Supervised Metric Learning with Generalization Guarantees

Aurélien Bellet

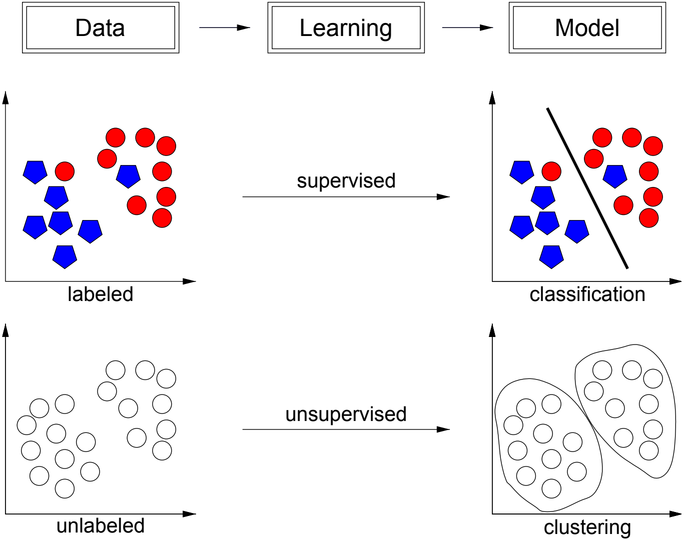
Laboratoire Hubert Curien, Université de Saint-Etienne, Université de Lyon

Reviewers: Pierre Dupont (UC Louvain) and Jose Oncina (U. Alicante)
Examiners: Rémi Gilleron (U. de Lille) and Liva Ralaivola (U. Aix-Marseille)
Supervisor: Marc Sebban (U. St-Etienne)
Co-supervisor: Amaury Habrard (U. St-Etienne)

PhD defense, December 11, 2012

Scientific context: machine learning

Learn to solve a task automatically



Data → Learning → Model

Numerical and structured data

Numerical data

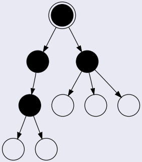
- Each data instance is a numerical feature vector.
- Example: the age, body mass index, blood pressure, ... of a patient.

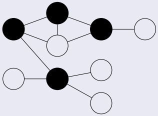
$$\mathbf{x} = \begin{pmatrix} 26 \\ 21.6 \\ 102 \\ \ldots \end{pmatrix}$$

Structured data

- Each instance is a structured object: a string, a tree or a graph.
- Examples: French words, DNA sequences, XML documents, molecules, social communities...

ACGGCTT





Metric learning

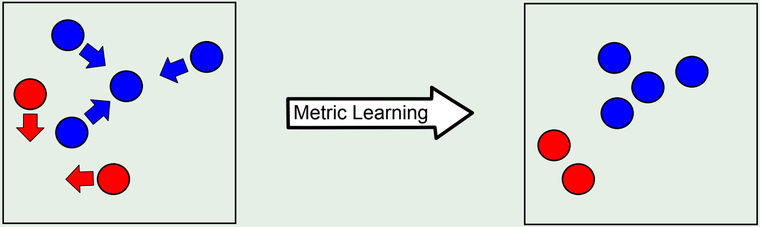
Adapt the metric to the problem of interest

A good metric as part of the solution

Many learning algorithms rely upon a notion of distance (or similarity) between pairs of examples.

Basic idea

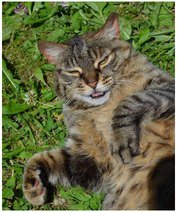
Learn a pairwise metric s.t. instances with same label are close and instances with different label are far away.



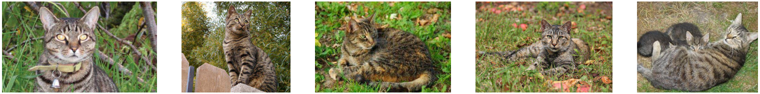
Metric learning

Application: image retrieval

Query image



Most similar images



Contributions of my thesis

1. A String Kernel Based on Learned Edit Similarities (PR '10)
2. Learning Good Similarities for Structured Data (ECML '11, MLJ '12)
3. Learning Good Similarities for Feature Vectors (ICML '12)
4. Robustness and Generalization for Metric Learning

Notations & Background

Supervised learning

Notations and basic notions

Input

A sample of N_T labeled examples $T = \{z_i = (x_i, y_i)\}_{i=1}^{N_T}$ independently and identically distributed (i.i.d.) according to an unknown distribution P over $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$. We focus on **binary classification**, where $\mathcal{Y} = \{-1, 1\}$.

