FOR LARGE-SCALE DATA

Aurélien Bellet

MAGNET Project-Team, Inria

Seminar Statistical Machine Learning (SMILE) in Paris
April 28, 2016

A BIT ABOUT ME

- 2009-12: **Ph.D.**, Université de Saint-Etienne
  - Supervisors: Marc Sebban, Amaury Habrard
- 2013-14: **Postdoc**, University of Southern California
  - Working with Fei Sha
- 2014-15: **Postdoc**, Télécom ParisTech
  - Working with Stéphan Clémençon
- Since nov. 2015: **Junior researcher** (CR2), Inria Lille
  - Magnet Team (head: Marc Tommasi)

OUTLINE

1. Introduction
2. A brief review of metric learning methods
3. Metric learning on large datasets
4. Metric learning in high dimensions

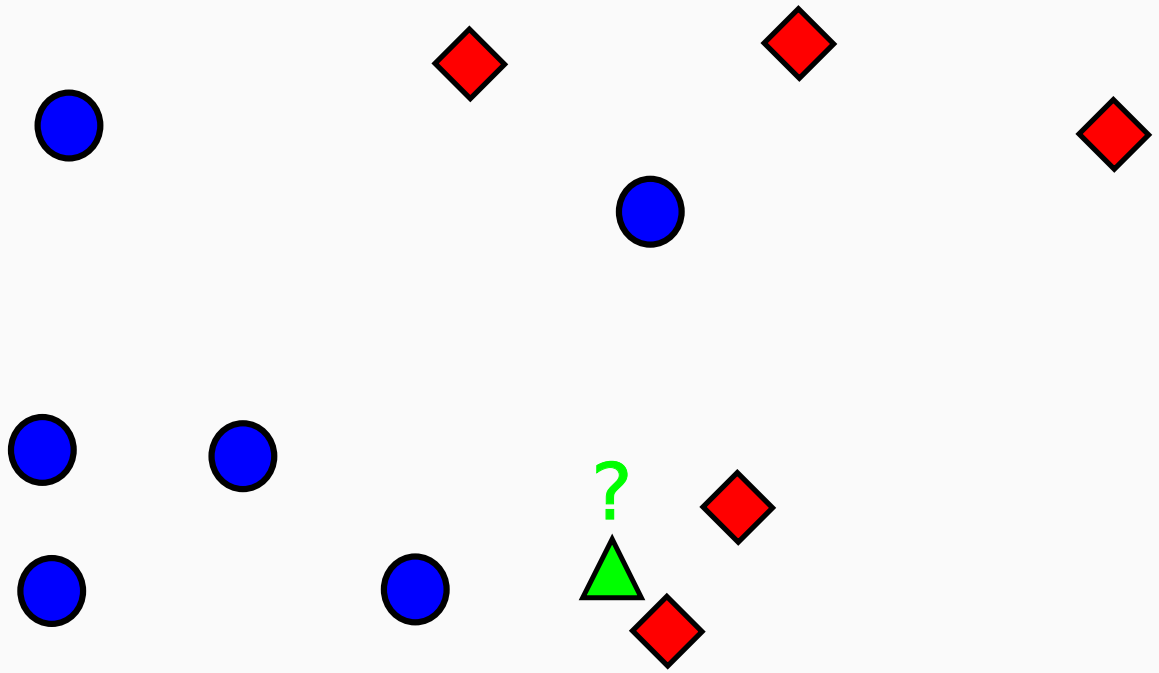
INTRODUCTION

FUNDAMENTAL QUESTION OF THIS TALK

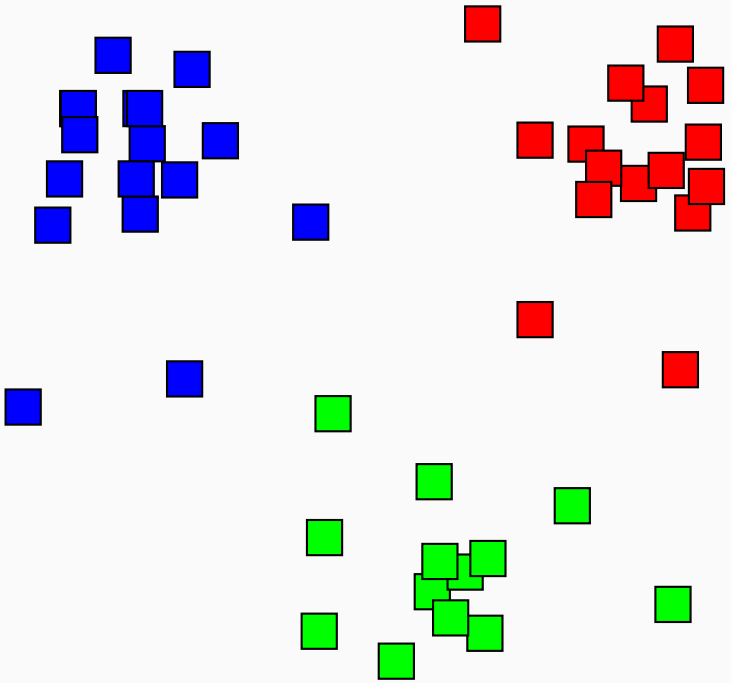
How to appropriately measure similarity or distance between things depending on the context?

- We (humans) are good at this [Tversky, 1977, Goldstone et al., 1997]
  - Recognize similar objects, sounds, ideas, etc, from past experience
  - Adapt the notion of similarity to the context
- Artificial systems need to do it too!
  - Categorize / retrieve data based on similarity to known examples
  - Detect situations similar to past experience

Nearest neighbor classification

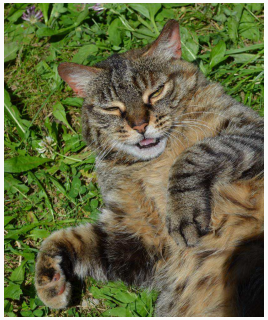


Clustering



Information retrieval

Query document

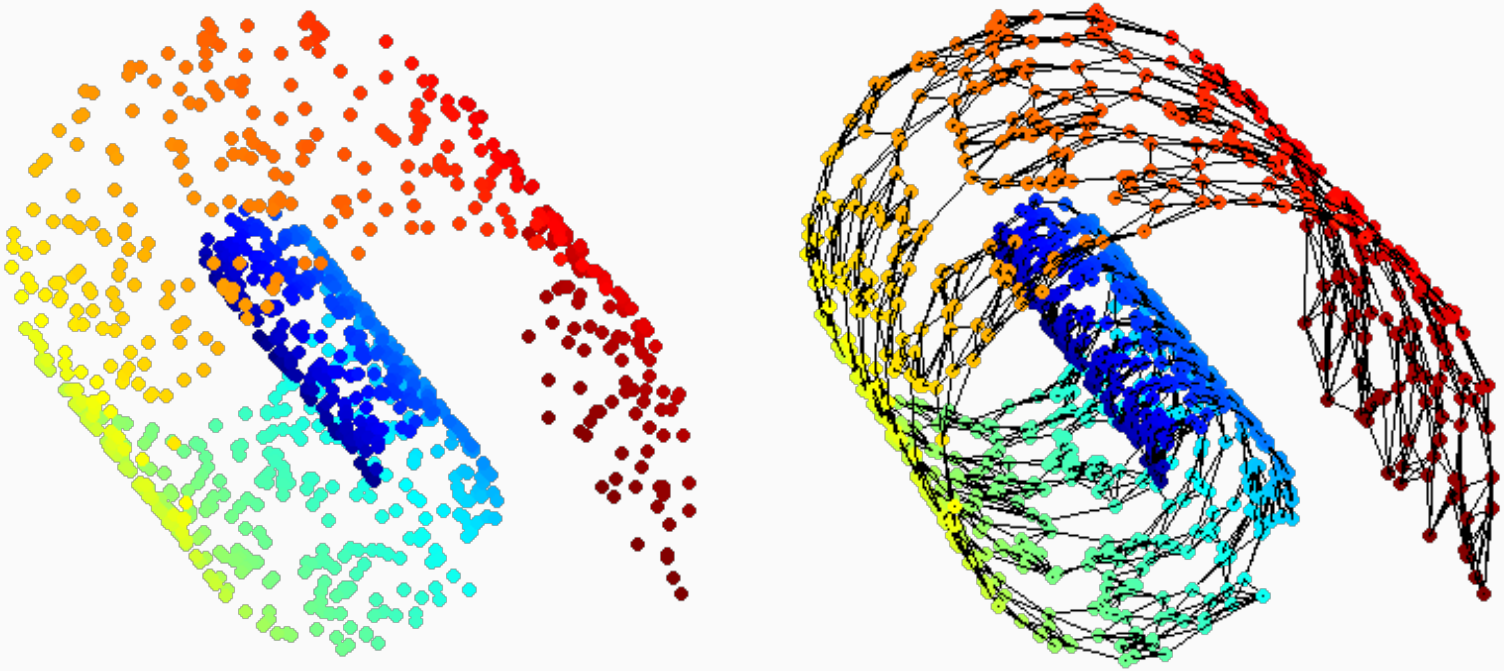


Most similar documents





Manifold learning/regularization



A GENERAL APPROACH: METRIC LEARNING

- Assume data represented in space $\mathcal{X}$ (e.g., $\mathcal{X} \subset \mathbb{R}^p$)
- We provide the system with some similarity judgments on data pairs/triplets for the task of interest


similar to



not similar to



more similar to

than to


(images taken from Caltech Faces dataset)

- The system uses this information to find the most "appropriate" pairwise similarity/distance function $F : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$

(Note: I will refer to F as a metric regardless of its properties)

WHY NOT SIMPLY LEARN A CLASSIFIER?

