# Supervised Learning with Similarity Functions

Purushottam Kar[1] and Prateek Jain[2]

[1]Indian Institute of Technology, Kanpur, UP, INDIA
[2]Microsoft Research India, Bangalore, KA, INDIA

Microsoft Research

## Introduction

- **Goal** : Supervised learning with indefinite kernels
- Why use indefinite kernels ?
  - Several domains possess natural notions of similarity
    - Bioinformatics : B.L.A.S.T. scores for protein sequences
    - OCR : tangent distance similarity measures
    - Image retrieval : earth mover's distance
  - Satisfiability for Mercer's theorem a hard-to-verify property
  - Not clear why non psd-ness should limit usability of a kernel

## Existing work

- Most works address only the problem of classification
- Broadly three main approaches
  - Use indefinite kernels directly [1] : results in non-convex formulations
  - Find a proxy PSD kernel [2] : expensive + loss of domain knowledge
  - Use kernel-task alignment [3] : efficient + generalization guarantees
- Several results for classification using the third approach [3, 4, 5]

## Our contributions

- Propose a notion of kernel "goodness" for general supervised learning
  - Previous notions obtained as a special case
- Develop landmarking-based algorithms to perform supervised learning
  - Consider three tasks : real regression, ordinal regression, ranking
- Provide generalization bounds
- Apply sparse learning techniques to reduce landmark complexity
  - Fast testing times + generalization guarantees
- Experimental evaluation of landmarking based techniques

## What is a *good* similarity function

- Previously considered for classification by [3]
  - "Margin" view : positives closer to positives than negatives by a margin
  - Cannot be extended for other supervised learning problems
- We take a "target value" view
  - Target value at a point recoverable from neighbors of the point
  - Implicitly enforces a smoothness prior

**Definition 1. Good similarity function**

A similarity function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is $(\epsilon_0, B)$-good for a learning task $y : \mathcal{X} \to \mathcal{Y}$ if for some bounded weighing function $w : \mathcal{X} \to [-B, B]$, for at least a $(1 - \epsilon_0)$ fraction of the domain, we have $y(\mathbf{x}) = \underset{\mathbf{x}' \sim \mathcal{D}}{\mathbb{E}} [\![ w(\mathbf{x}')y(\mathbf{x}')K(\mathbf{x}, \mathbf{x}') ]\!]$ .

- Need to modify a bit to incorporate surrogate loss functions
- Can be adapted to various learning tasks using appropriate loss functions
- Reduces to earlier notion [3] for binary classification

## Evaluating the model

- The proposed notion of goodness is evaluated on two grounds
- **Utility** : "good" similarity functions should yield effective predictors

**Definition 2. Utility criterion**

A similarity function $K$ is $\epsilon_0$-useful w.r.t. a loss function $\ell(\cdot, \cdot)$ if for any $\epsilon_1 > 0$, using polynomially many labeled and unlabeled samples, one can w.h.p. generate a hypothesis $\hat{f}(\mathbf{x}; K)$ s.t. $\underset{\mathbf{x} \sim \mathcal{D}}{\mathbb{E}} \left[\!\!\left[ \ell\left(\hat{f}(\mathbf{x}), y(\mathbf{x})\right) \right]\!\!\right] \leq \epsilon_0 + \epsilon_1$.

- **Admissibility** : PSD kernels with large margin should remain "good"

**Definition 3. Good PSD Kernel**

A kernel $K$ with RKHS $\mathcal{H}_K$ and feature map $\Phi_K : \mathcal{X} \to \mathcal{H}_K$ is $(\epsilon_0, \gamma)$-good w.r.t. loss function $\ell_K$ if for some $\mathbf{W}^* \in \mathcal{H}_K$, we have $\underset{\mathbf{x} \sim \mathcal{D}}{\mathbb{E}} \left[\!\!\left[ \ell_K\left( \frac{\langle \mathbf{W}^*, \Phi_K(\mathbf{x}) \rangle}{\gamma}, y(\mathbf{x}) \right) \right]\!\!\right] < \epsilon_0$.

## Learning with similarity functions

**Algorithm 4. (Landmarking based learning algorithm)**

- **Given** : An $(\epsilon_0, B)$-good kernel $K$ and training points : $\mathcal{T} = \left\{(\mathbf{x}_i^t, y_i)\right\}_{i=1}^n$
- Sample $d$ unlabeled landmarks from domain : $\mathcal{L} = \left\{\mathbf{x}_1^l, \ldots, \mathbf{x}_d^l\right\}$
- Let $\Psi_{\mathcal{L}} : \mathbf{x} \mapsto \frac{1}{\sqrt{d}}\left(K(\mathbf{x}, \mathbf{x}_1^l), \ldots, K(\mathbf{x}, \mathbf{x}_d^l)\right) \in \mathbb{R}^d$
- Obtain $\hat{\mathbf{w}} := \underset{\mathbf{w} \in \mathbb{R}^d : \|\mathbf{w}\|_2 \leq B}{\arg\min} \sum_{i=1}^n \ell_S\left(\left\langle \mathbf{w}, \Psi_{\mathcal{L}}(\mathbf{x}_i^t)\right\rangle, y_i\right)$
- **Output** : $\hat{f} : \mathbf{x} \mapsto \langle \hat{\mathbf{w}}, \Psi_{\mathcal{L}}(\mathbf{x}) \rangle$

