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
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 $M \in \mathbb{R}^{d \times d}$ parameterizing a (squared) Mahalanobis distance $d_M^2(x, y) = (x - x')^T M(x - x')$ such that $d_M^2$ satisfies best local constraints.
Motivation of our work

Limitations of Mahalanobis distance learning

- Must enforce $M \succeq 0$ (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.
$(\epsilon, \gamma, \tau)$-Good Similarity Functions
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 $(\epsilon, \gamma, \tau)$-good similarity function for a learning problem $P$ 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 $(x, \ell)$ satisfy:

   $$E_{(x', \ell') \sim P} [\ell \ell' K(x, x') | R(x')] \geq \gamma$$

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

$\epsilon, \gamma, \tau \in [0, 1]$
Implications for learning

Strategy

Each example is mapped to the space of “the similarity scores with the reasonable points” (similarity map).
Theorem (Balcan et al., 2008)

Given $K$ is $(\epsilon, \gamma, \tau)$-good, there exists a linear separator $\alpha$ in the above-defined projection space that has error close to $\epsilon$ at margin $\gamma$. 

$$K(x, G)$$

$$K(x, C)$$

$$K(x, A)$$
Hinge loss version of the definition.

**Definition (Balcan et al., 2008)**

A similarity function $K$ is an $(\epsilon, \gamma, \tau)$-good similarity function in hinge loss for a learning problem $P$ if there exists a (random) indicator function $R(x)$ defining a (probabilistic) set of “reasonable points” such that the following conditions hold:

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

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

Learning the separator $\alpha$ with a **linear program**

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

**Advantage: sparsity**

Thanks to **L$_1$-regularization**, $\alpha$ will have some zero-coordinates (depending on $\lambda$). Makes prediction much faster than $k$-NN.
Learning Good Similarity Functions for Linear Classification
We propose to optimize a **bilinear similarity** $K_A$: 

$$K_A(x, x') = x^T A x'$$

parameterized by the matrix $A \in \mathbb{R}^{d \times d}$ (not constrained to be PSD nor symmetric).

$K_A$ is efficiently computable for sparse inputs.
Empirical goodness

Goal

Optimize the \((\epsilon, \gamma, \tau)\)-goodness of \(K_A\) on a finite-size sample.

Notations

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

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

is the empirical goodness of \(K_A\) w.r.t. a single training point \(z_i \in T\), and

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

is the empirical goodness over \(T\).
SLLC (Similarity Learning for Linear Classification)

\[
\min_{A \in \mathbb{R}^{d \times d}} \epsilon_T + \beta \|A\|_F^2
\]

where \(\beta\) 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** $\epsilon$ of our learned similarity:

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

by its **empirical goodness** $\epsilon_T$:

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

**Theorem:** SLLC has a uniform stability in $\kappa / N_T$

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

where $\beta$ is the regularization parameter, $\gamma$ 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

<table>
<thead>
<tr>
<th></th>
<th>Breast</th>
<th>Iono.</th>
<th>Rings</th>
<th>Pima</th>
<th>Splice</th>
<th>Svmguide1</th>
<th>Cod-RNA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train size</td>
<td>488</td>
<td>245</td>
<td>700</td>
<td>537</td>
<td>1,000</td>
<td>3,089</td>
<td>59,535</td>
</tr>
<tr>
<td>Test size</td>
<td>211</td>
<td>106</td>
<td>300</td>
<td>231</td>
<td>2,175</td>
<td>4,000</td>
<td>271,617</td>
</tr>
<tr>
<td># dimensions</td>
<td>9</td>
<td>34</td>
<td>2</td>
<td>8</td>
<td>60</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td># runs</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th></th>
<th>Breast</th>
<th>Ion.</th>
<th>Rings</th>
<th>Pima</th>
<th>Splice</th>
<th>SVMguide1</th>
<th>Cod-RNA</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K_I$</td>
<td>96.57</td>
<td>89.81</td>
<td>100.00</td>
<td>75.62</td>
<td>83.86</td>
<td>96.95</td>
<td>95.91</td>
</tr>
<tr>
<td></td>
<td>(20.39)</td>
<td>(52.93)</td>
<td>(18.20)</td>
<td>(25.93)</td>
<td>(362)</td>
<td>(64)</td>
<td>(557)</td>
</tr>
<tr>
<td>SLLC</td>
<td>96.90</td>
<td>93.25</td>
<td>100.00</td>
<td>75.94</td>
<td>87.36</td>
<td>96.55</td>
<td>94.08</td>
</tr>
<tr>
<td></td>
<td>(1.00)</td>
<td>(1.00)</td>
<td>(1.00)</td>
<td>(1.00)</td>
<td>(1)</td>
<td>(8)</td>
<td>(1)</td>
</tr>
<tr>
<td>LMNN</td>
<td>96.46</td>
<td>88.68</td>
<td>100.00</td>
<td>73.50</td>
<td>87.59</td>
<td>96.23</td>
<td>94.98</td>
</tr>
<tr>
<td></td>
<td>(488)</td>
<td>(245)</td>
<td>(700)</td>
<td>(537)</td>
<td>(1,000)</td>
<td>(3,089)</td>
<td>(59,535)</td>
</tr>
<tr>
<td>ITML</td>
<td>96.38</td>
<td>88.29</td>
<td>100.00</td>
<td>72.80</td>
<td>84.41</td>
<td>96.80</td>
<td>95.42</td>
</tr>
<tr>
<td></td>
<td>(488)</td>
<td>(245)</td>
<td>(700)</td>
<td>(537)</td>
<td>(1,000)</td>
<td>(3,089)</td>
<td>(59,535)</td>
</tr>
</tbody>
</table>

