Similarity Learning for Provably Accurate Sparse Linear Classification

Aurélien Bellet Amaury Habrard Marc Sebban

Laboratoire Hubert Curien, UMR CNRS 5516, Université de Saint-Etienne, France

ICML 2012

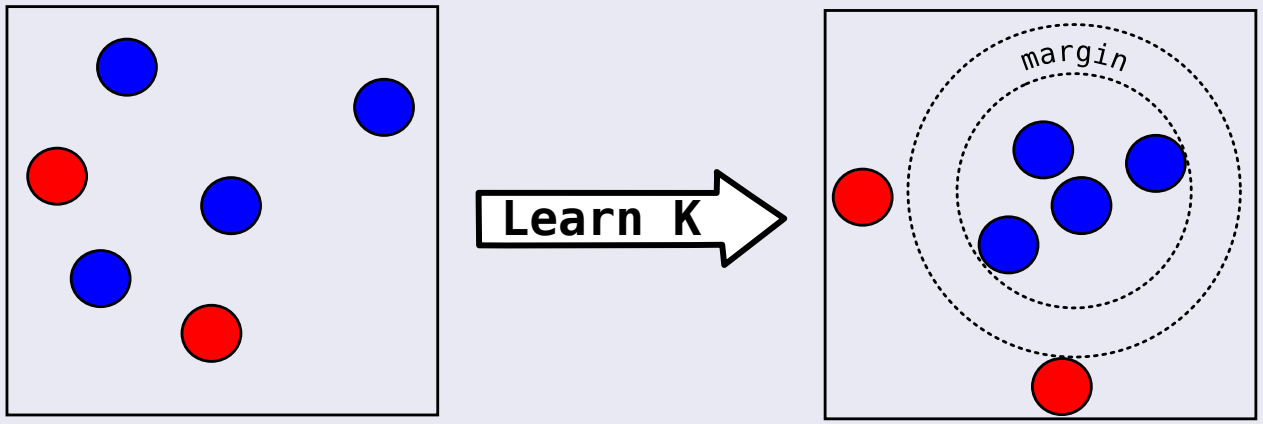
Introduction

Similarity/Distance Learning

Similarity learning

Similarity learning overview

Learning a similarity function $K(x, x')$ implying a new instance space where the performance of a given algorithm is improved.



Very popular approach

Find the positive semi-definite (PSD) matrix $\mathbf{M} \in \mathbb{R}^{d \times d}$ parameterizing a (squared) **Mahalanobis distance** $d^2_{\mathbf{M}}(x, y) = (\mathbf{x} - \mathbf{x}')^T \mathbf{M}(\mathbf{x} - \mathbf{x}')$ such that $d^2_{\mathbf{M}}$ satisfies best **local constraints**.

Motivation of our work

Limitations of Mahalanobis distance learning

- Must enforce $\mathbf{M} \succeq 0$ (costly).
- Works well in practice in k-NN (based on **local** neighborhoods), but not really appropriate for **global** classifiers?
- **No theoretical link** between the learned metric and the error of the classifier.

Goal of our work

- Learn a **non PSD** similarity function,
- designed to improve **global linear classifiers**,
- with **theoretical guarantees** on the classifier error.

(ϵ, γ, τ)-Good Similarity Functions

Definition

The theory of Balcan et al. (2006, 2008) makes the link between the **properties an arbitrary similarity function** and **its performance in binary linear classification**.