- Case 1: huge number of classes (likely with class imbalance)
  - No need to learn many classifiers (as in 1-vs-1, 1-vs-all)
  - No blow-up in number of parameters (as in Multinomial Log. Reg.)
- Case 2: labels are unknown (and costly to obtain)
  - Similarity judgments often easier to label than individual points
  - Fully unsupervised generation possible in some applications
- Case 3: a pairwise metric is really what we need
  - Information retrieval (rank results by similarity to a query)

EXAMPLE APPLICATION: FACE IDENTIFICATION

- Face identification combines all of the above
  - Huge number of classes, with few instances in each class
  - Similarity judgments easy to crowdsource / generate
  - Given a new image, rank database by similarity and decide whether to match
- State-of-the-art results in empirical evaluations
  - Labeled Faces in the Wild [Zhu et al., 2015]
  - YouTube Faces [Hu et al., 2014]
- Popular in industry as well

(examples of positive pairs correctly classified from [Guillaumin et al., 2009])

A BRIEF REVIEW OF METRIC LEARNING METHODS

GENERAL RECIPE

1. Pick a family of metrics
   - 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

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

MAHALANOBIS DISTANCE

- Mahalanobis (pseudo) distance:

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

  $M \in \mathbb{S}_+^p$ symmetric positive semi-definite (PSD) $p \times p$ 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 $\text{rank}(M) = k < p$, $L \in \mathbb{R}^{k \times p}$ performs dimensionality reduction

- For convenience, we will often work with the squared distance

Metric learning for clustering [Xing et al., 2002]

- Targeted task: clustering with side information

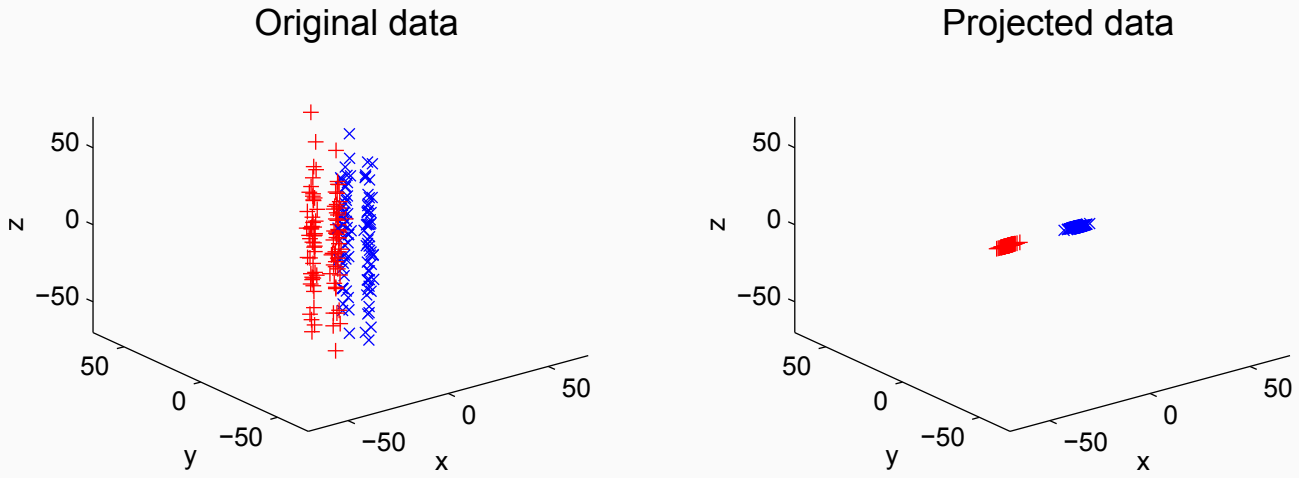
Formulation

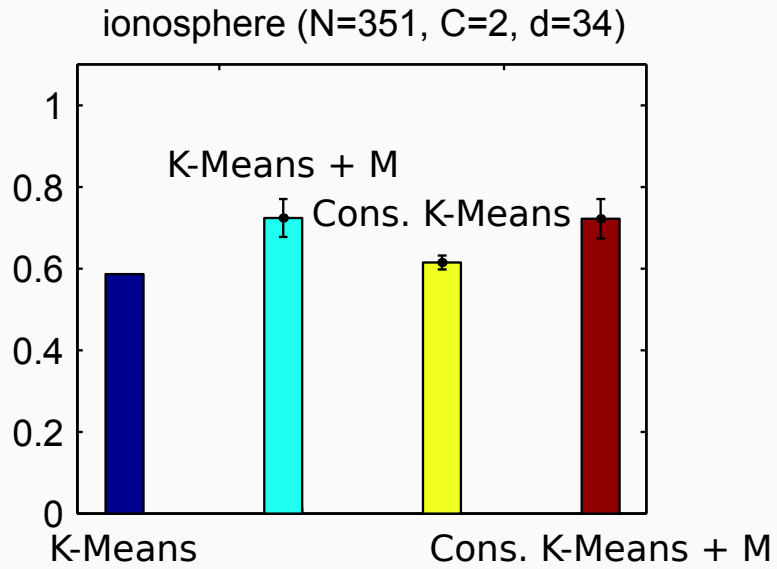
$$\max_{M \in \mathbb{S}_+^p} \sum_{(x_i, x_j) \in \mathcal{D}} D_M(x_i, x_j)$$

$$\text{s.t.} \sum_{(x_i, x_j) \in \mathcal{S}} D_M^2(x_i, x_j) \leq 1$$

- Problem is convex in M and always feasible (take $M = 0$)
- Solved with projected gradient ascent
  - Project onto distance constraint: $O(p^2)$ time
  - Project onto $\mathbb{S}_+^p$: $O(p^3)$ time (eigenvalue thresholding)
- Criterion based on sums of distances, as in K-Means

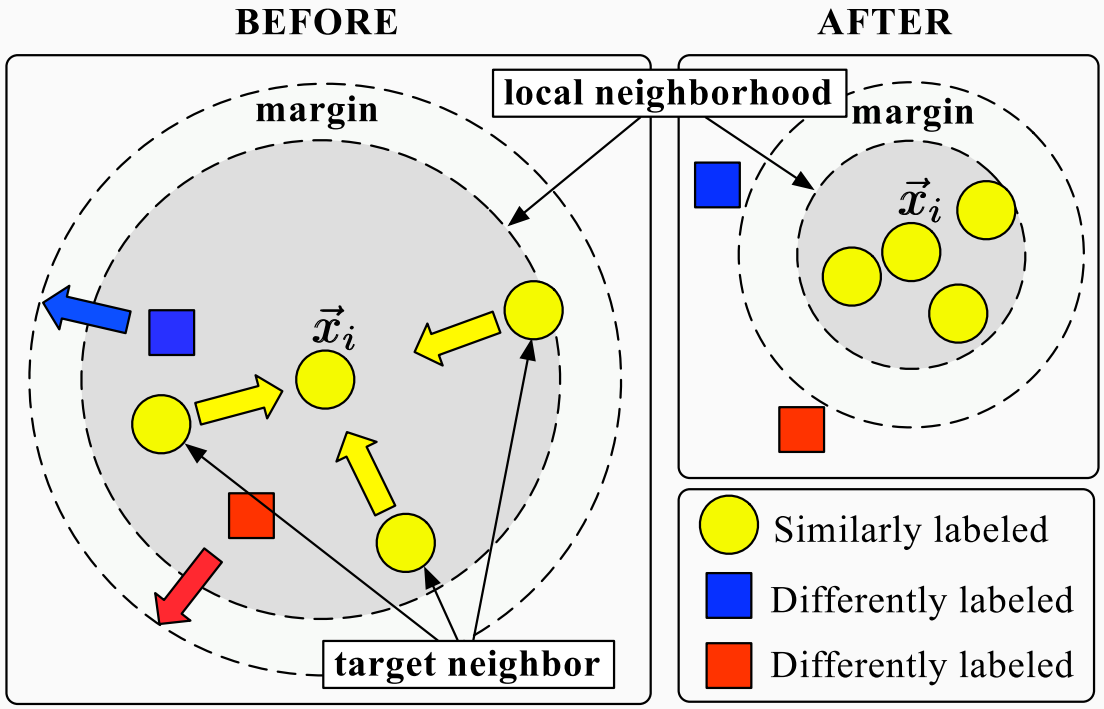
Metric learning for clustering [Xing et al., 2002]





