

Metric Learning: from Algorithms to Theoretical Guarantees

M. Sebban

LABORATOIRE HUBERT CURIEN, UMR CNRS 5516
University of Jean Monnet Saint-Étienne (France)

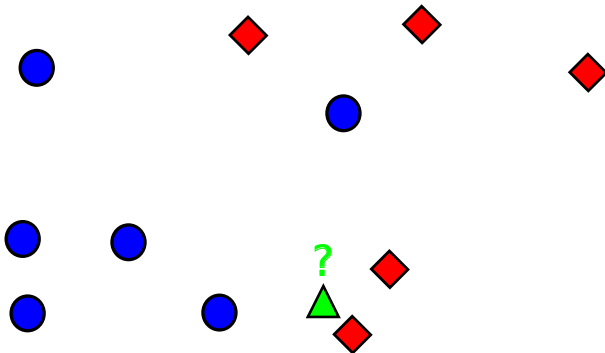
EPAT 2014, Carry-le-Rouet, June, 2014

- 1 Intuition behind Metric Learning
- 2 State of the Art
 - Mahalanobis Distance Learning
 - Nonlinear Metric Learning
 - Online Metric Learning
 - Algorithmic and Theoretical Limitations
- 3 Theoretical Guarantees in Metric learning
 - Generalization guarantees: Balcan et al. framework (2008)
 - Consistency Guarantees: Uniform Stability
- 4 Experiments

Importance of Metrics

Pairwise metric

The notion of **metric** plays an important role in many domains such as *classification, regression, clustering, ranking, etc.*



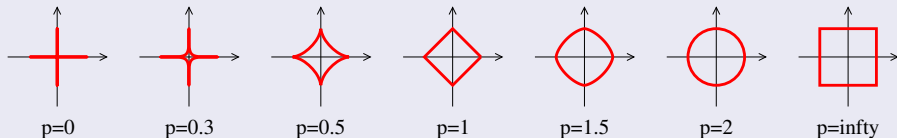
Minkowski distances: family of distances induced by ℓ_p norms

$$d_p(\mathbf{x}, \mathbf{x}') = \|\mathbf{x} - \mathbf{x}'\|_p = \left(\sum_{i=1}^d |x_i - x'_i|^p \right)^{1/p}$$

- For $p = 1$, the **Manhattan distance** $d_{man}(\mathbf{x}, \mathbf{x}') = \sum_{i=1}^d |x_i - x'_i|$.
- For $p = 2$, the “ordinary” **Euclidean distance**:

$$d_{euc}(\mathbf{x}, \mathbf{x}') = \left(\sum_{i=1}^d |x_i - x'_i|^2 \right)^{1/2} = \sqrt{(\mathbf{x} - \mathbf{x}')^T (\mathbf{x} - \mathbf{x}')}$$

- For $p \rightarrow \infty$, the **Chebyshev distance** $d_{che}(\mathbf{x}, \mathbf{x}') = \max_i |x_i - x'_i|$.



Key question

How to choose the right metric?

The notion of good metric is problem-dependent

Each problem has its own notion of similarity, which is often badly captured by standard metrics.

How to discriminate between humans and dogs?



Predicted label?

Limitations of standard metrics



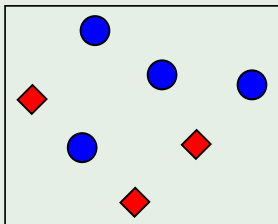
It's not what it looks Like...

Metric learning

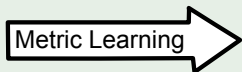
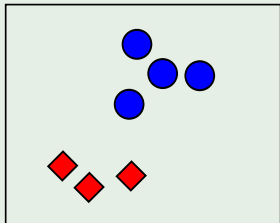
Adapt the metric to the problem of interest

Solution: learn the metric from data

Basic idea: learn a metric that assigns small (resp. large) distance to pairs of examples that are semantically similar (resp. dissimilar).



Metric Learning

A large white arrow with a black outline pointing from the left box to the right box, labeled "Metric Learning".

It typically **induces a change of representation space** which satisfies constraints.

“Learnable” Metrics

The Mahalanobis distance

$\forall \mathbf{x}, \mathbf{x}' \in \mathbb{R}^d$, the Mahalanobis distance is defined as follows:

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

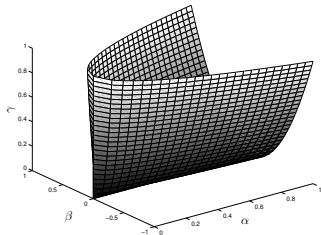
where $\mathbf{M} \in \mathbb{R}^{d \times d}$ is a symmetric PSD matrix ($\mathbf{M} \succeq 0$).