Definition (Balcan et al., 2008)

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

1. A $1 - \epsilon$ probability mass of examples $(\mathbf{x}, \ell)$ satisfy:

$$\mathbf{E}_{(\mathbf{x}', \ell') \sim P} \left[\ell \ell' K(\mathbf{x}, \mathbf{x}') | R(\mathbf{x}') \right] \geq \gamma$$

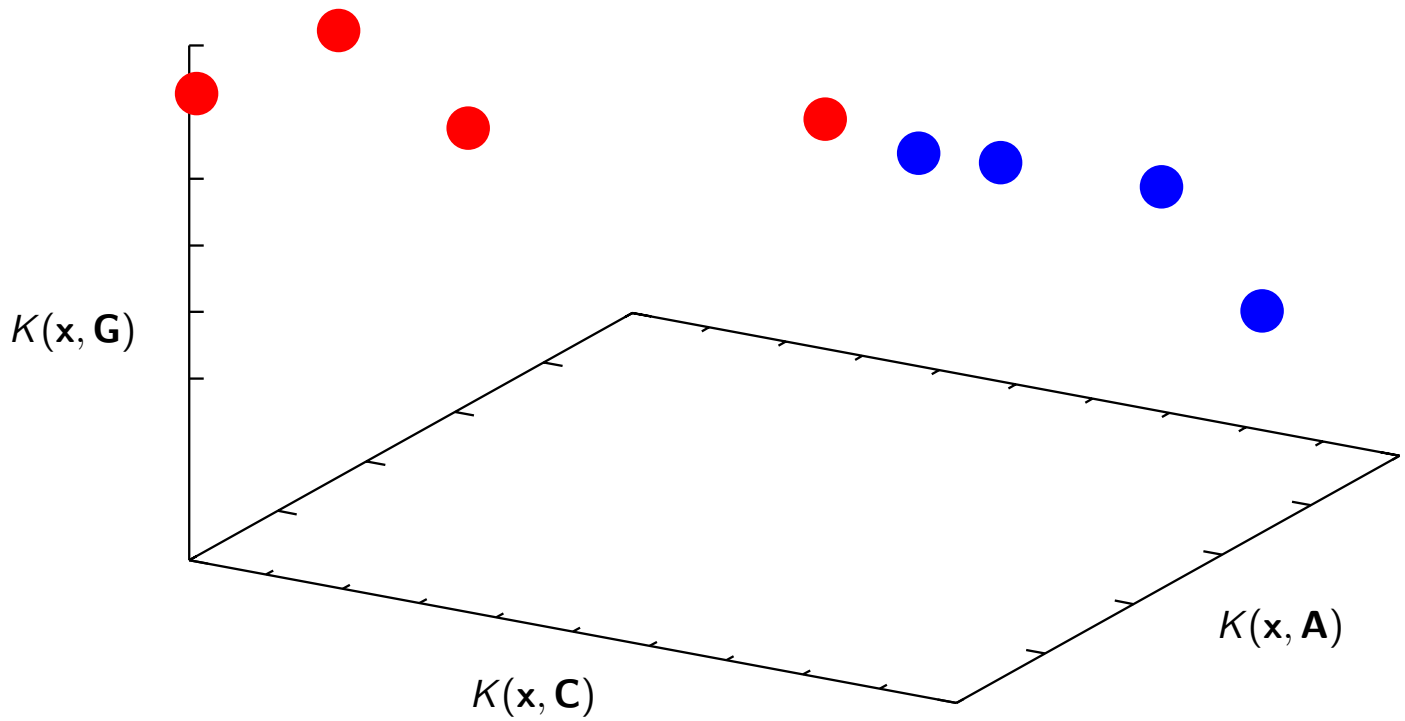
2. $\Pr_{\mathbf{x}'}[R(\mathbf{x}')] \geq \tau$.