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

- Targeted task: k-NN classification
- Constraints derived from labeled data $\{(x_i, y_i)\}_{i=1}^n$
  - $\mathcal{S} = \{(x_i, x_j) : y_i = y_j, x_j \text{ belongs to } k\text{-neighborhood of } x_i\}$
  - $\mathcal{R} = \{(x_i, x_j, x_k) : (x_i, x_j) \in \mathcal{S}, y_i \neq y_k\}$



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

Formulation

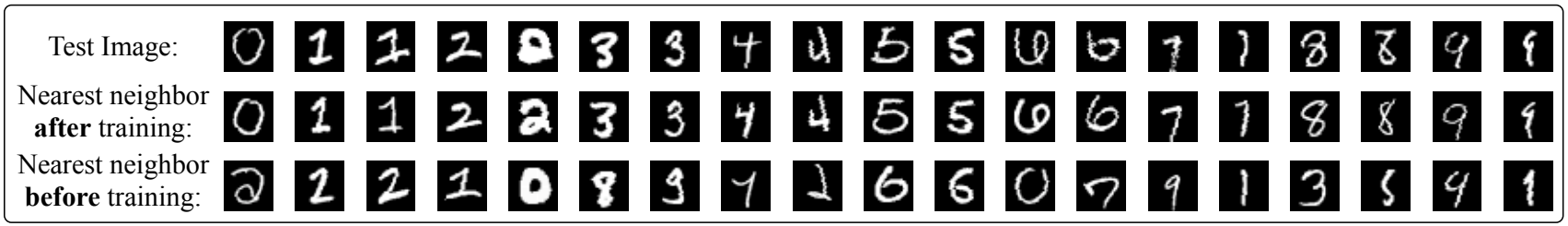
$$\min_{M \in \mathbb{S}_+^p, \boldsymbol{\xi} \geq 0} \quad (1-\mu) \sum_{(x_i, x_j) \in \mathcal{S}} D_M^2(x_i, x_j) \quad + \quad \mu \sum_{i,j,k} \xi_{ijk}$$

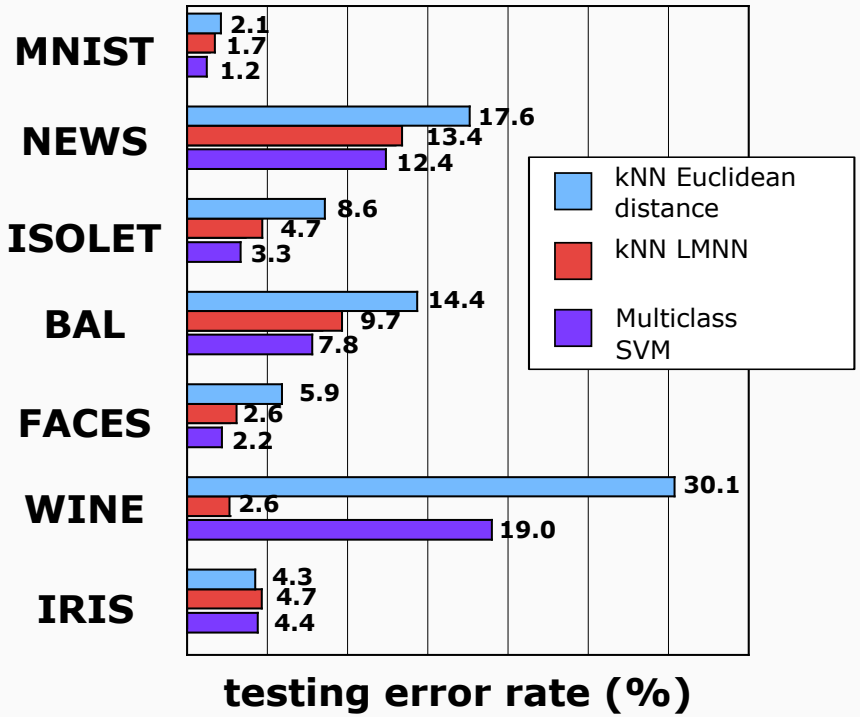
$$\text{s.t.} \quad D_M^2(x_i, x_k) - D_M^2(x_i, x_j) \geq 1 - \xi_{ijk} \quad \forall (x_i, x_j, x_k) \in \mathcal{R}$$

$\mu \in [0, 1]$ 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 the constraints?

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





Some algorithms for other tasks

- Learning to rank [McFee and Lanckriet, 2010]
- 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

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

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

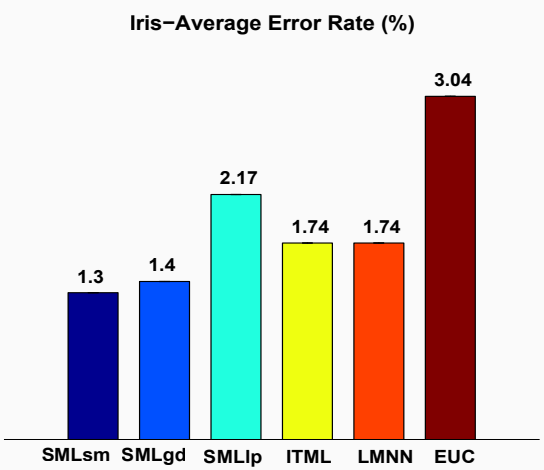
where $\boldsymbol{M} = \boldsymbol{V}\boldsymbol{\Sigma}\boldsymbol{V}^T$ and $\boldsymbol{M}_0 = \boldsymbol{U}\boldsymbol{\Theta}\boldsymbol{U}^T$ is positive definite (PD)

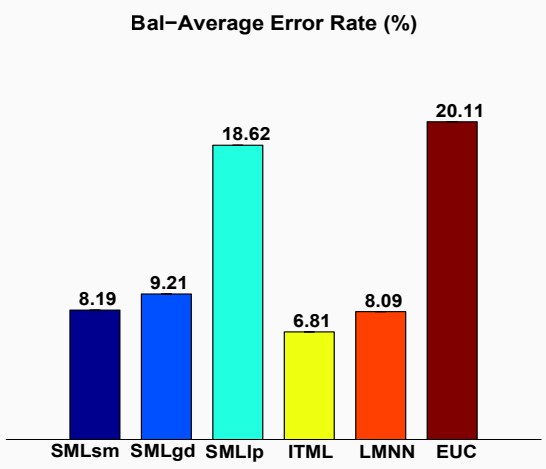
- Tends to stay close to good prior metric $\boldsymbol{M}_0$ (e.g., identity matrix)
- Convex in $\boldsymbol{M}$ (determinant of PD matrix is log-concave)
- Implicitly ensure that $\boldsymbol{M}$ is PD
- Efficient Bregman projections: $O(p^2)$ time