The original term refers to the case where \mathbf{x} and \mathbf{x}' are random vectors from the same distribution with covariance matrix Σ , with $\mathbf{M} = \Sigma^{-1}$.

PSD matrices

Definition (PSD matrix)

A matrix $\mathbf{M} \in \mathbb{R}^{d \times d}$ is positive semi-definite (PSD) if all its eigenvalues are nonnegative. The cone of symmetric PSD $d \times d$ real-valued matrices is denoted by \mathbb{S}_+^d . As a shortcut for $\mathbf{M} \in \mathbb{S}_+^d$ we use $\mathbf{M} \succeq 0$.



Useful properties

If $\mathbf{M} \succeq 0$, then

- $\mathbf{x}^T \mathbf{M} \mathbf{x} \geq 0 \quad \forall \mathbf{x}$ (as a linear operator, can be seen as nonnegative scaling).
- $\mathbf{M} = \mathbf{L}^T \mathbf{L}$ for some matrix \mathbf{L} .

Mahalanobis distance learning

Using the decomposition $\mathbf{M} = \mathbf{L}^T \mathbf{L}$, where $\mathbf{L} \in \mathbb{R}^{k \times d}$, where k is the rank of \mathbf{M} , one can rewrite $d_{\mathbf{M}}(\mathbf{x}, \mathbf{x}')$.

$$\begin{aligned} d_{\mathbf{M}}(\mathbf{x}, \mathbf{x}') &= \sqrt{(\mathbf{x} - \mathbf{x}')^T \mathbf{L}^T \mathbf{L} (\mathbf{x} - \mathbf{x}')} \\ &= \sqrt{(\mathbf{L}\mathbf{x} - \mathbf{L}\mathbf{x}')^T (\mathbf{L}\mathbf{x} - \mathbf{L}\mathbf{x}')}. \end{aligned}$$

Mahalanobis distance learning = Learning a linear projection

If \mathbf{M} is learned, a Mahalanobis distance implicitly corresponds to **computing the Euclidean distance after a learned linear projection** of the data (learned under constraints) by \mathbf{L} in a k -dimensional space.

Metric learning in a nutshell: Basic setup

Learning from side information

- Must-link / cannot-link constraints:

$$\mathcal{S} = \{(x_i, x_j) : x_i \text{ and } x_j \text{ should be similar}\},$$

$$\mathcal{D} = \{(x_i, x_j) : x_i \text{ and } x_j \text{ should be dissimilar}\}.$$

- Relative constraints:

$$\mathcal{R} = \{(x_i, x_j, x_k) : x_i \text{ should be more similar to } x_j \text{ than to } x_k\}.$$

Metric learning in a nutshell

General formulation

Given a metric, find its parameters \mathbf{M}^* as

$$\mathbf{M}^* = \arg \min_{\mathbf{M} \succeq 0} [\ell(\mathbf{M}, \mathcal{S}, \mathcal{D}, \mathcal{R}) + \lambda R(\mathbf{M})],$$

where

Metric learning in a nutshell

General formulation

Given a metric, find its parameters \mathbf{M}^* as

$$\mathbf{M}^* = \arg \min_{\mathbf{M} \succeq 0} [\ell(\mathbf{M}, \mathcal{S}, \mathcal{D}, \mathcal{R}) + \lambda R(\mathbf{M})],$$

where

- $\ell(\mathbf{M}, \mathcal{S}, \mathcal{D}, \mathcal{R})$ is a loss function that penalizes violated constraints,

Metric learning in a nutshell

General formulation

Given a metric, find its parameters \mathbf{M}^* as

$$\mathbf{M}^* = \arg \min_{\mathbf{M} \succeq 0} [\ell(\mathbf{M}, \mathcal{S}, \mathcal{D}, \mathcal{R}) + \lambda R(\mathbf{M})],$$

where

- $\ell(\mathbf{M}, \mathcal{S}, \mathcal{D}, \mathcal{R})$ is a loss function that penalizes violated constraints,
- $R(\mathbf{M})$ is some regularizer on \mathbf{M} ,

Metric learning in a nutshell

General formulation

Given a metric, find its parameters \mathbf{M}^* as

$$\mathbf{M}^* = \arg \min_{\mathbf{M} \succeq 0} [\ell(\mathbf{M}, \mathcal{S}, \mathcal{D}, \mathcal{R}) + \lambda R(\mathbf{M})],$$

where

- $\ell(\mathbf{M}, \mathcal{S}, \mathcal{D}, \mathcal{R})$ is a loss function that penalizes violated constraints,
- $R(\mathbf{M})$ is some regularizer on \mathbf{M} ,
- and $\lambda \geq 0$ is the regularization parameter.