$\epsilon, \gamma, \tau \in [0, 1]$

Implications for learning

Strategy

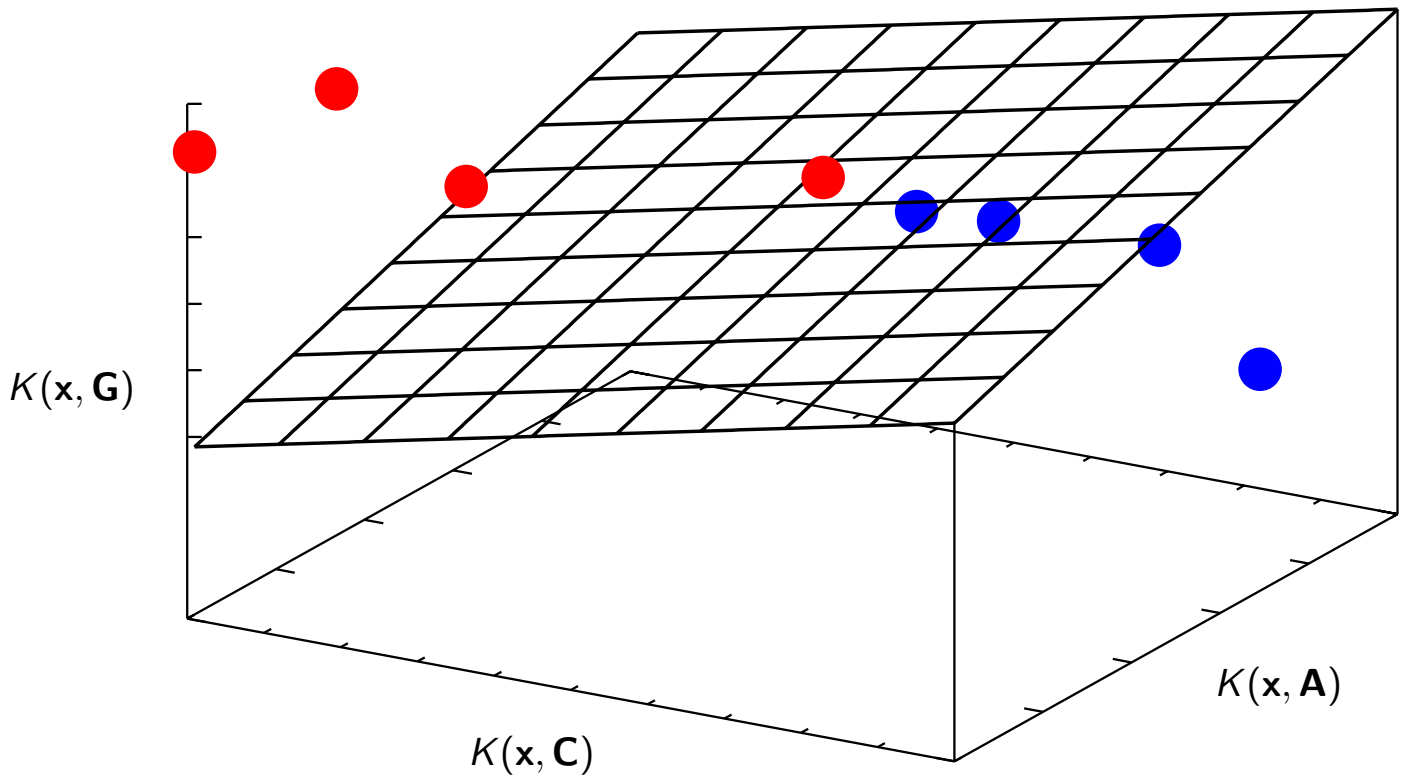
Each example is mapped to the space of "the similarity scores with the reasonable points" (**similarity map**).



Implications for learning

Theorem (Balcan et al., 2008)

Given K is (ϵ, γ, τ)-good, there exists a linear separator α in the above-defined projection space that has error close to ϵ at margin γ.



Hinge loss definition

Hinge loss version of the definition.

Definition (Balcan et al., 2008)

A similarity function K is an (ϵ, γ, τ)-*good similarity function in hinge loss* for a learning problem P if there exists a (random) indicator function $R(\mathbf{x})$ defining a (probabilistic) set of "reasonable points" such that the following conditions hold:

1. $\mathbb{E}_{(\mathbf{x},\ell)\sim P}\left[[1 - \ell g(\mathbf{x})/\gamma]_+\right] \leq \epsilon$,
   where $g(\mathbf{x}) = \mathbb{E}_{(\mathbf{x}',\ell')\sim P}[\ell' K(\mathbf{x}, \mathbf{x}')|R(\mathbf{x}')]$ and
   $[1 - c]_+ = \max(1 - c, 0)$ is the hinge loss,
2. $\Pr_{\mathbf{x}'}[R(\mathbf{x}')] \geq \tau$.