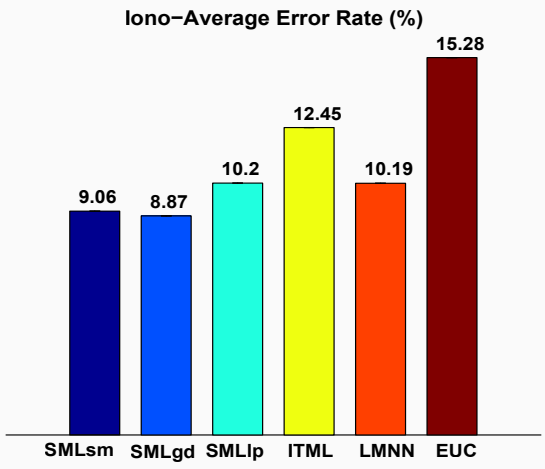
Interesting regularizers

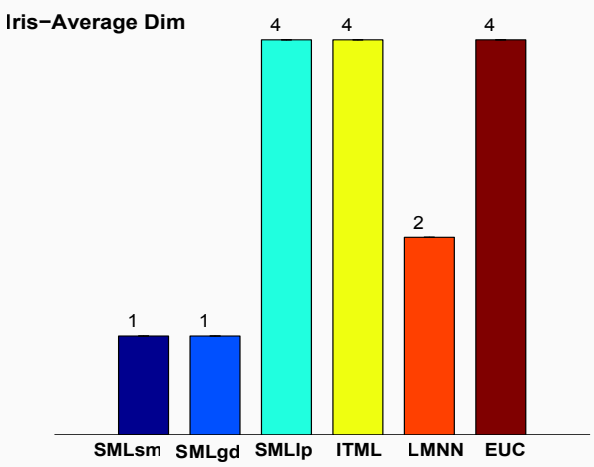
- Mixed $L_{2,1}$ norm $\|M\|_{2,1} = \sum_{i=1}^{p} \|M_i\|_2$
  - Tends to zero-out entire columns $\rightarrow$ feature selection
  - Used in [Ying et al., 2009]
  - Convex but nonsmooth
  - Efficient proximal gradient algorithms (see e.g., [Bach et al., 2012])
- Trace (or nuclear) norm $\|M\|_* =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

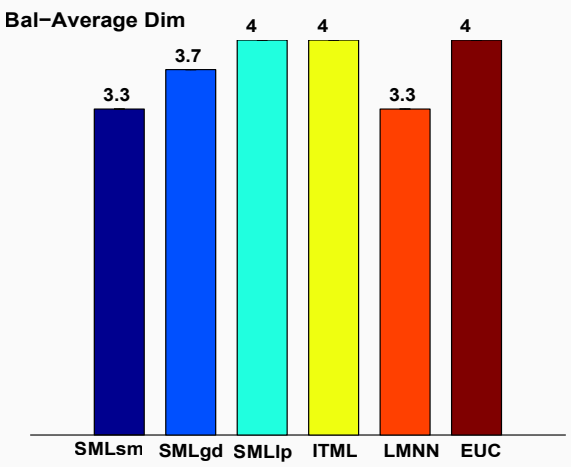
$L_{2,1}$ norm illustration

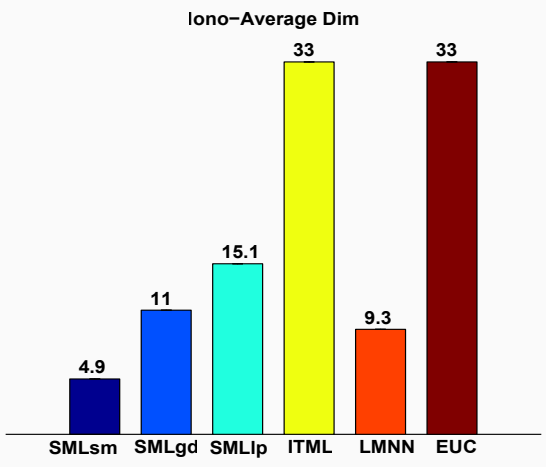








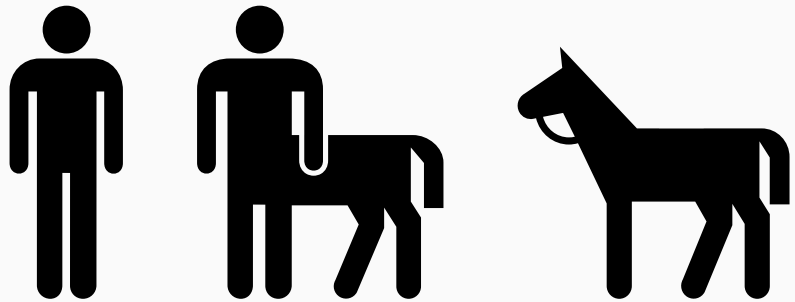




(image taken from [Ying et al., 2009])

CHALLENGING THE DISTANCE PROPERTIES

- Mahalanobis distance satisfies some distance properties
  - Nonnegativity, symmetry, triangle inequality
  - Natural regularization, required by some applications
- In practice, these properties may not be satisfied
  - 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., 2012, Cheng, 2013]
  - No PSD constraint on $M \rightarrow$ computationally easier
  - Theory of learning with similarity functions [Balcan and Blum, 2006]

NONLINEAR EXTENSIONS

- So far, we have essentially been learning a linear projection
  - Convex formulations, robustness to overfitting
  - Inability to capture nonlinear structure
- More flexible approach: learn nonlinear mapping ϕ to optimize

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

- Possible parameterizations for ϕ:
  - Mahalanobis distance in kernel space [Chatpatanasiri et al., 2010]
  - Regression trees [Kedem et al., 2012]
  - Deep neural nets [Chopra et al., 2005, Hu et al., 2014]
- The resulting learning problems are often nonconvex

Multiple Metric LMNN [Weinberger and Saul, 2009]

- Learn a metric per cluster in a coupled fashion

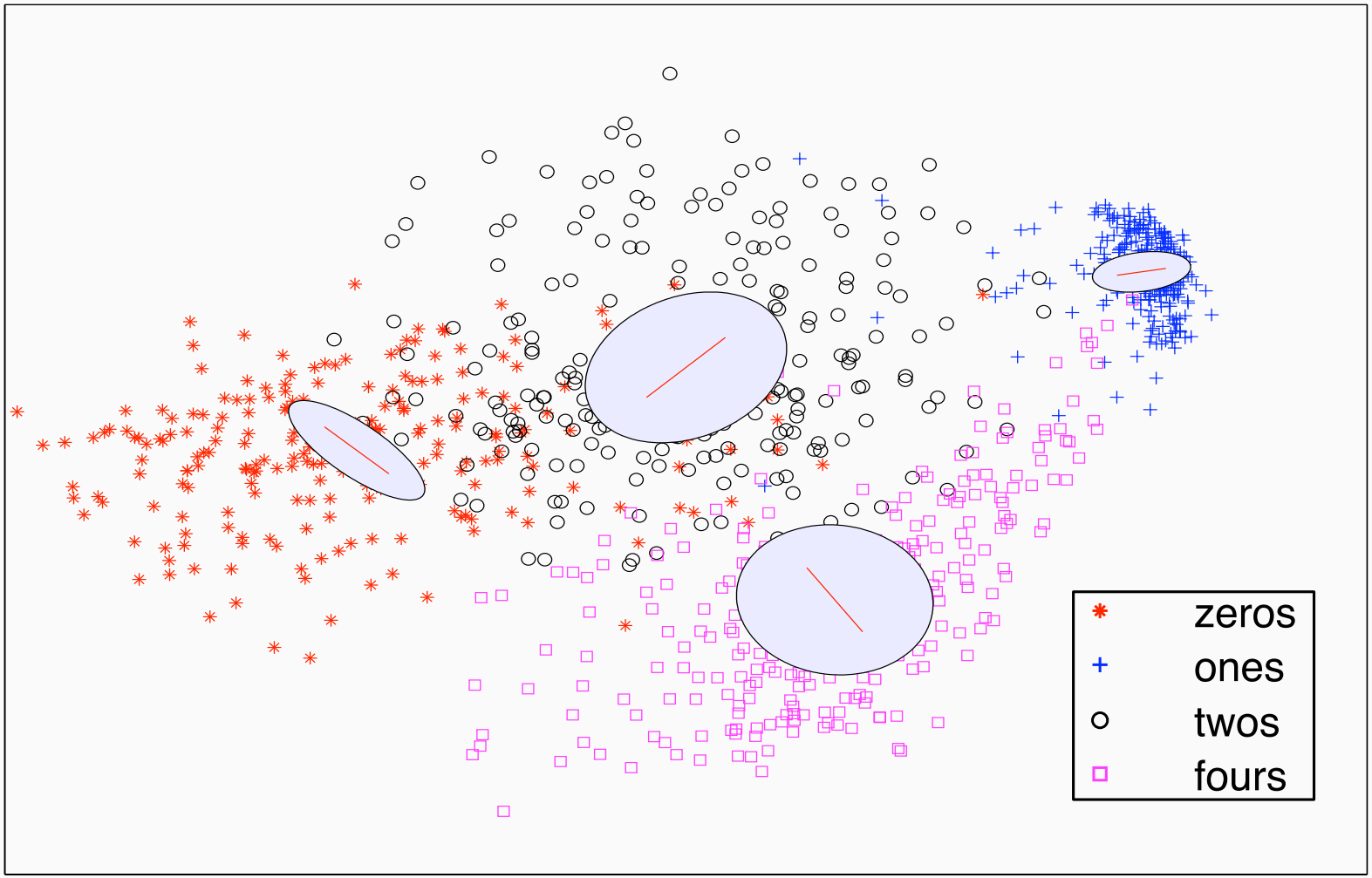
Formulation

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

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

- The learning problem remains convex
- Prone to overfitting
- More effective methods exist [Wang et al., 2012, Shi et al., 2014]