Output

A hypothesis (model) h that is able to **accurately predict** the labels of (unseen) examples drawn from P.

Definition (True risk)

Given a loss function ℓ measuring the agreement between the prediction $h(x)$ and the true label y, we define the **true risk** by:

$$R^\ell(h) \quad = \quad \mathbb{E}_{z\sim P}\left[\ell(h, z)\right].$$

Supervised learning

Notations and basic notions

Definition (Empirical risk)

The **empirical risk** of an hypothesis h is the average loss suffered on the training sample T:

$$R_T^\ell(h) \quad = \quad \frac{1}{N_T}\sum_{i=1}^{N_T}\ell(h, z_i).$$

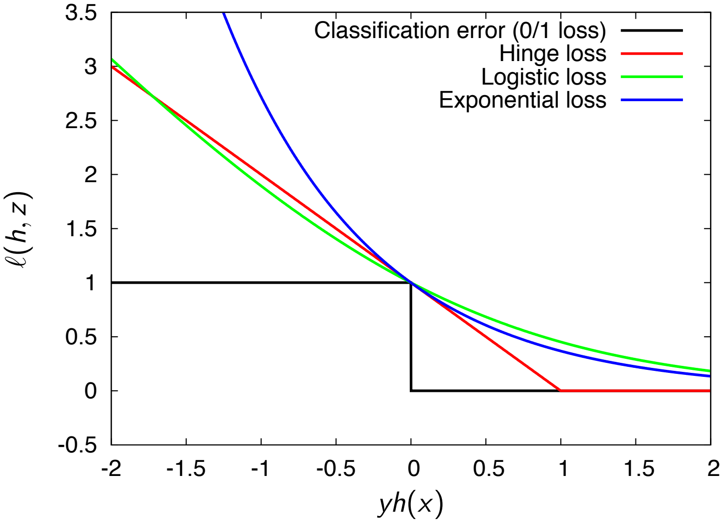
Generalization guarantees

Under some conditions, we may be able to **bound the deviation between the true risk and the empirical risk** of an hypothesis, i.e., how much we "trust" $R_T^\ell(h)$:

$$\Pr[|R^\ell(h) - R_T^\ell(h)| > \mu] \leq \delta. \qquad \text{(PAC bounds)}$$

Supervised learning

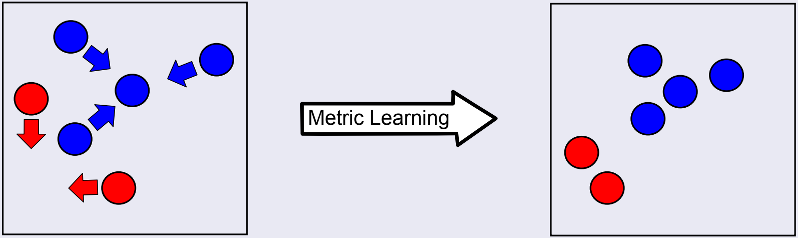
Loss functions



Metric learning

Basic setting

Finding a better representation space



Optimize over local constraints

Existing methods learn the parameters of some metric with respect to **local pair-based or triplet-based constraints** such that:

- x_i and x_j should be close to (or far away from) each other.
- x_i should be closer to x_k than to x_j.