Learning rule

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

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

Advantage: sparsity

Thanks to **L_1-regularization**, α will have some zero-coordinates (depending on λ). Makes prediction much faster than k-NN.

Learning Good Similarity Functions for Linear Classification

Form of similarity function

- We propose to optimize a **bilinear similarity** $K_{\mathbf{A}}$:

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

  parameterized by the matrix $\mathbf{A} \in \mathbb{R}^{d \times d}$ (not constrained to be PSD nor symmetric).
- $K_{\mathbf{A}}$ is efficiently computable for sparse inputs.

Empirical goodness

Goal

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

Notations

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

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

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

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

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.

- SLLC can be cast as a **convex QP** and efficiently solved.
- Only **one constraint per training example**.
- Different from classic metric learning approaches: similarity constraints must be satisfied only **on average**, learn **global** similarity (same R for all training examples).

Theoretical analysis

We want to bound the **goodness in generalization** ϵ of our learned similarity:

$$\epsilon = \mathbb{E}_{\mathbf{z}=(\mathbf{x},l)\sim P} V(\mathbf{A}, \mathbf{z}, R)$$

by its **empirical goodness** ϵ_T:

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

Theoretical analysis ctd

Theorem: SLLC has a uniform stability in κ/N_T

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

where β is the regularization parameter, γ the margin and $\hat{\tau}$ the proportion of reasonable points in the training sample.

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

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

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

Guarantee on the error of the classifier and **convergence rate independent from dimensionality**.

Experimental set-up

- 7 datasets

	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

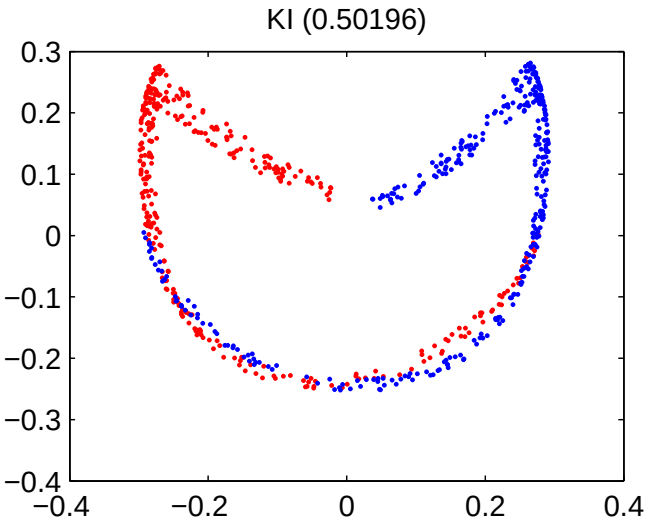
	Breast	Iono.	Rings	Pima	Splice	Svmguide1	Cod-RNA
K_I	96.57 (20.39)	89.81 (52.93)	100.00 (18.20)	75.62 (25.93)	83.86 (362)	**96.95** **(64)**	**95.91** **(557)**
SLLC	**96.90** **(1.00)**	**93.25** **(1.00)**	**100.00** **(1.00)**	**75.94** **(1.00)**	87.36 **(1)**	96.55 (8)	94.08 (1)
LMNN	96.46 (488)	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)