Metric learning in a nutshell

General formulation

Given a metric, find its parameters \mathbf{M}^* as

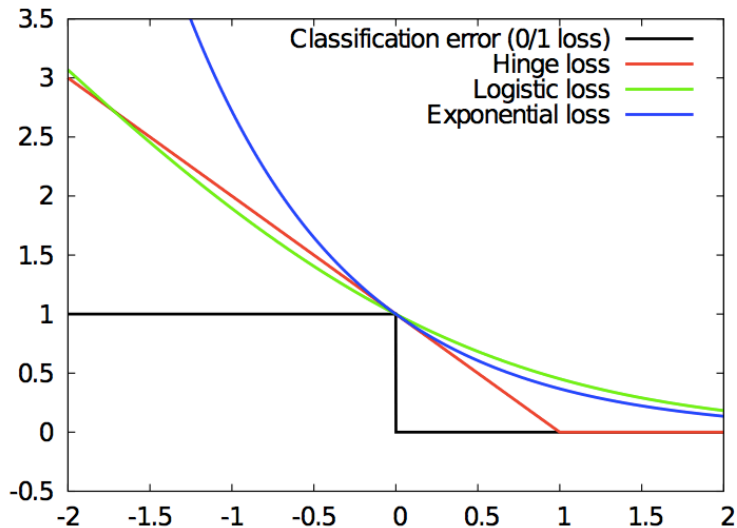
$$\mathbf{M}^* = \arg \min_{\mathbf{M} \geq 0} [\ell(\mathbf{M}, \mathcal{S}, \mathcal{D}, \mathcal{R}) + \lambda R(\mathbf{M})],$$

where

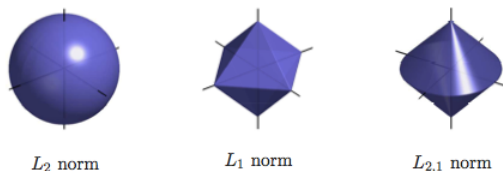
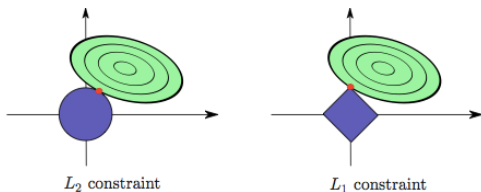
- $\ell(\mathbf{M}, \mathcal{S}, \mathcal{D}, \mathcal{R})$ is a loss function that penalizes violated constraints,
- $R(\mathbf{M})$ is some regularizer on \mathbf{M} ,
- and $\lambda \geq 0$ is the regularization parameter.

State of the art methods essentially differ by the choice of **constraints**, **loss function** and **regularizer** on \mathbf{M} .

Loss functions for binary classification



Regularization



- The mixed $L_{2,1}$ norm on matrix \mathbf{M} is defined as $\|\mathbf{M}\|_{2,1} = \sum_{i=1}^d \|\mathbf{M}_i\|_2$.
- The nuclear norm (also called trace norm): \mathbf{M} : $\|\mathbf{M}\|_* = \text{tr}(\mathbf{M})$.

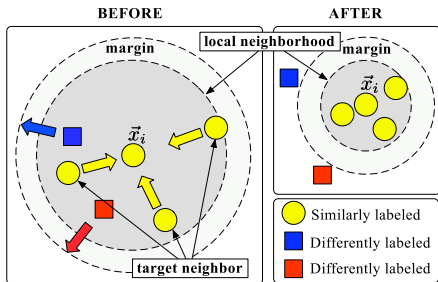
LMNN (Weinberger et al. 2005)

Main Idea

Define constraints tailored to k -NN in a local way: the k nearest neighbors should be of same class (“**target neighbors**”), while examples of different classes should be kept away (“**impostors**”):

$$\mathcal{S} = \{(\mathbf{x}_i, \mathbf{x}_j) : y_i = y_j \text{ and } \mathbf{x}_j \text{ belongs to the } k\text{-neighborhood of } \mathbf{x}_i\},$$

$$\mathcal{R} = \{(\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k) : (\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{S}, y_i \neq y_k\}.$$



LMNN (Weinberger et al. 2005)

Formulation

$$\min_{\mathbf{M} \succeq 0} (1 - \mu) \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{S}} d_{\mathbf{M}}^2(\mathbf{x}_i, \mathbf{x}_j) + \mu \sum_{i,j,k} \xi_{ijk}$$

$$\text{s.t. } d_{\mathbf{M}}^2(\mathbf{x}_i, \mathbf{x}_k) - d_{\mathbf{M}}^2(\mathbf{x}_i, \mathbf{x}_j) \geq 1 - \xi_{ijk} \quad \forall (\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k) \in \mathcal{R},$$

where μ controls the “pull/push” trade-off.

