Opera-Lib

A framework for operator-valued kernel

R. Brault[1] F. d'Alché Buc[2]

[1]IBISC (AROB@S)
Université d'Évry

[2]STatistiques et Applications
Institut Mines-Télécom

CDS OSI, 2015

Outline

Goals

Learning input/output vector/structured data, e.g.:

- Predict molecule activity on a line of diseases,
- Learning vector-field,
- Recommendation system,
- ...

How to use the information in the input/output and overtake independent prediction?

- Neural-Networks,
- Random forests,
- Kernel Methods (OVK).

Outline

A word about OVKs

Idea

Extend scalar kernels to vector-valued learning

	Scalar	Operator-Valued
Space	$k(x, x') \in \mathbb{R}$	$K(x, x') \in \mathcal{L}(\mathbb{R})$
Symmetry	$k(x, x') = k(x, x')$	$k(x, x') = k(x, x')^T$
Positivity	$\sum_{i,j} y_i k(x_i, x_j) y_j$	$\sum_{i,j} \langle y_i, k(x_i, x_j) y_j \rangle$

- Mercer theorem, representer theorem, universality... (Micchelli and Pontil [2005], Carmeli et al. [2010]).

Kernel engineering

Construction

- $K(x, x') = A$, A psd. $\rightarrow$ is an OVK.
- Let k_s be a scalar kernel and K be an OVK $\rightarrow$ $k_s(x, x')K(x, x')$ is an OVK.
- K_1, K_2 OVKs $\rightarrow$ $K1(x, x') + K2(x, x')$ is an OVK

Example

- Decomposable kernel: $K(x, x') = k_s(x, x')A$.
- Curl-free kernel: $K(x, x') = -\nabla\nabla^T k_s(x, x')$.
- Divergence-free kernel: $K(x, x') = (\nabla\nabla^T - \nabla^T\nabla)k_s(x, x')$.

Álvarez et al. [2012], Caponnetto et al. [2008], Macedo and Castro [2008].

Learning

Problem

$$\arg\min_{C} \sum_{i=1}^{n} \left\| \sum_{j=1}^{n} K(x_i, x_j) C_j - y_i \right\|^2 + \sum_{k=1}^{K} \lambda_k f_k(C) + \mu g(C)$$

.

f_k:

- Ridge.
- Lasso.
- Group-Lasso.
- Elastic-net.

Optimisation

$\rightarrow$ Proximal gradient descent.