Multiple Metric LMNN [Weinberger and Saul, 2009]



STATISTICAL FRAMEWORK FOR SUPERVISED METRIC LEARNING

- Feature space $\mathcal{X} \subset \mathbb{R}^p$
- Discrete label space $\mathcal{Y} = \{1, \dots, C\}$
- Unknown distribution P over $\mathcal{X} \times \mathcal{Y}$
- We have access to a training set of n labeled observations

$$z_i = (x_i, y_i) \sim P, \qquad i = 1, \dots, n$$

- Let us consider similarity functions $S_M : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ parameterized by a symmetric matrix $M \in \mathbb{R}^{p \times p}$

STATISTICAL FRAMEWORK FOR SUPERVISED METRIC LEARNING

- We measure the performance of S_M on a data pair (z, z') using a loss function $L(M, z, z')$ convex in M, for instance

$$L(M, z, z') = \begin{cases} \max(0, 1 - S_M(x, x')) & \text{if } y = y' \\ \max(0, S_M(x, x') - 1) & \text{if } y \neq y' \end{cases}$$

  (Note: we can also define the loss on data triplets, as in LMNN)

- Goal: minimize the empirical risk (average loss on training set)

$$\min_{M \in \mathbb{R}^{d \times d}} R_n(M) = \frac{2}{n(n-1)} \sum_{1 \leq i < j \leq n} L(M, z_i, z_j)$$

- We want to guarantee small expected risk

$$R(M) = \mathop{\mathbb{E}}_{z, z' \sim P} [L(M, z, z')]$$

COMPUTATIONAL ISSUES IN THE LARGE-SCALE SETTING

- To summarize, we need an algorithm to solve

$$\min_{\boldsymbol{M} \in \mathbb{R}^{p \times p}} R_n(\boldsymbol{M}) = \frac{2}{n(n-1)} \sum_{i<j} L(\boldsymbol{M}, \boldsymbol{z}_i, \boldsymbol{z}_j)$$

- This is difficult when n or p are large
  - Merely computing $R_n(\boldsymbol{M})$ requires summing up over $O(n^2)$ terms
  - Number of parameters to learn is $O(p^2)$

Second part of the talk:
Large $n \to$ use sampling techniques
Large $p \to$ use greedy optimization

METRIC LEARNING ON LARGE DATASETS

NIPS '15 + JMLR '16 (WITH S. CLÉMENÇON, I. COLIN AND G. PAPA)

CONTRIBUTIONS

- We will consider a more general setting where the empirical risk is an average over d-tuples
- Metric learning is just a particular case
- We will show that it is possible to **drastically reduce the training complexity while preserving the accuracy of the solution**
- This will be achieved using a simple subsampling strategy

U-STATISTICS

- *U*-statistic of degree d with kernel H [Hoeffding, 1948]:

$$U_n(H) = \frac{1}{\binom{n}{d}} \sum_{1 \le i_1 < \cdots < i_d \le n} H(X_{i_1}, \ldots, X_{i_d})$$

  - $H : \mathcal{X}^d \to \mathbb{R}$ symmetric
  - In metric learning we have $d = 2$ or $d = 3$
  - Other applications: clustering, learning to rank

- U_n has minimum variance among all unbiased estimators of

$$U(H) = \mathop{\mathbb{E}}_{X_1, \ldots, X_d \sim P} [H(X_1, \ldots, X_d)]$$

- But for $d \ge 2$, not a sum of independent terms!

EMPIRICAL RISK MINIMIZATION WITH U-STATISTICS

- $\mathcal{G}$: class of learning rules (e.g., linear classifiers)
- $H_g : \mathcal{X}^d \to \mathbb{R}$: loss function associated with $g \in \mathcal{G}$
- True risk of rule $g \in \mathcal{G}$: $U(H_g) = \mathbb{E}_{X_1,\ldots,X_d \sim P}\left[H_g(X_1, \ldots, X_d)\right]$
- Empirical risk of $g \in \mathcal{G}$: $U_n(H_g) = \frac{1}{\binom{n}{d}} \sum_{1 \leq i_1 < \cdots < i_d \leq n} H_g(X_{i_1}, \ldots, X_{i_d})$
- Empirical Risk Minimization (ERM): choose rule

$$\hat{g} \in \arg\min_{g \in \mathcal{G}} U_n(H_g)$$

- Let $\hat{g} \in \arg\min_{g \in \mathcal{G}} U_n(H_g)$ the empirical risk minimizer
- Under suitable assumptions [Clémençon et al., 2008]

$$U(H_{\hat{g}}) - \inf_{g \in \mathcal{G}} U(H_g) = O_{\mathbb{P}}(1/\sqrt{n})$$

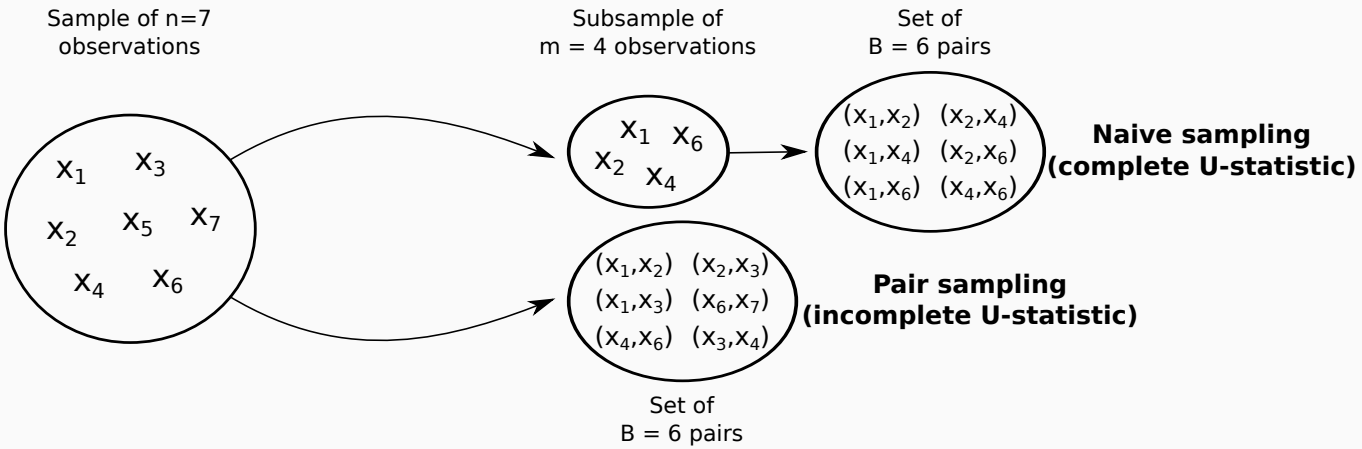
- **How to find $\hat{g}$ efficiently? $U_n(H_g)$ has $O(n^d)$ terms!**
  - Big Data problem even for relatively small datasets
- We will exploit the dependence structure of U_n

INCOMPLETE U-STATISTIC

- Let $\mathcal{D}_B$ be a set of cardinality B drawn by sampling with replacement from the set of d-tuples
- Approximate $U_n(H)$ by the incomplete version

$$\widetilde{U}_B(H_g) = \frac{1}{B} \sum_{I \in \mathcal{D}_B} H_g(X_{I_1}, \ldots, X_{I_d})$$

- This is different from a U-statistic based on a subsample



Theorem ([Clémençon et al., 2016])

Let $\mathcal{H} = \{\mathcal{H}_g : g \in \mathcal{G}\}$ be a VC major class of functions with VC dimension $V < +\infty$ and uniformly bounded by $M_{\mathcal{H}} < +\infty$.
For all $\eta > 0$, we have $\forall n, \forall B \geq 1$,

$$\mathbb{P}\left\{\sup_{H\in\mathcal{H}} \left|\widetilde{U}_B(H) - U_n(H)\right| > \eta\right\} \leq 2\left(1 + \binom{n}{d}\right)^V \times e^{-B\eta^2/M_{\mathcal{H}}^2}$$