Remarks

- **Advantages:** Convex, with a solver based on working set and subgradient descent. Can deal with millions of constraints and very popular in practice.
- **Drawback:** Subject to overfitting in high dimension.

ITML (Davis et al. 2007)

Information-Theoretical Metric Learning (ITML) introduces LogDet divergence regularization. This Bregman divergence on PSD matrices is defined as:

$$D_{ld}(\mathbf{M}, \mathbf{M}_0) = \text{trace}(\mathbf{M}\mathbf{M}_0^{-1}) - \log \det(\mathbf{M}\mathbf{M}_0^{-1}) - d.$$

where d is the dimension of the input space and \mathbf{M}_0 is some PSD matrix we want to remain close to. ITML is formulated as follows:

$$\begin{aligned} \min_{\mathbf{M} \succeq 0} \quad & D_{ld}(\mathbf{M}, \mathbf{M}_0) + \gamma \sum_{i,j,k} \xi_{ij} \\ \text{s.t.} \quad & d_{\mathbf{M}}^2(\mathbf{x}_i, \mathbf{x}_j) \leq u + \xi_{ij} \quad \forall (\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{S} \\ & d_{\mathbf{M}}^2(\mathbf{x}_i, \mathbf{x}_j) \geq v - \xi_{ij} \quad \forall (\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{D}, \end{aligned}$$

The LogDet divergence is finite iff \mathbf{M} is PSD.

Nonlinear metric learning

The big picture

Nonlinear metric learning: 3 approaches

- 1 Kernelization of linear methods.
- 2 Learning a nonlinear metric.
- 3 Learning several local linear metrics.

Nonlinear metric learning

Kernelization of linear methods

- Some algorithms have been shown to be kernelizable, but in general this is not trivial: a new formulation of the problem has to be derived, where interface to the data is **limited to inner products**, and sometimes a different implementation is necessary.
- When the number of training examples n is large, **learning n^2 parameters may be intractable**.

A solution: KPCA trick (Chatpatanasiri et al., 2010)

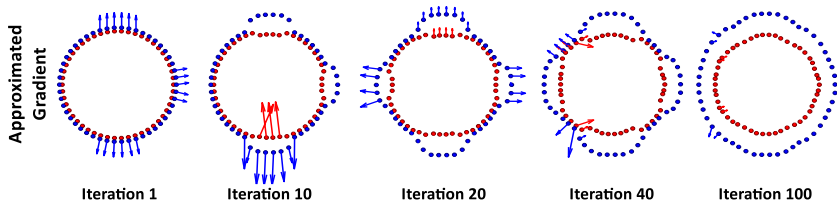
- Use KPCA (PCA in kernel space) to get a nonlinear but low-dimensional projection of the data.
- Then use unchanged algorithm!

Nonlinear metric learning

Learning a nonlinear metric: GB-LMNN (Kedem et al. 2012)

Main idea

- Learn a nonlinear mapping ϕ to optimize the Euclidean distance $d_\phi(\mathbf{x}, \mathbf{x}') = \|\phi(\mathbf{x}) - \phi(\mathbf{x}')\|_2$ in the transformed space.
- $\phi = \phi_0 + \alpha \sum_{t=1}^T h_t$, where ϕ_0 is the mapping learned by linear LMNN, and h_1, \dots, h_T are **gradient boosted regression trees**.
- Intuitively, each tree divides the space into 2^p regions, and instances falling in the **same region are translated by the same vector**.



Nonlinear metric learning

Local metric learning

Motivation

- Simple linear metrics perform well locally.
- Since everything is linear, can keep formulation convex.

M^2 -LMNN (Weinberger and Saul 2008,2009)

- Partition in C clusters (in a supervised or unsupervised way).
- C Mahalanobis distances are learned.

Pitfalls

- How to split the space?
- How to avoid a blow-up in number of parameters to learn, and avoid overfitting?
- How to obtain a proper (continuous) global metric?
- ...