Metric learning

Methods of the literature

Very popular approach

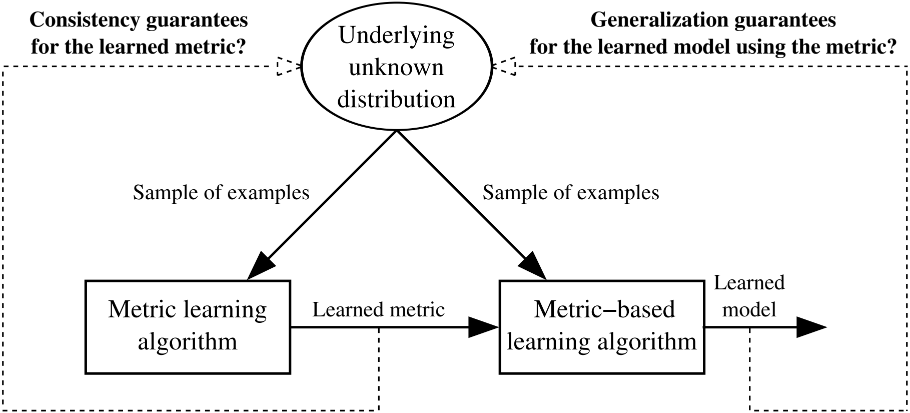
- Find the positive semi-definite (PSD) matrix $\mathbf{M} \in \mathbb{R}^{d \times d}$ parameterizing a (squared) **Mahalanobis distance** $d^2_{\mathbf{M}}(\mathbf{x}, \mathbf{x}') = (\mathbf{x} - \mathbf{x}')^T \mathbf{M} (\mathbf{x} - \mathbf{x}')$ such that $d^2_{\mathbf{M}}$ satisfies best the constraints.
- Different methods essentially differ by the choice of **constraints**, **loss function** and **regularizer** on $\mathbf{M}$.

Solving the problems

- For **feature vectors**: convex optimization.
- For **structured data**: inference in probabilistic models.

Metric learning

Generalization



“Plug and hope” strategy

For most approaches, no generalization guarantees are established.

Contributions

Contributions

Metric learning algorithms designed to improve **sparse linear classifiers**, with **generalization guarantees**.

1. We will first tackle the case of **structured data**,
2. and extend these ideas to **numerical data**.

An important building block

The work of [Balcan et al., 2008a] which

- establishes **a link between properties of a similarity function and generalization of a linear classifier**.
- but provides no algorithm to **learn** such a good similarity.

Learning with (ϵ, γ, τ)-Good Similarity Functions

Definition of goodness

Definition (Balcan et al., 2008)

A similarity function $K \in [-1, 1]$ is an (ϵ, γ, τ)-**good similarity function** if there exists an indicator function $R(x)$ defining a set of "reasonable points" such that the following conditions hold:

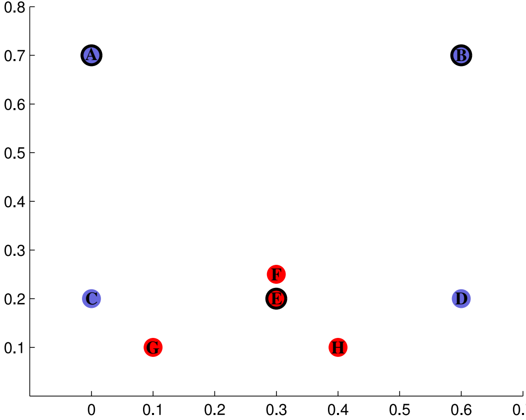
1. A $1 - \epsilon$ probability mass of examples (x, y) satisfy:

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

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

Intuition behind the definition

In this example, $K(\mathbf{x}, \mathbf{x}') = 1 - \|\mathbf{x} - \mathbf{x}'\|_2$ is $(0, 0.006, 3/8)$-good, $(2/8, 0.01, 3/8)$-good...

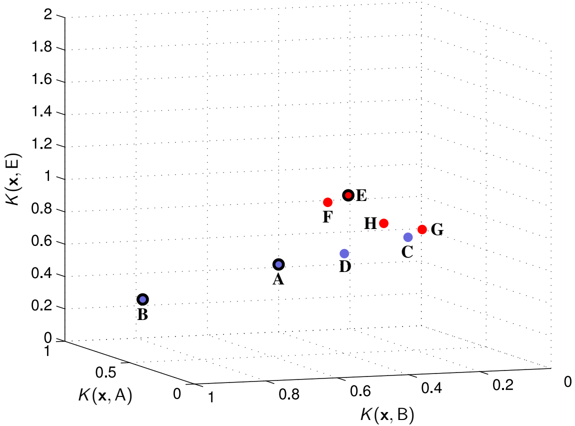


	A	B	C	D	E	F	G	H
A	1	0.40	0.50	0.22	0.42	0.46	0.39	0.28
B	0.40	1	0.22	0.50	0.42	0.46	0.22	0.37
E	0.42	0.42	0.70	0.70	1	0.95	0.78	0.86
Margin γ	0.3277	0.3277	0.0063	0.0063	0.0554	0.0106	0.0552	0.0707