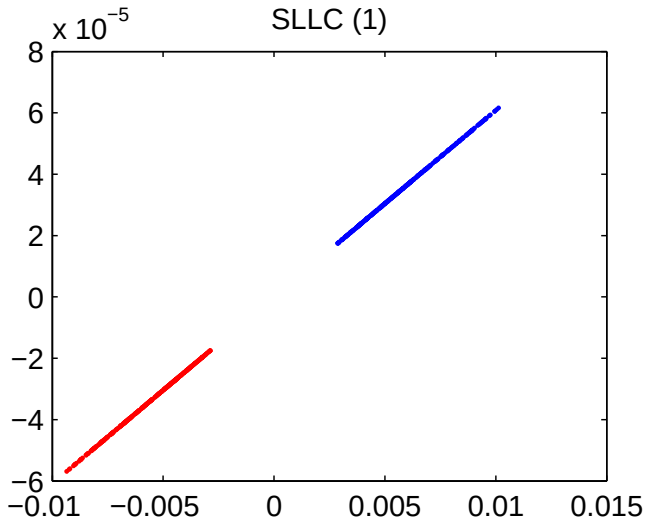
- SLLC outperforms K_I, LMNN and ITML on 4 out of 7 datasets.
- Always leads to **extremely sparse models**.

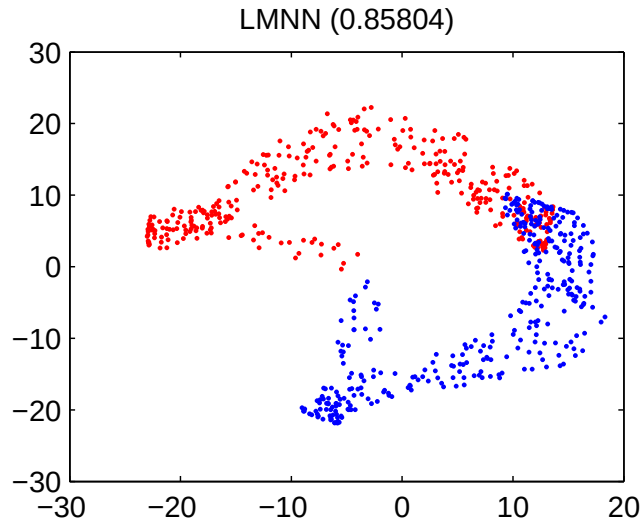
Experiments: linear classification

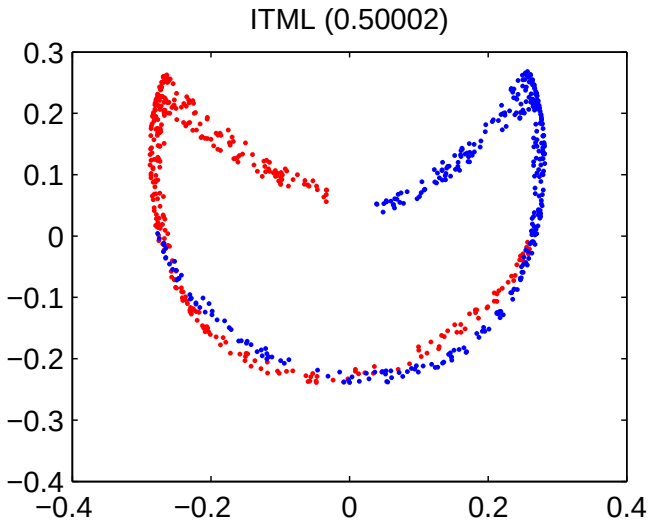
	Breast	Iono.	Rings	Pima	Splice	Svmguide1	Cod-RNA
K_I	96.57 (20.39)	89.81 (52.93)	100.00 (18.20)	75.62 (25.93)	83.86 (362)	**96.95** **(64)**	**95.91** **(557)**
SLLC	**96.90** **(1.00)**	**93.25** **(1.00)**	**100.00** **(1.00)**	**75.94** **(1.00)**	**87.36** **(1)**	96.55 (8)	94.08 (1)
LMNN	96.81 (9.98)	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: projection space









Experiments: k-NN

	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.

Conclusion

Conclusion

Making use of Balcan et al.'s theory, we propose a novel similarity learning method that:

- has guarantees in terms of the error of a linear classifier,
- is effective in practice as compared to the state-of-the-art,
- produces extremely sparse models.