Online learning

Warning

If the number of training constraints is **very large**, previous algorithms become huge, possibly **intractable optimization problems**.

One solution: online learning

- In **online metric learning**, the algorithm receives **training pairs** one at a time and **updates the current hypothesis** at each step.
- Often come with guarantees in the form of **regret bounds** stating that the accumulated loss suffered along the way is **not much worse than that of the best hypothesis chosen in hindsight**.

Online learning

Regret bound

A **regret bound** has the following general form:

$$\sum_{t=1}^T \ell(h_t, z_t) - \sum_{t=1}^T \ell(h^*, z_t) \leq O(T),$$

where T is the number of steps, h_t is the hypothesis at time t and h^* is the best batch hypothesis.

Mahalanobis distance learning

LEGO (Jain et al. 2008)

Formulation

At each step, receive $(\mathbf{x}_t, \mathbf{x}'_t, y_t)$ where y_t is the target distance between \mathbf{x}_t and \mathbf{x}'_t , and update as follows:

$$\mathbf{M}^{t+1} = \arg \min_{\mathbf{M} \succeq 0} D_{ld}(\mathbf{M}, \mathbf{M}^t) + \lambda \ell(\mathbf{M}, \mathbf{x}_t, \mathbf{x}'_t, y_t),$$

where ℓ is a loss function (square loss, hinge loss...).

Remarks

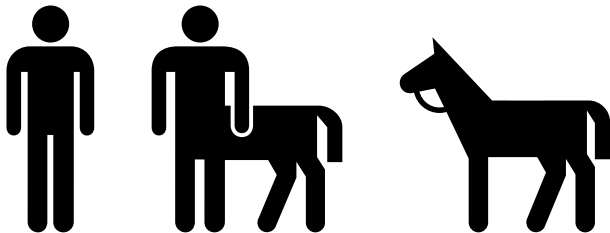
- It turns out that the above update has a closed-form solution which maintains $\mathbf{M} \succeq 0$ automatically.
- Can derive a regret bound.

Limitations of the state of the art ML algorithms

Algorithmic limitations

Drawbacks of Mahalanobis distance learning:

- Maintaining $\mathbf{M} \succeq 0$ is often costly, especially in high dimensions.
- Objects must have same dimension.
- Distance properties can be useful (e.g., for fast neighbor search), but restrictive. Evidence that our notion of (visual) similarity violates the triangle inequality (example below).



Similarity learning

Cosine similarity

The cosine similarity (widely used in data mining) measures the cosine of the angle between two instances, and can be computed as

$$K_{\cos}(\mathbf{x}, \mathbf{x}') = \frac{\mathbf{x}^T \mathbf{x}'}{\|\mathbf{x}\|_2 \|\mathbf{x}'\|_2}.$$

Bilinear similarity

The bilinear similarity is related to the cosine but does not include normalization and is parameterized by a matrix \mathbf{M} :

$$K_{\mathbf{M}}(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T \mathbf{M} \mathbf{x}',$$

where \mathbf{M} is not required to be PSD nor symmetric.

Limitations of the state of the art ML algorithms

Theoretical limitations

Establishing theoretical guarantees for the metric learning algorithms has so far received very little attention. However, we may be interested in theoretical results on:

- the **algorithm** which makes use of it (“plug and hope” strategy):
generalization guarantees,
- and on the **learned metric** d_M itself (optimized w.r.t. training data):
consistency guarantees.

Bellet, A., Habrard, A., and Sebban, M. *Similarity Learning for Provably Accurate Sparse Linear Classification*, ICML 2012.

Generalization Guarantees

Deriving generalization guarantees

Generalization guarantees for the classifier using the metric: (ϵ, γ, τ) -goodness

Definition (Balcan et al., 2008)

A similarity function $K \in [-1, 1]$ is (ϵ, γ, τ) -**good** w.r.t. to an indicator function $R(x)$ defining a set of “reasonable points” if:

- 1 A $1 - \epsilon$ probability mass of examples (x, y) satisfy:

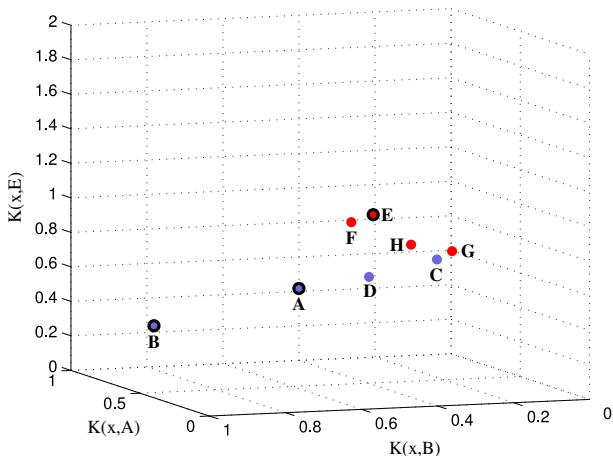
$$\mathbb{E}_{(x', y') \sim P} [yy'K(x, x') | R(x')] \geq \gamma.$$

- 2 $\Pr_{x'}[R(x')] \geq \tau.$ $\epsilon, \gamma, \tau \in [0, 1]$

- The first condition requires that a $1 - \epsilon$ proportion of examples x are **on average** more similar to reasonable examples of the same class than to reasonable examples of the opposite class by a margin γ .
- The second condition means that at least a τ proportion of the examples are reasonable.

Strategy

If R is known, use K to map the examples to the space ϕ of “the similarity scores with the reasonable points” (**similarity map**).

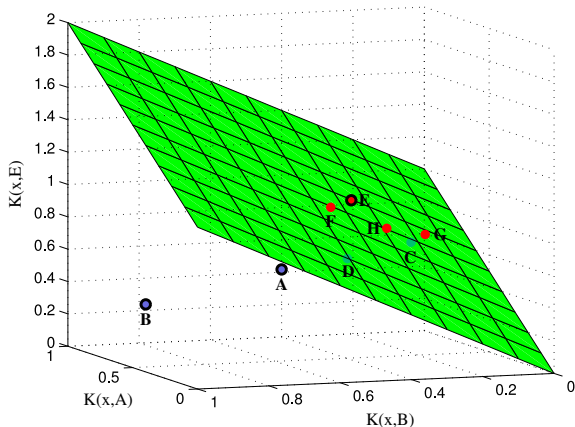


Deriving generalization guarantees

Generalization guarantees for the classifier using the metric: (ϵ, γ, τ) -goodness

A trivial linear classifier

By definition of (ϵ, γ, τ) -goodness, we have a linear classifier in ϕ that achieves true risk ϵ at margin γ .



Deriving generalization guarantees

Generalization guarantees for the classifier using the metric: (ϵ, γ, τ) -goodness

Theorem (Balcan et al., 2008)

*If R is unknown, given K is (ϵ, γ, τ) -good and enough points to create a similarity map, with high probability **there exists a linear separator α** that has true risk ϵ at margin γ .*

Question

Can we find this linear classifier in an efficient way?

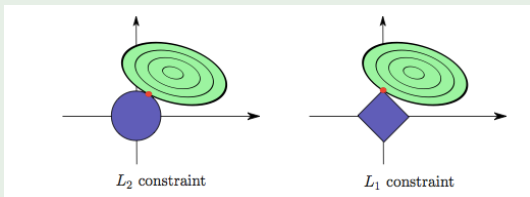
Deriving generalization guarantees

Answer

Basically, yes: solve a Linear Program with 1-norm regularization. We get a sparse linear classifier.

$$\min_{\alpha} \sum_{i=1}^n \left[1 - \sum_{j=1}^n \alpha_j y_i K(x_i, x_j) \right]_+ + \lambda \|\alpha\|_1$$

L_1 norm induces sparsity



SLLC (Bellet et al. 2012)

The performance of the linear classifier theoretically depends on how well the similarity function satisfies the definition of goodness.

$$\mathbb{E}_{(x', y') \sim P} [yy' K(x, x') | R(x')] \geq \gamma.$$

SLLC optimizes the empirical goodness of K over the training set.

Formulation of SLLC

$$\min_{\mathbf{M} \in \mathbb{R}^{d \times d}} \frac{1}{n} \sum_{i=1}^n \left[1 - y_i \frac{1}{\gamma |R|} \sum_{\mathbf{x}_j \in R} y_j K_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j) \right]_+ + \beta \|\mathbf{M}\|_{\mathcal{F}}^2,$$

where

$$K_{\mathbf{M}}(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T \mathbf{M} \mathbf{x}'.$$

SLLC (Bellet et al. 2012)

Properties of SLLC

SLLC has a number of desirable properties:

- SLLC optimizes a **link between the quality of the metric and the quality of the linear classifier**.
- Unlike classic algorithms, which rely on pair or triplet-based constraints, SLLC satisfies constraints that are **defined over an average of similarity scores**.
- SLLC has **only one constraint per training example**, instead of one for each pair or triplet.
- We can derive **consistency guarantees** on the learned similarity.