Simple case: R is known

Strategy

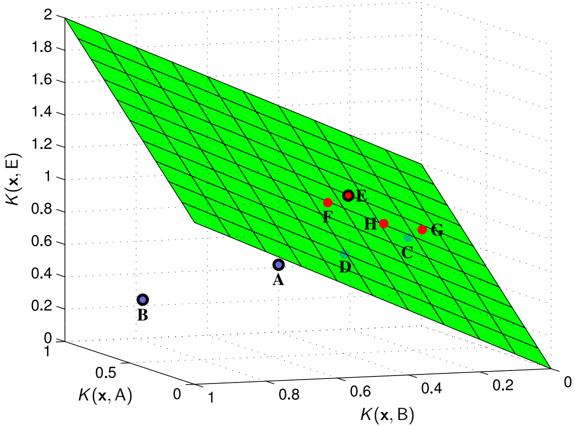
Use K to map the examples to the space ϕ of "the similarity scores with the reasonable points" (**similarity map**).



Simple case: R is known

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.



What if R is unknown?

Theorem (Balcan et al., 2008)

Given K is (ϵ, γ, τ)-good and enough points to create a similarity map, there exists a linear separator $\boldsymbol{\alpha}$ that has true risk close to ϵ at margin $\gamma/2$.

Question

Can we find this linear classifier in an efficient way?

Answer

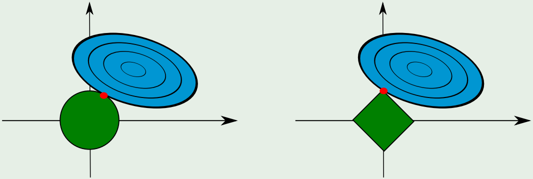
Basically, yes (only need to slightly reformulate the definition).

Learning rule

Learning the separator α with a **linear program**

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

L_1 norm induces sparsity



L_2 constraint L_1 constraint

Summary

A nice theory

The true risk of the sparse linear classifier depends on how well the similarity function satisfies the definition (basically, $R^\ell(h) \leq \epsilon$).

Limitation

For real-world problems, standard similarity functions may poorly satisfy the definition (large ϵ gives worthless guarantees).

Our idea

- Given a training set T, a set of reasonable points and a margin γ, try to **optimize the empirical goodness** over T.
- Generalization guarantees for the metric itself **implies** guarantees for the classifier!

Learning Good Similarities for Structured Data

Relevant publications

- Bellet, A., Habrard, A., and Sebban, M. (2011). Learning Good Edit Similarities with Generalization Guarantees. In *ECML/PKDD*, pages 188–203.
- Bellet, A., Habrard, A., and Sebban, M. (2012a). Good edit similarity learning by loss minimization. *Machine Learning*, 89(1):5–35.

Why structured data?

Motivation 1: structured metrics are convenient

Metrics for structured data (strings, trees, graphs) act as **proxies** to manipulate complex objects: can use any metric-based algorithm!

Motivation 2: drawbacks of the state-of-the-art

- Little work on metric learning from structured data.
- Most of it has focused on **edit distance learning**, through **likelihood maximization in probabilistic models** (costly and not flexible).

Motivation 3: avoid the PSD constraint

- Edit distance-based metrics are not PSD and difficult/costly to transform into kernels (cf Chapter 4).
- On the other hand, (ϵ, γ, τ)-goodness is well-suited to edit similarities (cf preliminary study of Chapter 5).

The edit distance

The edit distance is the cost of the cheapest sequence of operations (*script*) turning a string into another. Allowable operations are *insertion*, *deletion* and *substitution* of symbols. Costs are gathered in a matrix **C**.