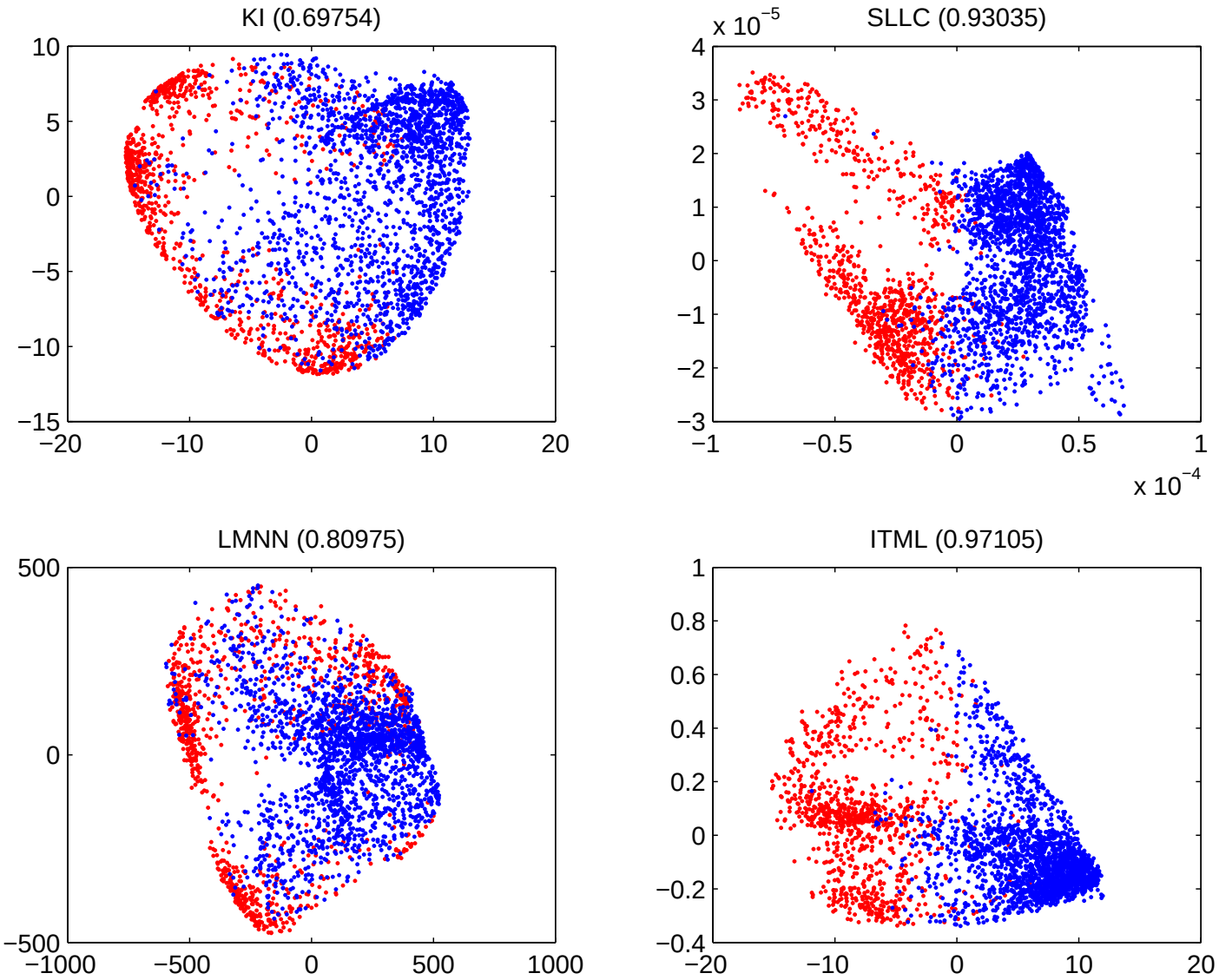
Future work could include:

- playing with other regularizers ($L_{1,2}$-norm?),
- deriving an online algorithm.