Consistency Guarantees

Deriving consistency guarantees

Consistency guarantees for the learned metric: uniform stability

Definition (Uniform stability for metric learning)

A learning algorithm \mathcal{A} has a **uniform stability** in κ/n , where $\kappa > 0$, if

$$\forall(T, \mathbf{x}), \forall i, \sup_{\mathbf{x}_1, \mathbf{x}_2} |\ell(\mathcal{A}_T, \mathbf{x}_1, \mathbf{x}_2) - \ell(\mathcal{A}_{T^{i,\mathbf{x}}}, \mathbf{x}_1, \mathbf{x}_2)| \leq \frac{\kappa}{n},$$

where \mathcal{A}_T is the metric learned by \mathcal{A} from T , and $T^{i,\mathbf{x}}$ is the set obtained by replacing $\mathbf{x}_i \in T$ by a new example \mathbf{x} .

Theorem (Uniform stability bound)

For any algorithm \mathcal{A} with uniform stability κ/n , with probability $1 - \delta$ over the random sample T , we have:

$$R^\ell(\mathcal{A}_T) \leq R_T^\ell(\mathcal{A}_T) + \frac{2\kappa}{n} + (2\kappa + B) \sqrt{\frac{\ln(2/\delta)}{2n}},$$

where B is a problem-dependent constant.

Stability of SLLC

Formulation of SLLC

$$\min_{\mathbf{M} \in \mathbb{R}^{d \times d}} \frac{1}{n} \sum_{i=1}^n \left[1 - y_i \frac{1}{\gamma |R|} \sum_{\mathbf{x}_j \in R} y_j K_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j) \right]_+ + \beta \|\mathbf{M}\|_{\mathcal{F}}^2,$$

where

$$K_{\mathbf{M}}(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T \mathbf{M} \mathbf{x}'.$$

Lemma

Let n and $|R|$ be the number of training examples and reasonable points respectively, $|R| = \hat{\tau} n$ with $\hat{\tau} \in]0, 1]$. SLLC has a uniform stability in $\frac{\kappa}{n}$ with

$$\kappa = \frac{1}{\gamma} \left(\frac{1}{\beta \gamma} + \frac{2}{\hat{\tau}} \right),$$

where β is the regularization parameter and γ the margin.

Consistency guarantees of SLLC

Theorem

Let $\gamma > 0$, $\delta > 0$ and $n_{\mathcal{T}} > 1$. With probability at least $1 - \delta$, for any model \mathbf{M} learned with SLLC, we have:

$$\epsilon \leq \hat{\epsilon} + \frac{1}{n} \left(\frac{1}{\gamma} \left(\frac{1}{\beta\gamma} + \frac{2}{\hat{\tau}} \right) \right) + \left(\frac{1}{\gamma} \left(\frac{1}{\beta\gamma} + \frac{2}{\hat{\tau}} \right) + 1 \right) \sqrt{\frac{\ln 1/\delta}{2n}}$$

where:

- $\hat{\epsilon} = \frac{1}{n} \sum_{i=1}^n \left[1 - y_i \frac{1}{\gamma|R|} \sum_{k=1}^{|R|} y_k K_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_k) \right]_+$.
- $\epsilon = \mathbb{E}_{(\mathbf{x}_i, y_i) \sim P} \left[1 - y_i \frac{1}{\gamma|R|} \sum_{k=1}^{|R|} y_k K_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_k) \right]_+$.

Experimental Results

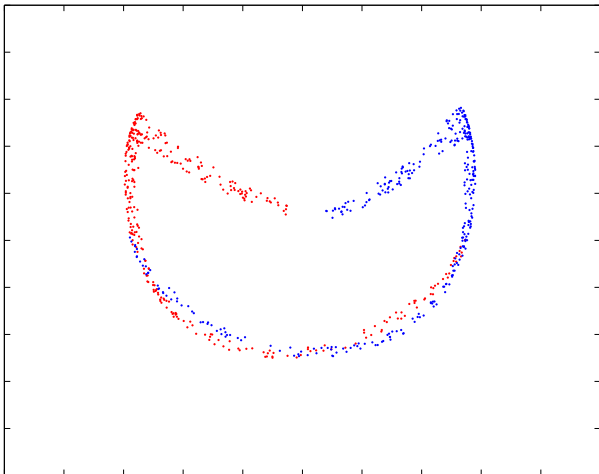
Comparison between a kernelized version (using a KPCA) of SLLC and:

- Standard bilinear similarity.
- LMNN
- LMNN KPCA
- ITML
- ITML KPCA

Experiments with linear classifiers

Dataset	Breast	Iono.	Rings	Pima	Splice	Svmguide1	Cod-RNA
K_I	96.57	89.81	100.00	75.62	83.86	96.95	95.91
	20.39	52.93	18.20	25.93	362	64	557
SLLC	96.90	93.25	100.00	75.94	87.36	96.55	94.08
	1.00	1.00	1.00	1.00	1	8	1
LMNN	96.81	90.21	100.00	75.15	85.61	95.80	88.40
	9.98	13.30	18.04	69.71	315	157	61
LMNN KPCA	96.01	86.12	100.00	74.92	86.85	96.53	95.15
	8.46	9.96	8.73	22.20	156	82	591
ITML	96.80	92.09	100.00	75.25	81.47	96.70	95.06
	9.79	9.51	17.85	56.22	377	49	164
ITML KPCA	96.23	93.05	100.00	75.25	85.29	96.55	95.14
	17.17	18.01	15.21	16.40	287	89	206

Rings



Conclusion and Perspectives: What next?

- **Scalability with both n and d**
 - Optimization over the manifold of low-rank matrices [Cheng, 2013, Shalit et al., 2012].
 - Combination of simple classifiers [Kedem et al., 2012, Xiong et al., 2012].

Conclusion and Perspectives: What next?

- **Scalability with both n and d**
 - Optimization over the manifold of low-rank matrices [Cheng, 2013, Shalit et al., 2012].
 - Combination of simple classifiers [Kedem et al., 2012, Xiong et al., 2012].
- **More theoretical understanding**
 - So far, only results for linear classification have been obtained [Bellet et al., 2012b, Guo and Ying, 2014].
 - What about kNN classification, clustering or information retrieval?

Conclusion and Perspectives: What next?

- **Scalability with both n and d**
 - Optimization over the manifold of low-rank matrices [Cheng, 2013, Shalit et al., 2012].
 - Combination of simple classifiers [Kedem et al., 2012, Xiong et al., 2012].
- **More theoretical understanding**
 - So far, only results for linear classification have been obtained [Bellet et al., 2012b, Guo and Ying, 2014].
 - What about kNN classification, clustering or information retrieval?
- **Unsupervised metric learning**
 - What is a good metric for clustering: preliminary work on this question [Balcan et al., 2008b, Lajugie et al., 2014].

Conclusion and Perspectives: What next?

- **Scalability with both n and d**
 - Optimization over the manifold of low-rank matrices [Cheng, 2013, Shalit et al., 2012].
 - Combination of simple classifiers [Kedem et al., 2012, Xiong et al., 2012].
- **More theoretical understanding**
 - So far, only results for linear classification have been obtained [Bellet et al., 2012b, Guo and Ying, 2014].
 - What about kNN classification, clustering or information retrieval?
- **Unsupervised metric learning**
 - What is a good metric for clustering: preliminary work on this question [Balcan et al., 2008b, Lajugie et al., 2014].
- **Adapting the metric to changing data**
 - Life Long learning (ERC grant C. Lampert).

A (first) quick advertisement...

Recent survey

There exist many other metric learning approaches. Most of them are discussed at more length in our recent survey:

- Bellet, A., Habrard, A., and Sebban, M. (2013). *A Survey on Metric Learning for Feature Vectors and Structured Data*. Technical report,

available at the following address: <http://arxiv.org/abs/1306.6709>

A (second) quick advertisement...



CAp'2014

**Conférence francophone sur
l'Apprentissage automatique**

Saint-Étienne, du 8 au 10 juillet 2014

<http://cap2014.sciencesconf.org>

Présidents du Comité Scientifique
 Marc SEBBAN - LaHC, Université de Saint-Étienne
 Ludovic DENOYER - LJP6, Université Paris 6

Président du Comité d'Organisation
 Amaury HABRARD - LaHC, Université de Saint-Étienne

HackDay, le 7 Juillet 2014
<http://hackday.lip6.fr>
 24h pour développer une application
 en apprentissage automatique

Conférenciers Invités
 Francis BACH - INRIA, ENS Paris, France
 Hendrik BLOCHEEL - KU Leuven, Belgique

Dates importantes

- ✓ Ouverture de la soumission des articles : 1^{er} mars 2014
- ✓ Clôture de la soumission des articles : 5 avril 2014
- ✓ Notification aux auteurs : 15 mai 2014
- ✓ Version finale : 1^{er} juin 2014