Example 1: Standard (Levenshtein) distance

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

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

Example 2: Specific Cost Matrix

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

$\Longrightarrow$ edit distance between $\mathtt{abb}$ and $\mathtt{aa}$ is 10 ($\mathtt{a} \rightarrow \$, \mathtt{b} \rightarrow \mathtt{a}, \mathtt{b} \rightarrow \mathtt{a}$)

\$: empty symbol, Σ: alphabet, **C**: $(|\Sigma|+1) \times (|\Sigma|+1)$ matrix with positive values.

Key simplification #1

Learning the costs of the Levenshtein script

- **No closed-form expression** for the edit distance → methods of the literature use iterative procedures.
- We make the following key simplification: **fix the edit script**.

Definition of $e_\mathbf{C}$

$$e_\mathbf{C}(\mathrm{x},\mathrm{x}') = \sum_{0\leq i,j\leq|\Sigma|} C_{i,j} \times \#_{i,j}(\mathrm{x},\mathrm{x}'),$$

where $\#_{i,j}(\mathrm{x},\mathrm{x}')$ is the number of times the operation $i \to j$ appears in the Levenshtein script.

We will in fact optimize:

Definition of $K_\mathbf{C}$

$$K_\mathbf{C}(\mathrm{x},\mathrm{x}') = 2\exp(-e_\mathbf{C}(\mathrm{x},\mathrm{x}')) - 1 \in [-1,1]$$

Key simplification #2

Optimize a relaxed version of the goodness

Avoid a nonconvex formulation

Optimizing the (ϵ, γ, τ)-goodness of $K_{\mathbf{C}}$ would result in a **nonconvex** formulation (summing/subtracting up exponential terms).

A new criterion

$$\mathbb{E}_{(x,y)}\left[\mathbb{E}_{(x',y')}\left[\left[1 - yy'K_{\mathbf{C}}(x,x')/\gamma\right]_+ |R(x')\right]\right] \leq \epsilon' \qquad (1)$$

Interpretation

- Eq. (1) bounds the criterion of (ϵ, γ, τ)-goodness: **"goodness" is required with respect to each reasonable point** (instead of considering the average similarity to these points).
- Consequently, optimizing (1) implies the use of **pair-based constraints**.

Problem formulation

Recall the underlying idea

Moving closer pairs of same class and further away those of opposite class.

GESL: a convex QP formulation

$$\min_{\mathbf{C},B_1,B_2} \quad \frac{1}{N_T N_L} \sum_{\substack{1\leq i\leq N_T,\\ j:f_{land}(z_i,z_j)=1}} \ell(\mathbf{C}, z_i, z_j) + \beta\|\mathbf{C}\|^2_{\mathcal{F}}$$

$$\text{s.t.} \quad B_1 \geq -\log(\tfrac{1}{2}), \quad 0 \leq B_2 \leq -\log(\tfrac{1}{2}), \quad B_1 - B_2 = \eta_\gamma$$

$$C_{i,j} \geq 0, \quad 0 \leq i, j \leq |\Sigma|,$$

where $\ell(\mathbf{C}, z_i, z_j) = \begin{cases} [B1 - e_{\mathbf{C}}(\mathrm{x_i}, \mathrm{x_j})]_+ \text{ if } y_i \neq y_j \\ [e_{\mathbf{C}}(\mathrm{x_i}, \mathrm{x_j}) - B2]_+ \text{ if } y_i = y_j \end{cases}.$

Two parameters: β (regularization parameter on the edit costs) and η_γ (the “desired margin”).

Generalization guarantees

Theorem: GESL has a uniform stability in κ/N_T

$$\kappa = \frac{2(2+\alpha)W^2}{\beta\alpha}$$

W is a bound on the string sizes; $0 \leq \alpha \leq 1$ such that $N_L = \alpha N_T$.