- Landmarks can be subsampled from training points themselves
  - Provide generalization guarantees for such "double-dipping"
- **Sparse Regression** : often only a small fraction of landmarks are useful
  - Landmark pruning essential for fast predictors
  - Propose modified model that takes into account only "useful" landmarks
  - Use sparse learning techniques [6] to learn a predictor
  - Utility guarantee ensures sparsity as well as generalization error bounds

## References

[1] Ong et al. Learning with non-positive Kernels. In *ICML*, 2004.

[2] Chen et al. Similarity-based Classification: Concepts and Algorithms. *JMLR*, 2009.

[3] Balcan and Blum. On a Theory of Learning with Similarity Functions. In *ICML*, 2006.

[4] Wang et al. On Learning with Dissimilarity Functions. In *ICML*, 2007.

[5] Kar and Jain. Similarity-based Learning via Data Driven Embeddings. In *NIPS*, 2011.

[6] Shalev-Shwartz et al. Trading Accuracy for Sparsity in Optimization Problems with Sparsity Constraints. *SIAM J. on Optimization*, 2010.

## Overview of theoretical guarantees

| Task | Utility | Samples required | Admissibility for $(\epsilon, \gamma)$-good kernel |
|---|---|---|---|
| Classification [3] | $(\epsilon, \gamma) \Rightarrow (\epsilon + \epsilon_1)$ Misclassification rate | $\mathcal{O}\left(\frac{1}{\gamma^2 \epsilon_1^2}\right)$ U + $\mathcal{O}\left(\frac{1}{\gamma^2 \epsilon_1^2}\right)$ L | $\left(\epsilon + \epsilon_1, \Theta\left(\epsilon_1 \gamma^2\right)\right)$ |
| Regression | $(\epsilon, B) \Rightarrow (B\epsilon + \epsilon_1)$ Mean squared error | $\mathcal{O}\left(\frac{B^2}{\epsilon_1^2}\right)$ U + $\mathcal{O}\left(\frac{B^2}{\epsilon_1^2}\right)$ L | $\left(\epsilon + \epsilon_1, \Theta\left(\frac{1}{\epsilon_1 \gamma^2}\right)\right)$ |
| Ordinal Regression | $(\epsilon, B, \Delta) \Rightarrow (\Psi_\Delta(\epsilon) + \epsilon_1)$ Absolute error | $\mathcal{O}\left(\frac{B^2}{\Delta^2 \epsilon_1^2}\right)$ U + $\mathcal{O}\left(\frac{B^2}{\Delta^2 \epsilon_1^2}\right)$ L | $(\epsilon, \gamma, \Delta)$-good $\Rightarrow$ $\left(\gamma_1 \epsilon + \epsilon_1, \Theta\left(\frac{\gamma_1^2}{\epsilon_1 \gamma^2}\right), \gamma_1 \Delta\right)$ |
| Ranking | $(\epsilon, B) \Rightarrow \mathcal{O}\left(\sqrt{\frac{m\epsilon}{\log m}} + \epsilon_1\right)$ NDCG loss | $\mathcal{O}\left(\frac{B^6 m^8}{\epsilon_1^4 \log^2 m}\right)$ U + $\mathcal{O}\left(\frac{B^6 m^4}{\epsilon_1^4 \log^2 m}\right)$ L | $\left(\epsilon + \epsilon_1, \mathcal{O}\left(\sqrt{\frac{m^3}{\epsilon_1^3 \gamma^6}}\right)\right)$ |

## Experimental results

| Datasets | Sigmoid kernel | | Manhattan kernel | |
|---|---|---|---|---|
| | KR | Land-Sp | KR | Land-Sp |
| Abalone $N = 4177$ | 2.1e-002 | 6.2e-003 | 1.7e-002 | 6.0e-003 |
| CAHousing $N = 20640$ | 5.9e-002 | 1.6e-002 | 5.8e-002 | 1.5e-002 |
| CPUData $N = 8192$ | 4.1e-002 | 1.4e-003 | 4.3e-002 | 1.2e-003 |
| PumaDyn-32 $N = 8192$ | 1.8e-001 | 1.4e-002 | 1.8e-001 | 1.4e-002 |