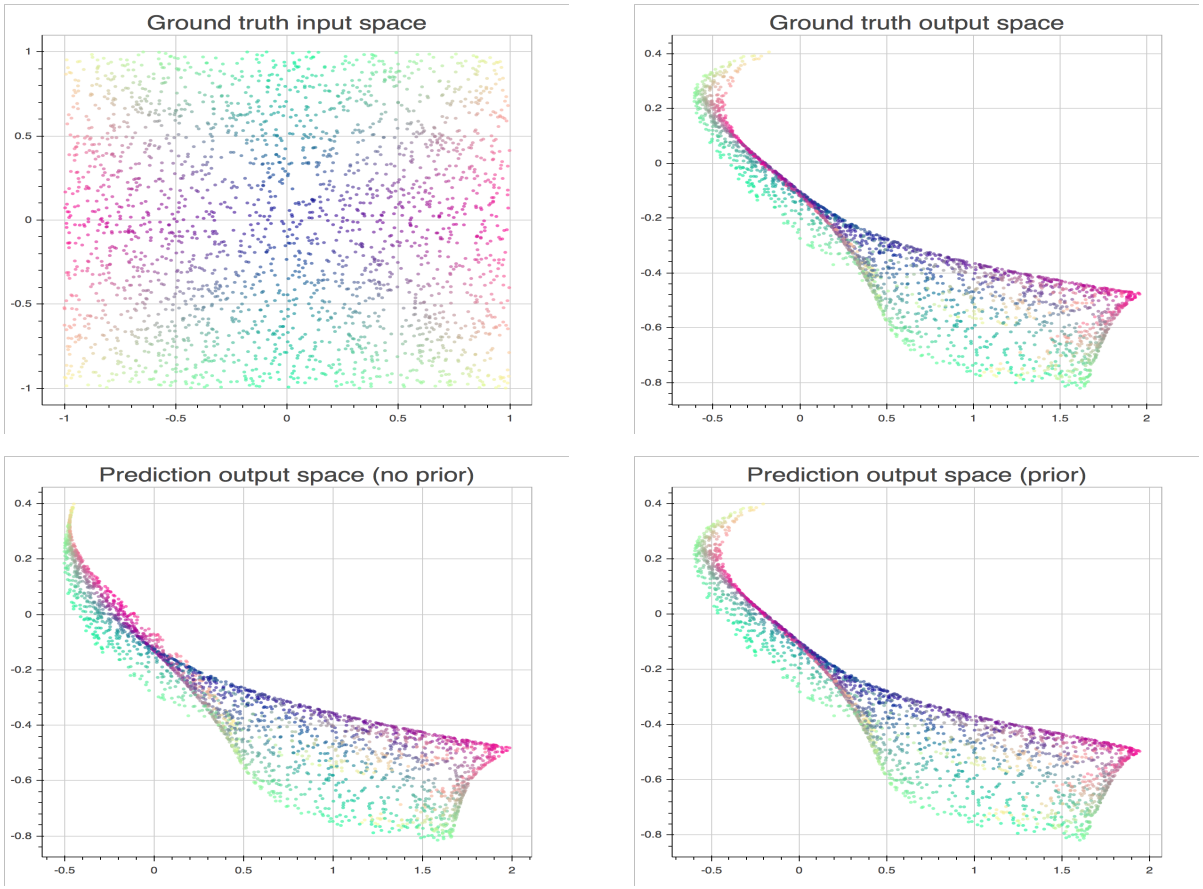
Outline

Sample Code

```
import operalib, operalib.Kernel
import operalib.Model, operalib.Risk
import operalib.Optimizer
```

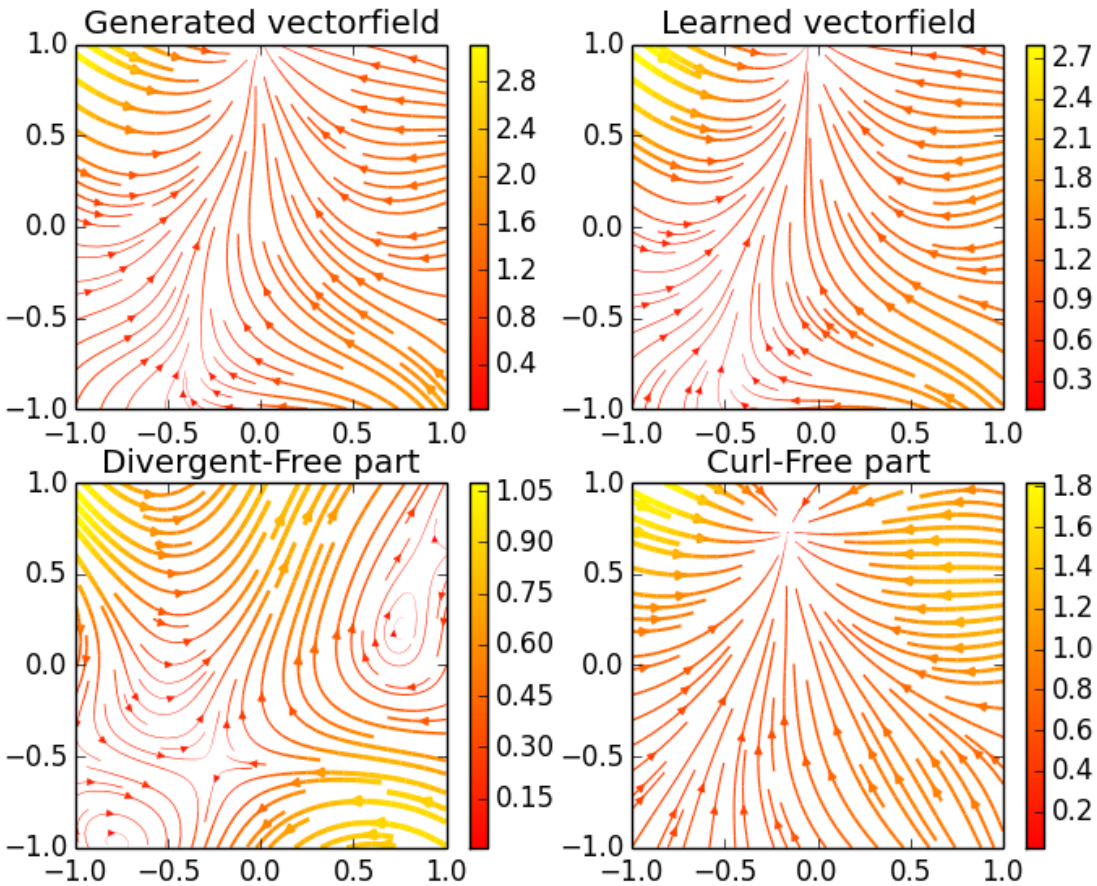
```
A = numpy.array( ground_truth )

k = operalib.Kernel.DecomposableKernel(gamma, d, A)
rsk = operalib.Risk.Ridge( 0 )
model = operalib.Model.Model(k, x)
L = operalib.Lipschitz_constant(model)
opt = operalib.Optimizer.FISTA(rsk, L, 1, 50)

opt.fit(model, X_train, y)
opt(X_test)
```