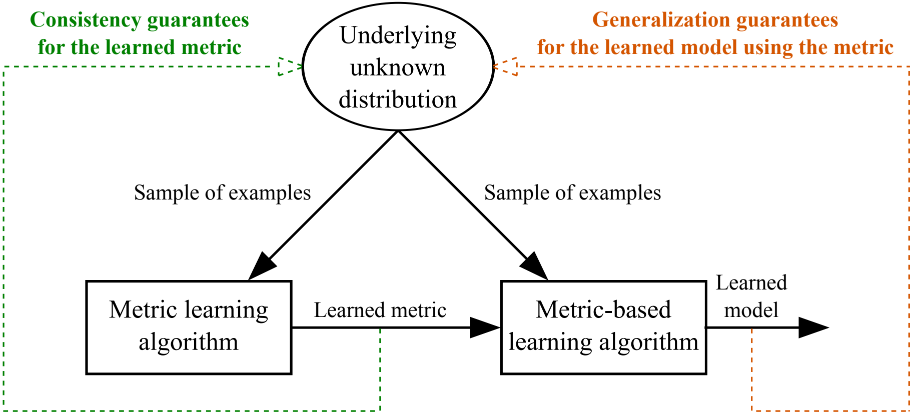
Theorem: generalization bound - convergence in $O(\sqrt{1/N_T})$

With probability at least $1-\delta$:

$$R^\ell(\mathbf{C}) < R^\ell_T(\mathbf{C}) + 2\frac{\kappa}{N_T} + (2\kappa + B)\sqrt{\frac{\ln(2/\delta)}{2N_T}}.$$

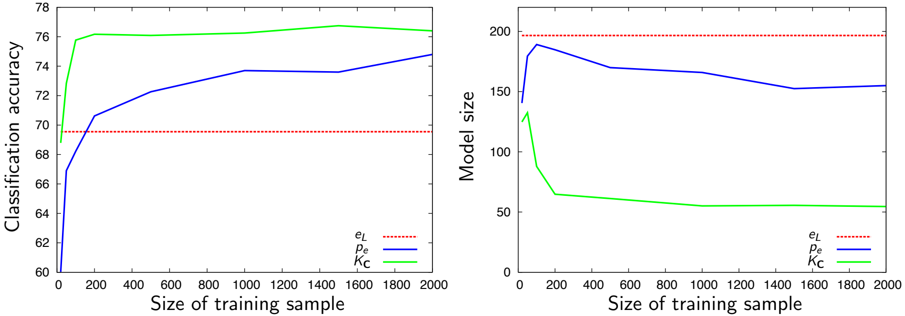
- Gives a (loose) bound on the **true** goodness.
- “Independence” from the **size of the alphabet**.

Generalization guarantees



Experimental results

Task: classify words as either French or English



(more experiments on handwritten digit recognition)

Summary & perspectives

Summary

- An edit similarity learning method which addresses the classic drawbacks of state-of-the-art methods.
- The learned similarity is used to build a sparse linear classifier.
- Generalization guarantees in terms of (i) the learned similarity and (ii) the true risk of the classifier.

Perspectives: extension to trees

- Straightforward extension to **tree-structured data** (use a Levenshtein tree edit script).
- Ongoing experiments in melody recognition (with J.F. Bernabeu, Universidad de Alicante).

Learning Good Similarities for Feature Vectors

Relevant publication

- Bellet, A., Habrard, A., and Sebban, M. (2012b). Similarity Learning for Provably Accurate Sparse Linear Classification. In *ICML*.

Form of similarity function

Bilinear Similarity

Optimize the **bilinear similarity** $K_{\mathbf{A}}$:

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

parameterized by the matrix $\mathbf{A} \in \mathbb{R}^{d \times d}$.

Advantages

- Unlike Mahalanobis, **A** is not constrained to be PSD nor symmetric (easier to optimize).
- $K_{\mathbf{A}}$ is efficiently computable for sparse inputs.
- Can define a similarity between objects of different dimension by taking **A** nonsquare.

Empirical goodness

Goal

Optimize the (ϵ, γ, τ)-goodness of $K_{\mathbf{A}}$ on a finite-size sample.

Notations

Given a training sample $T = \{\mathbf{z_i} = (\mathbf{x_i}, y_i)\}_{i=1}^{N_T}$, a subsample $R \subseteq T$ of N_R reasonable points and a margin γ,

$$\ell(\mathbf{A}, \mathbf{z_i}, R) = [1 - y_i \frac{1}{\gamma N_R} \sum_{k=1}^{N_R} y_k K_{\mathbf{A}}(\mathbf{x_i}, \mathbf{x_k})]_+$$

is the empirical goodness of $K_{\mathbf{A}}$ w.r.t. a single training point $\mathbf{z_i} \in T$, and

$$\epsilon_T = \frac{1}{N_T} \sum_{i=1}^{N_T} \ell(\mathbf{A}, \mathbf{z_i}, R)$$

is the empirical goodness over T.