- Prob. of large deviation decreases exponentially fast with B
- Main ingredients of the proof
  - Write $\widetilde{U}_B(H) - U_n(H)$ as an average of B independent variables
  - Sauer's lemma
  - Union bound and Hoeffding's inequality

Corollary ([Clémençon et al., 2016])

Let $\tilde{g}$ be an empirical risk minimizer of $\widetilde{U}_B$ over $\mathcal{H}$, and $\delta > 0$. Under the previous assumptions, with probability at least $1 - \delta$, we have:

$$U(H_{\tilde{g}}) - \inf_{g \in \mathcal{G}} U(H_g) \leq O\left(\sqrt{\frac{V \log(n) + \log(2/\delta)}{n}} + \sqrt{\frac{V \log(\binom{n}{d}) + \log(4/\delta)}{B}} \right)$$

- **Choosing $B = O(n)$ preserves the $O_{\mathbb{P}}(1/\sqrt{n})$ learning rate!**
- In contrast: complete U-statistic with $O(n)$ terms leads to much slower rate of $O_{\mathbb{P}}(\sqrt{1/n^{\frac{1}{d}}})$
- Other results (not covered here): fast rates, model selection

OPTIMIZATION OF U-STATISTICS

- $\Theta \subset \mathbb{R}^q$ parameter space
- $H : \mathcal{X}^d \times \Theta \to \mathbb{R}$ strongly convex and smooth in 2nd argument
- Reformulation of true risk

$$L(\theta) \overset{def}{=} U(H(\cdot;\, \theta))$$

- Reformulation of empirical risk

$$\widehat{L}_n(\theta) \overset{def}{=} U_n(H(\cdot;\, \theta))$$

- Reformulation of ERM problem

$$\min_{\theta \in \Theta} \widehat{L}_n(\theta)$$

- Initialize $\theta_0 \in \Theta$ and follow the iterations

$$\theta_{t+1} = \theta_t - \eta_t \nabla_\theta \widehat{L}_n(\theta_t), \qquad \eta_t \geq 0$$

- Gradient of $\widehat{L}_n(\theta)$ is

$$\nabla_\theta \widehat{L}_n(\theta) = \frac{1}{\binom{n}{d}} \sum_{1 \leq i_1 < \cdots < i_d \leq n} \nabla_\theta H(X_{i_1}, \ldots, X_{i_d};\ \theta)$$

- **Each gradient involves summing over $\binom{n}{d}$ terms!**

- Stochastic Gradient Descent (SGD): approximate gradient at each step using a random mini-batch of terms

Use incomplete U-statistic with B terms to estimate the gradient

Theorem ([Papa et al., 2015])

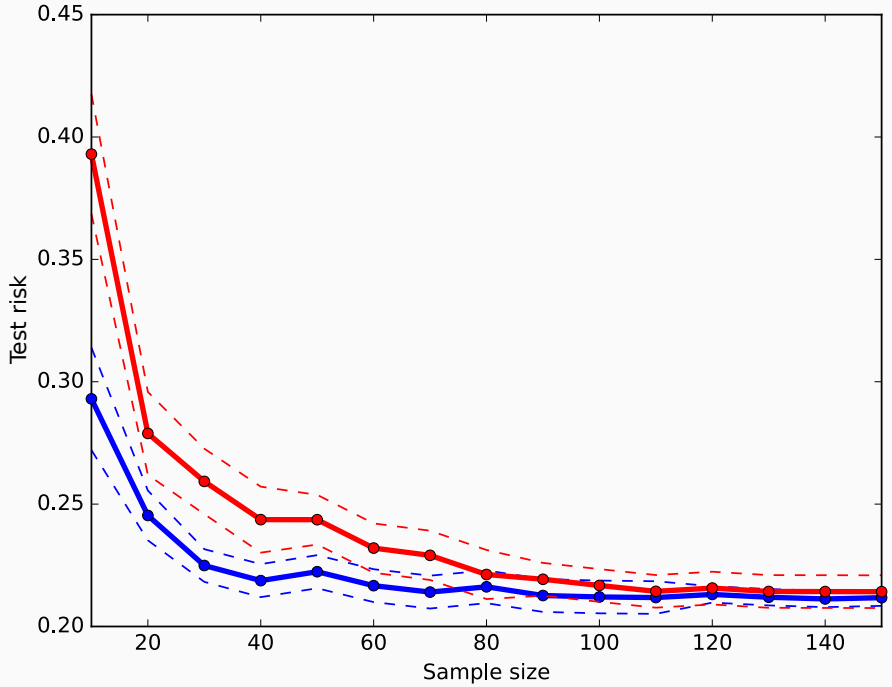
Let $\mathcal{H} = \{H(\cdot;\ \theta) : \theta \in \Theta\}$ be a VC major class of functions with VC dimension $V < +\infty$ and uniformly bounded by $M_{\mathcal{H}} < +\infty$. Let $\theta^* = \arg\min_{\theta\in\Theta} L(\theta)$. Under appropriate conditions on the step size, we have for $\forall n$:

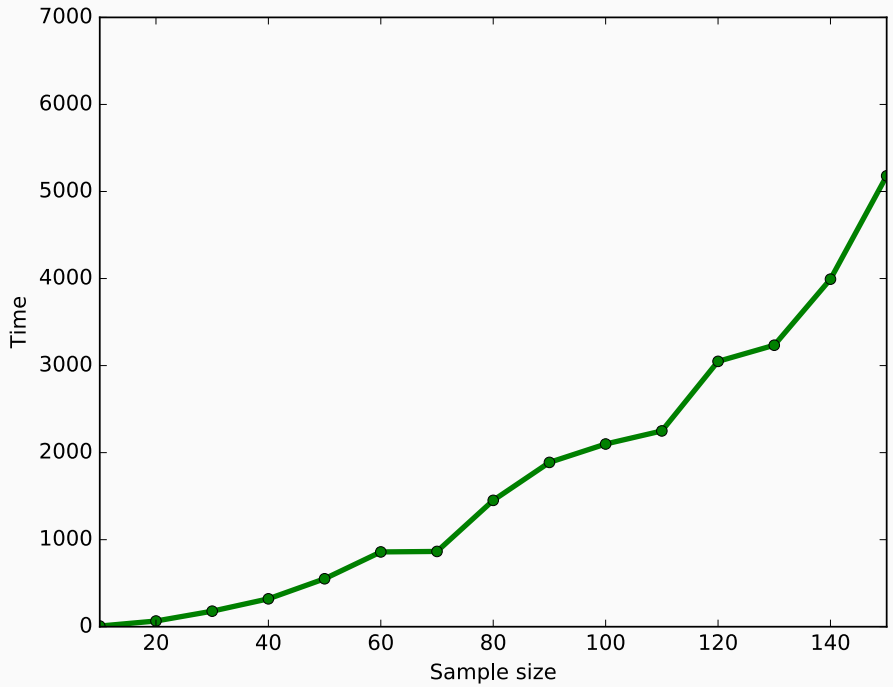
$$\mathbb{E}[|L(\theta_t) - L(\theta^*)|] \leqslant O\left(\frac{1}{Bt} + \mathcal{M}_{\mathcal{H}}\sqrt{\frac{V\log(n)}{n}}\right)$$

- Decomposition into optimization and generalization errors
- Set $B = \binom{n'}{d}$. Alternative: use complete U-statistic of size n'
  - Both estimates consist of B terms
  - But B is replaced by n' in the bound!