- SLLC outperforms $K_I$, LMNN and ITML on 4 out of 7 datasets.
- Always leads to **extremely sparse models**.
## Experiments: linear classification

<table>
<thead>
<tr>
<th></th>
<th>Breast</th>
<th>Iono.</th>
<th>Rings</th>
<th>Pima</th>
<th>Splice</th>
<th>SVMguide1</th>
<th>Cod-RNA</th>
</tr>
</thead>
<tbody>
<tr>
<td>K₁</td>
<td>96.57</td>
<td>89.81</td>
<td>100.00</td>
<td>75.62</td>
<td>83.86</td>
<td>96.95</td>
<td>95.91</td>
</tr>
<tr>
<td></td>
<td>(20.39)</td>
<td>(52.93)</td>
<td>(18.20)</td>
<td>(25.93)</td>
<td>(362)</td>
<td>(64)</td>
<td>(557)</td>
</tr>
<tr>
<td>SLLC</td>
<td>96.90</td>
<td>93.25</td>
<td>100.00</td>
<td>75.94</td>
<td>87.36</td>
<td>96.55</td>
<td>94.08</td>
</tr>
<tr>
<td></td>
<td>(1.00)</td>
<td>(1.00)</td>
<td>(1.00)</td>
<td>(1.00)</td>
<td>(1)</td>
<td>(8)</td>
<td>(1)</td>
</tr>
<tr>
<td>LMNN</td>
<td>96.81</td>
<td>90.21</td>
<td>100.00</td>
<td>75.15</td>
<td>86.85</td>
<td>96.53</td>
<td>95.15</td>
</tr>
<tr>
<td></td>
<td>(9.98)</td>
<td>(13.30)</td>
<td>(8.73)</td>
<td>(69.71)</td>
<td>(156)</td>
<td>(82)</td>
<td>(591)</td>
</tr>
<tr>
<td>ITML</td>
<td>96.80</td>
<td>93.05</td>
<td>100.00</td>
<td>75.25</td>
<td>85.29</td>
<td>96.70</td>
<td>95.14</td>
</tr>
<tr>
<td></td>
<td>(9.79)</td>
<td>(18.01)</td>
<td>(15.21)</td>
<td>(16.40)</td>
<td>(287)</td>
<td>(49)</td>
<td>(206)</td>
</tr>
</tbody>
</table>
Experiments: projection space

KI (0.50196)

SLLC (1)

LMNN (0.85804)

ITML (0.50002)
## Experiments: $k$-NN

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

<table>
<thead>
<tr>
<th></th>
<th>Breast</th>
<th>Iono.</th>
<th>Pima</th>
<th>Splice</th>
<th>Svmsguide1</th>
<th>Cod-RNA</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K_1$</td>
<td>96.71</td>
<td>83.57</td>
<td>72.78</td>
<td>77.52</td>
<td>93.93</td>
<td>90.07</td>
</tr>
<tr>
<td>SLLC</td>
<td><strong>96.90</strong></td>
<td><strong>93.25</strong></td>
<td><strong>75.94</strong></td>
<td>87.36</td>
<td>93.82</td>
<td>94.08</td>
</tr>
<tr>
<td>LMNN</td>
<td>96.46</td>
<td>88.68</td>
<td>73.50</td>
<td><strong>87.59</strong></td>
<td>96.23</td>
<td>94.98</td>
</tr>
<tr>
<td>ITML</td>
<td>96.38</td>
<td>88.29</td>
<td>72.80</td>
<td>84.41</td>
<td><strong>96.80</strong></td>
<td><strong>95.42</strong></td>
</tr>
</tbody>
</table>
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.
Conclusion

Thank you!

Come to the poster for more details :-)
Backup slide 1: another projection space example
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.**

<table>
<thead>
<tr>
<th></th>
<th>Breast</th>
<th>Iono.</th>
<th>Rings</th>
<th>Pima</th>
<th>Splice</th>
<th>SVMguide1</th>
<th>Cod-RNA</th>
</tr>
</thead>
<tbody>
<tr>
<td>SLLC</td>
<td>4.76</td>
<td>5.36</td>
<td>0.05</td>
<td>4.01</td>
<td>158.38</td>
<td>185.53</td>
<td>2471.25</td>
</tr>
<tr>
<td>LMNN</td>
<td>25.99</td>
<td>16.27</td>
<td>37.95</td>
<td>32.14</td>
<td>309.36</td>
<td>331.28</td>
<td>10418.73</td>
</tr>
<tr>
<td>ITML</td>
<td>1.68</td>
<td>3.09</td>
<td>0.19</td>
<td>2.74</td>
<td>3.41</td>
<td>0.83</td>
<td>5.98</td>
</tr>
</tbody>
</table>
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.