Results



10000 points, $\gamma = 0.5$, random psd A.
MSE with prior: $0.17 * 10^{-6}$,
MSE without prior: $0.26 * 10^{-3}$

Structure learning



Extract structure using a sum of curl-free / div-free kernel.

Outline

More structure

- POKR (Output kernel regression) (Brouard et al. [2011]).
- OKVAR (Autoregressive model) (Lim et al. [2015]).
- MKL (Multiple kernel) (Kadri et al. [2012, 2013]).
- Scaling the method to larger dataset (Audiffren and Kadri [2013]).

`https://github.com/RomainBrault/Operalib-2`

References I

Álvarez, M. A., Rosasco, L., and Lawrence, N. D. (2012). Kernels for vector-valued functions: a review. *Foundations and Trends in Machine Learning*, 4(3):195–266.

Audiffren, J. and Kadri, H. (2013). Online learning with multiple operator-valued kernels. *arXiv preprint arXiv:1311.0222*.

Brouard, C., d'Alché-Buc, F., and Szafranski, M. (2011). Semi-supervised penalized output kernel regression for link prediction. In *Proc. of the 28th Int. Conf. on Machine Learning*.

Caponnetto, A., Micchelli, C. A., , M., and Ying, Y. (2008). Universal multitask kernels. *Journal of Machine Learning Research*, 9:1615–1646.

Carmeli, C., De Vito, E., Toigo, A., and Umanit, V. (2010). Vector valued reproducing kernel hilbert spaces and universality. *Analysis and Applications*, 8:19–61.

References II

Kadri, H., Ghavamzadeh, M., and Preux, P. (2013). A generalized kernel approach to structured output learning. In *Proceedings of the 30th International Conference on Machine Learning*.

Kadri, H., Rakotomamonjy, A., Preux, P., and Bach, F. R. (2012). Multiple operator-valued kernel learning. In *Advances in Neural Information Processing Systems*, pages 2429–2437.

Lim, N., d'Alché-Buc, F., Auliac, C., and Michailidis, G. (2015). Operator-valued kernel-based vector autoregressive models for network inference. *Machine Learning*, 99(3):489–513.

Macedo, Y. and Castro, R. (2008). Learning div-free and curl-free vector fields by matrix-valued kernels. Technical report, Preprint A 679/2010 IMPA.

Micchelli, C. A. and Pontil, M. A. (2005). On learning vector-valued functions. *Neural Computation*, 17:177–204.