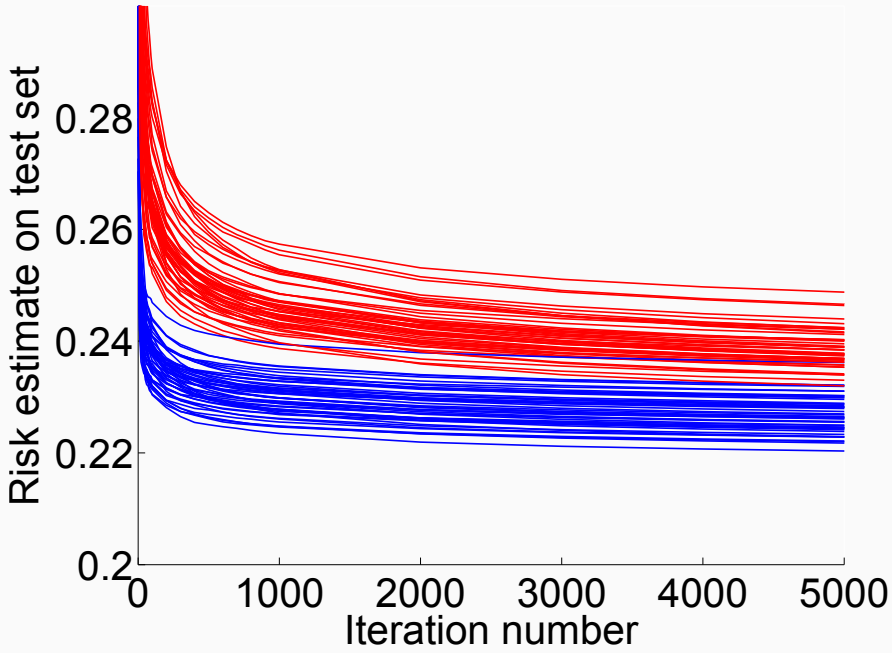
Approximate risk by complete or incomplete U-statistic

- Pairwise metric learning
- MNIST dataset: $n = 60,000 \rightarrow 2 \times 10^9$ pairs

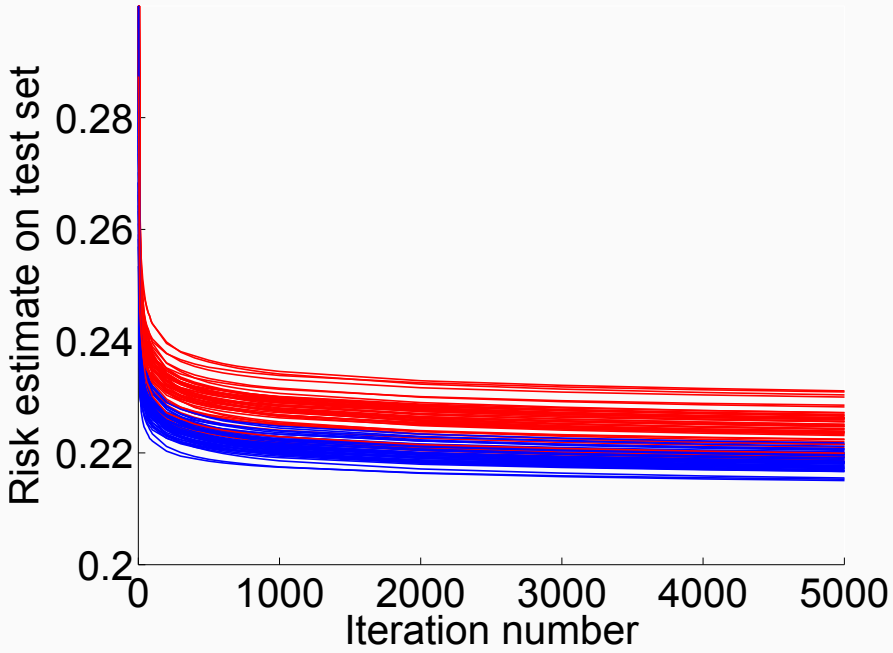




SGD: Approximate gradient with complete or incomplete U-statistic



B=10



B=55

METRIC LEARNING IN HIGH DIMENSIONS

AISTATS '15 (WITH K. LIU AND F. SHA)

CONTRIBUTIONS

- Assume data points are high-dimensional ($p > 10^4$) but s-sparse (on average) with $s \ll p$
  - Bags-of-words (text, image), bioinformatics, etc
- We want to avoid the pitfalls of existing algorithms
  - Training time in $O(p^2)$ to $O(p^3)$, memory in $O(p^2)$
  - Severe overfitting
- We will introduce a **greedy optimization algorithm**
  - Incorporate the best pair of features at each iteration
  - Use early stopping to control model complexity

BASIS SET

- We want to learn a bilinear similarity function $S_M(x, x') = x^T M x'$
- Given $\lambda > 0$, for any $i, j \in \{1, \dots, p\}$, $i \neq j$ we define

$$P_\lambda^{(ij)} = \begin{pmatrix} \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \lambda & \cdot & \lambda & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \lambda & \cdot & \lambda & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \end{pmatrix} \quad N_\lambda^{(ij)} = \begin{pmatrix} \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \lambda & \cdot & -\lambda & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & -\lambda & \cdot & \lambda & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \end{pmatrix}$$

$$\mathcal{B}_\lambda = \bigcup_{ij} \left\{ P_\lambda^{(ij)}, N_\lambda^{(ij)} \right\}$$

$$M \in \mathcal{D}_\lambda = \text{conv}(\mathcal{B}_\lambda)$$

- One basis involves only 2 features:

$$S_{P_\lambda^{(ij)}}(x, x') = \lambda(x_i x_i' + x_j x_j' + x_i x_j' + x_j x_i')$$

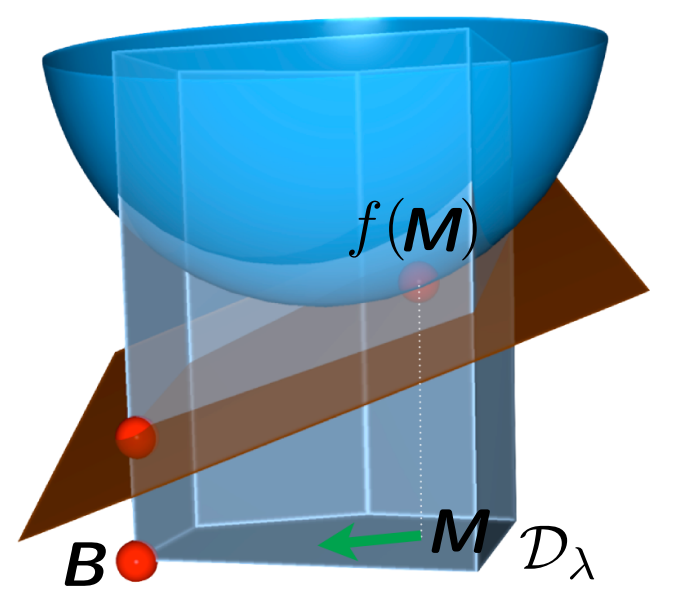
$$S_{N_\lambda^{(ij)}}(x, x') = \lambda(x_i x_i' + x_j x_j' - x_i x_j' - x_j x_i')$$

Optimization problem (L is the smoothed hinge loss)

$$\min_{M \in \mathbb{R}^{d \times d}} \quad f(M) = \frac{1}{|\mathcal{R}|} \sum_{(x_i, x_j, x_k) \in \mathcal{R}} L\left(1 - x_i^T M x_j + x_i^T M x_k\right)$$

$$\text{s.t.} \quad M \in \mathcal{D}_\lambda$$

- Use a Frank-Wolfe algorithm [Jaggi, 2013] to solve it



(figure from [Jaggi, 2013])

Let $M^{(0)} \in \mathcal{D}_\lambda$

for $k = 0, 1, \ldots$ **do**

$B^{(k)} = \arg\min_{B \in \mathcal{B}_\lambda} \left\langle B, \nabla f(M^{(k)}) \right\rangle$

$M^{(k+1)} = (1 - \gamma) M^{(k)} + \gamma B^{(k)}$

end for

Proposition ([Liu et al., 2015])

Let $L = \frac{1}{|\mathcal{R}|} \sum_{(x_i, x_j, x_k) \in \mathcal{R}} \|x_i (x_j - x_k)^T\|_F^2$. *At any iteration* $k \geq 1$, *the iterate* $M^{(k)} \in \mathcal{D}_\lambda$ *of the algorithm:*

- *has at most rank* $k+1$ *with* $4(k+1)$ *nonzero entries*
- *uses at most* $2(k+1)$ *distinct features*
- *satisfies* $f(M^{(k)}) - f(M^*) \leq 16L\lambda^2/(k+2)$

- An optimal basis can be found in $O(|\mathcal{R}|s^2)$ time and memory
- An approximately optimal basis can be found in $O(ms^2)$ with $m \ll C$ using a Monte Carlo approximation of the gradient
  - Or even $O(ms)$ using a heuristic (good results in practice)
- Storing $M^{(k)}$ requires only $O(k)$ memory
  - Or even the entire sequence $M^{(0)}, \ldots, M^{(k)}$ at the same cost

- Given $\{z_i = (x_i, y_i)\}_{i=1}^n$ drawn from P, let the empirical risk be

$$R_n(M) = \frac{1}{n(n-1)(n-2)} \sum_{i \neq j \neq k} \mathbb{I}\{y_i = y_j \neq y_k\} \cdot L(1 - x_i^T M x_j + x_i^T M x_k)$$

- The expected risk is

$$R(M) = \mathop{\mathbb{E}}_{z,z',z'' \sim P} \mathbb{I}\{y = y' \neq y''\} \cdot L(1 - x^T M x' + x^T M x'')$$