Thank you!

Come to the poster for more details :-)

Backup slide 1: another projection space example



Backup slide 2: time complexity

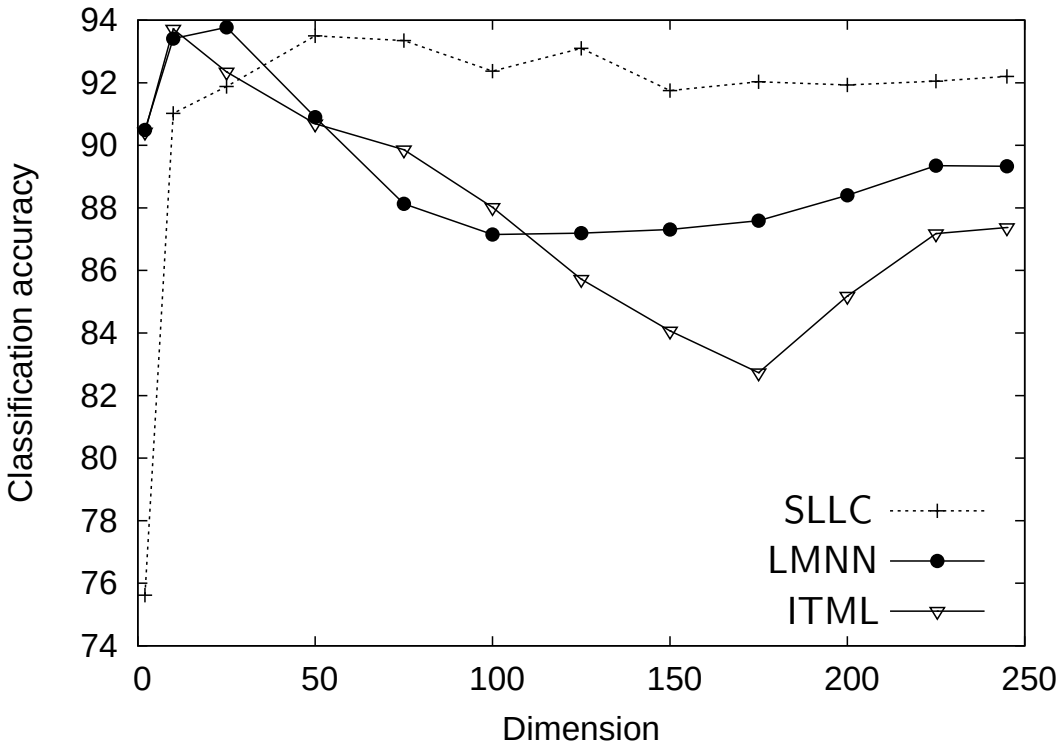
- LMNN and ITML have their own sophisticated solver.
- For SLLC we just use a standard convex programming package.
- **SLLC is much faster than LMNN** but **remains slower than ITML**.

	Breast	Iono.	Rings	Pima	Splice	Svmguide1	Cod-RNA
SLLC	4.76	5.36	0.05	4.01	158.38	185.53	2471.25
LMNN	25.99	16.27	37.95	32.14	309.36	331.28	10418.73
ITML	1.68	3.09	0.19	2.74	3.41	0.83	5.98

Backup slide 3: kernelization

- Our approach is very simple: learn a global linear similarity, use it to learn a global linear classifier.
- Would be interesting to be able to learn more powerful similarities and classifiers.
- We **kernelize** SLLC to be able to learn in a **nonlinear** feature space induced by a kernel.
- This is done with the **KPCA trick** (Chatpatanasiri et al., 2010): projection of data in kernel space + dimensionality reduction.
- Then we apply SLLC in this new feature space.

Backup slide 4: overfitting



LMNN and ITML overfit the data as dimensionality grows.