Formulation

SLLC (Similarity Learning for Linear Classification)

$$\min_{\mathbf{A}\in\mathbb{R}^{d\times d}} \quad \epsilon_T \quad + \quad \beta\|\mathbf{A}\|_{\mathcal{F}}^2$$

where β is a regularization parameter.

Desirable properties

- SLLC can be efficiently solved in a batch or online way via unconstrained or constrained optimization.
- Different from classic metric learning approaches (including GESL): similarity constraints must be satisfied only **on average**, learn **global** similarity (R is common to all training examples).

Generalization guarantees

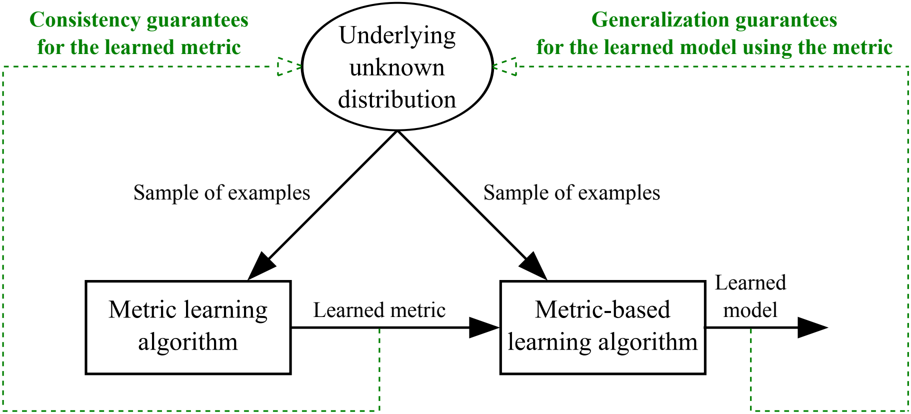
Theorem: generalization bound - convergence in $O(\sqrt{1/N_T})$

With probability $1-\delta$, we have:

$$\epsilon \leq \epsilon_T + \frac{\kappa}{N_T} + (2\kappa+1)\sqrt{\frac{\ln 1/\delta}{2N_T}}.$$

- **Tighter bound on the true goodness** (and thus on the true risk of the classifier).
- "Independence" from the **dimensionality of the problem**.

Generalization guarantees



Experimental set-up

- 7 UCI datasets of varying size and complexity:

	Breast	Iono.	Rings	Pima	Splice	Svmguide1	Cod-RNA
train size	488	245	700	537	1,000	3,089	59,535
test size	211	106	300	231	2,175	4,000	271,617
# dimensions	9	34	2	8	60	4	8
# runs	100	100	100	100	1	1	1

- We compare SLLC to K_I (cosine baseline) and two widely-used Mahalanobis distance learning methods: LMNN and ITML.

Experiments

Overall Results

	Breast	Iono.	Rings	Pima	Splice	Svmguide1	Cod-RNA
K_I	96.57 (20.39)	89.81 (52.93)	100.00 (18.20)	75.62 (25.93)	83.86 (362)	96.95 (64)	95.91 (557)
SLLC	96.90 (1.00)	93.25 (1.00)	100.00 (1.00)	75.94 (1.00)	87.36 (1)	96.55 (8)	94.08 (1)
LMNN	96.46 (488)	88.68 (245)	100.00 (700)	73.50 (537)	87.59 (1,000)	96.23 (3,089)	94.98 (59,535)
ITML	96.38 (488)	88.29 (245)	100.00 (700)	72.80 (537)	84.41 (1,000)	96.80 (3,089)	95.42 (59,535)

- SLLC outperforms K_I, LMNN and ITML on 4 out of 7 datasets.
- Always leads to **extremely sparse models**.