Theorem ([Liu et al., 2015])

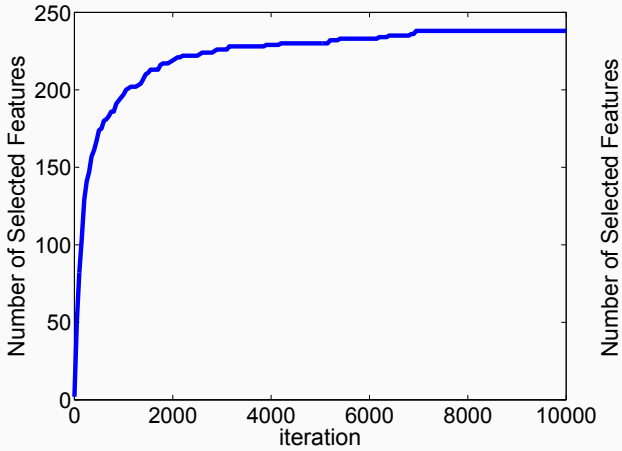
Let $k \geq 1$ and $M^{(k)}$ be the solution returned by the algorithm after k iterations. With high probability, we have

$$R(M^{(k)}) - \inf_{M \in \mathbb{R}^{d \times d}} R(M) \leq O\left(\frac{1}{k}\right) + O\left(\sqrt{\frac{\log k}{n}}\right)$$

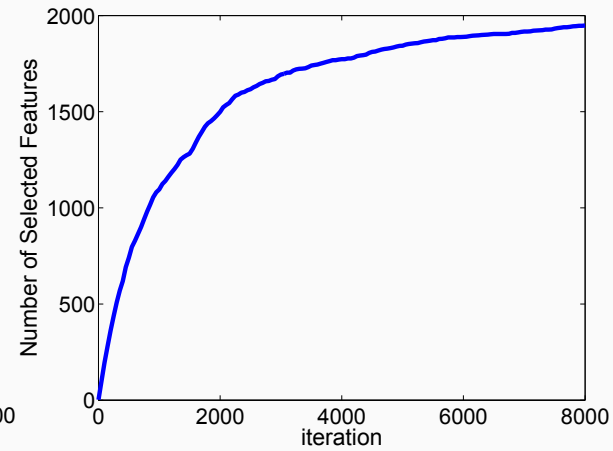
- The number of iterations rules an explicit trade-off between optimization error and model complexity

EXPERIMENTS

- In practice the model complexity stabilizes



(a) dexter dataset



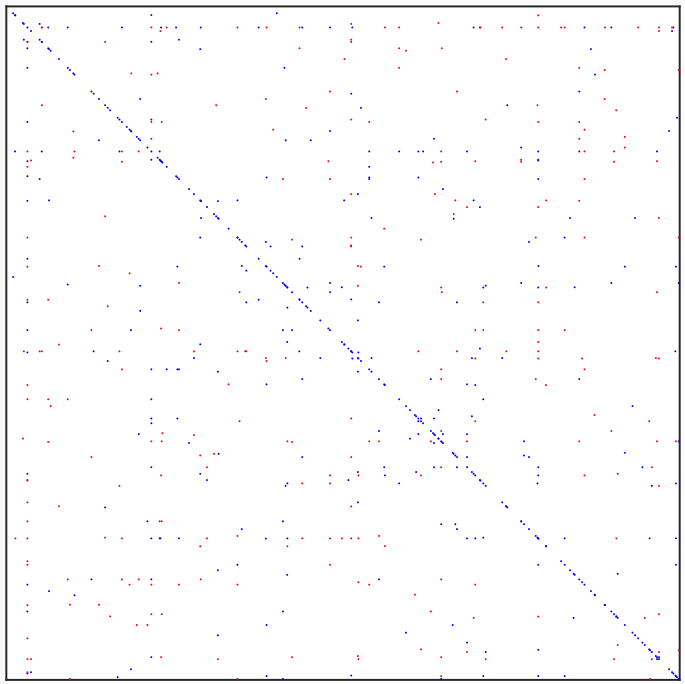
(b) rcv1_4 dataset

- Nearest neighbor error on datasets with p up to 10^5

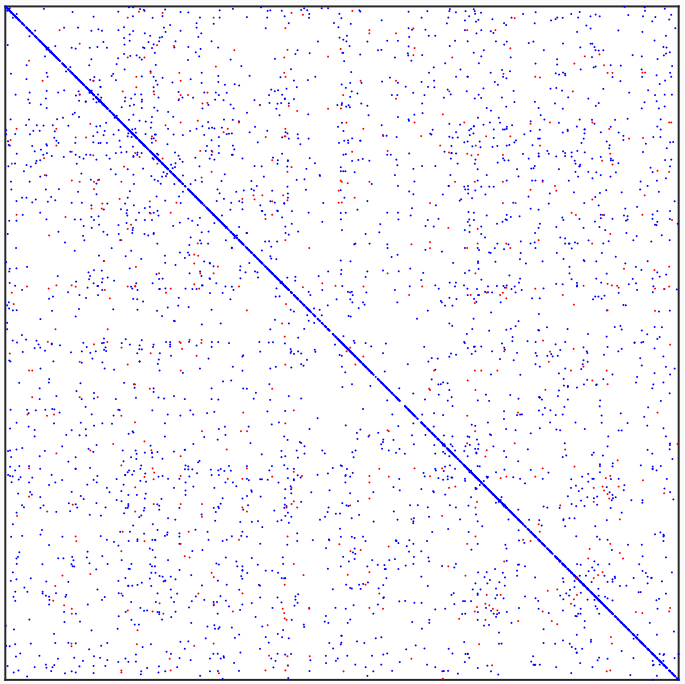
Datasets	Identity	RP+OASIS	PCA+OASIS	Diag-L_2	Diag-L_1	**Ours**
dexter	20.1	24.0	9.3	8.4	8.4	**6.5**
dorothea	9.3	11.4	9.9	6.8	6.6	**6.5**
rcv1_2	6.9	7.0	4.5	3.5	3.7	**3.4**
rcv1_4	11.2	10.6	6.1	6.2	7.2	**5.7**

EXPERIMENTS

- The learned matrices are very sparse
  - Easier to interpret
  - Fast similarity computation



(a) dexter ($20,000 \times 20,000$ matrix, 712 nonzeros)



(b) rcv1_4 ($29,992 \times 29,992$ matrix, 5263 nonzeros)

TAKE-HOME MESSAGE

- Distance and similarity functions are essential components of many systems and learning algorithms
- Metric learning can be used to automatically learn a good measure from data
  - For various types of metrics
  - In many learning scenarios
  - With statistical learning guarantees
- It is possible to **scale-up metric learning algorithms without sacrificing accuracy**
  - Large datasets $\rightarrow$ random sampling
  - High-dimensional features $\rightarrow$ greedy optimization

Thank you!

Questions?

REFERENCES I

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

[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.

References II

[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., 2016] Clémençon, S., Bellet, A., and Colin, I. (2016).
Scaling-up Empirical Risk Minimization: Optimization of Incomplete U-statistics.
Journal of Machine Learning Research, to appear.

[Clémençon et al., 2008] Clémençon, S., Lugosi, G., and Vayatis, N. (2008).
Ranking and Empirical Minimization of U-statistics.
Annals of Statistics, 36(2):844–874.

[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.

References III

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

[Guillaumin et al., 2009] Guillaumin, M., Verbeek, J. J., and Schmid, C. (2009).
Is that you? Metric learning approaches for face identification.
In ICCV, pages 498–505.

[Hoeffding, 1948] Hoeffding, W. (1948).
A Class of Statistics with Asymptotically Normal Distribution.
The Annals of Mathematical Statistics, 19(3):293–325.

[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.

References IV

[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.

[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.

[Papa et al., 2015] Papa, G., Bellet, A., and Clémençon, S. (2015).
SGD Algorithms based on Incomplete U-statistics: Large-Scale Minimization of Empirical Risk.
In NIPS.

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

References V

[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.

[Wang et al., 2012] Wang, J., Woznica, A., and Kalousis, A. (2012).
Parametric Local Metric Learning for Nearest Neighbor Classification.
In NIPS, pages 1610–1618.

REFERENCES VI

[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.

[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.

[Zhu et al., 2015] Zhu, X., Lei, Z., Yan, J., Yi, D., and Li, S. Z. (2015).
High-fidelity pose and expression normalization for face recognition in the wild.
In CVPR, pages 787–796.