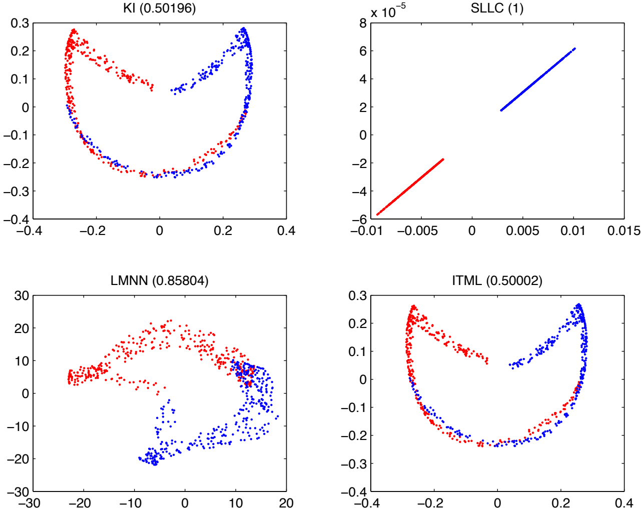
Experiments

Linear classification

	Breast	Iono.	Rings	Pima	Splice	Svmguide1	Cod-RNA
K_I	96.57 (20.39)	89.81 (52.93)	100.00 (18.20)	75.62 (25.93)	83.86 (362)	96.95 (64)	95.91 (557)
SLLC	96.90 (1.00)	93.25 (1.00)	100.00 (1.00)	75.94 (1.00)	87.36 (1)	96.55 (8)	94.08 (1)
LMNN	96.81 (9.98)	90.21 (13.30)	100.00 (8.73)	75.15 (69.71)	86.85 (156)	96.53 (82)	95.15 (591)
ITML	96.80 (9.79)	93.05 (18.01)	100.00 (15.21)	75.25 (16.40)	85.29 (287)	96.70 (49)	95.14 (206)

Experiments

PCA projection of the "similarity map" space (Rings dataset)



Experiments

k-Nearest Neighbors

	Breast	Iono.	Pima	Splice	Svmguide1	Cod-RNA
K_I	96.71	83.57	72.78	77.52	93.93	90.07
SLLC	96.90	93.25	75.94	87.36	93.82	94.08
LMNN	96.46	88.68	73.50	87.59	96.23	94.98
ITML	96.38	88.29	72.80	84.41	96.80	95.42

Surprisingly, SLLC also outperforms LMNN and ITML on the small datasets.

Summary & perspectives

Summary

- A global similarity learning method with an efficient formulation.
- Leads to extremely sparse linear classifiers.
- Tighter generalization bounds than with GESL.

Perspectives

- Experiment with online optimization algorithms to make the approach scalable to very large datasets.
- Study the influence of other regularizers (e.g., $L_{2,1}$ norm). Thanks to an adaptation of algorithmic robustness (cf Chapter 7 of the thesis) to metric learning, we do not have to give up generalization guarantees!

General Perspectives

General perspectives

On the practical side

- Like GESL, make implementation of SLLC publicly available.
- Online learning of (ϵ, γ, τ)-good similarities.
- Play with sparsity-inducing regularizers.
- . . .

On the theoretical side

- What about guarantees for Nearest Neighbors?
- What about the regression setting? Could use the recent adaptation of goodness to this setting [Kar and Jain, 2012].
- What properties of a metric make quality clusterings? First attempt: [Balcan et al., 2008b].
- . . .

Thank you for your attention

References I

[Balcan et al., 2008a] Balcan, M.-F., Blum, A., and Srebro, N. (2008a).
Improved Guarantees for Learning via Similarity Functions.
In *COLT*, pages 287–298.

[Balcan et al., 2008b] Balcan, M.-F., Blum, A., and Vempala, S. (2008b).
A Discriminative Framework for Clustering via Similarity Functions.
In *STOC*, pages 671–680.

[Bellet et al., 2011] Bellet, A., Habrard, A., and Sebban, M. (2011).
Learning Good Edit Similarities with Generalization Guarantees.
In *ECML/PKDD*, pages 188–203.

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

References II

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

[Kar and Jain, 2012] Kar, P. and Jain, P. (2012).
Supervised Learning with Similarity Functions.
In *NIPS*.