Table: MSE for real regression : Kernel regression vs. Sparse learning

| Datasets | Sigmoid kernel | | Manhattan kernel | |
|---|---|---|---|---|
| | KR | ORLand | KR | ORLand |
| Wine-Red $N = 1599$ | 6.8e-001 | 4.2e-001 | 6.7e-001 | 4.5e-001 |
| Wine-White $N = 4898$ | 6.2e-001 | 8.9e-001 | 6.2e-001 | 4.9e-001 |
| Bank-32 $N = 8192$ | 2.7e+000 | 1.6e+000 | 2.6e+000 | 1.6e+000 |
| House-16 $N = 22784$ | 2.7e+000 | 1.5e+000 | 2.8e+000 | 1.4e+000 |

Table: Absolute error for ordinal regression : Kernel regression vs. Landmarking

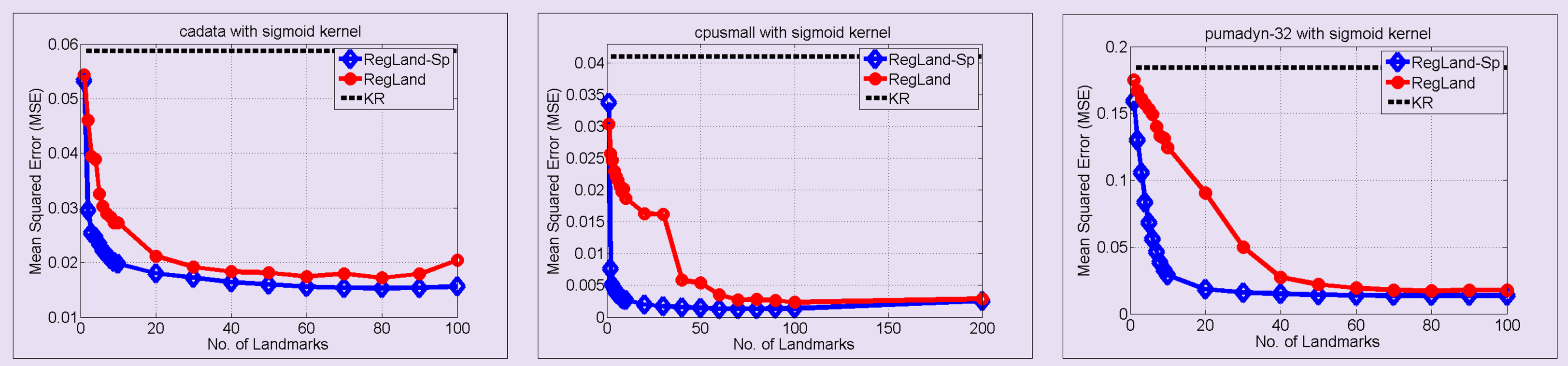


Figure : MSE for landmarking (RegLand), sparse landmarking (RegLand-Sp) and kernel regression (KR)

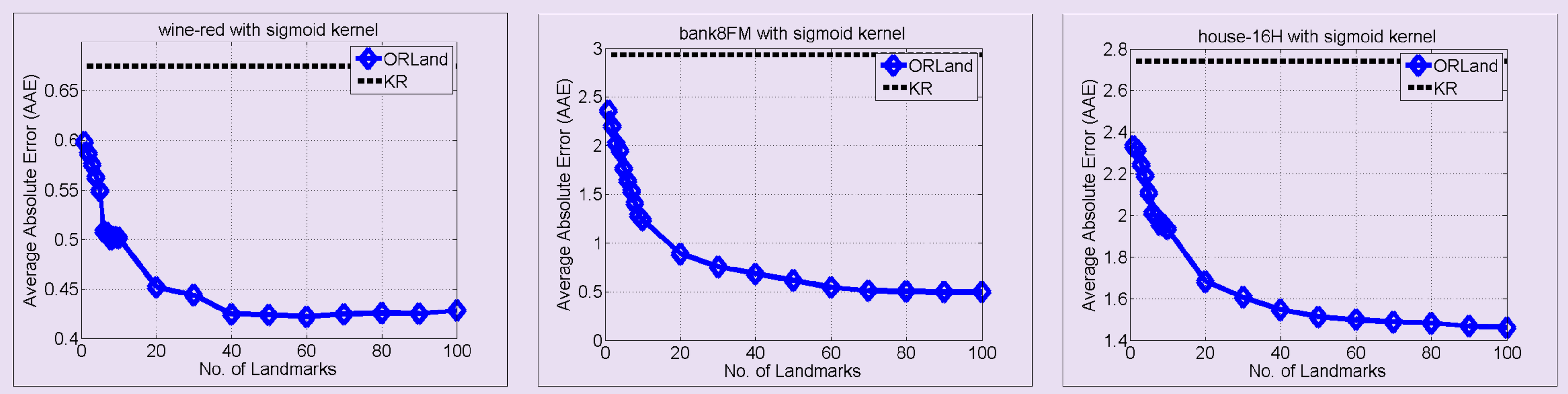


Figure : Absolute error for landmarking (ORLand) and kernel regression (KR)

26th Annual Conference on Neural Information Processing Systems (**NIPS 2012**)

**Full Paper : ArXiV : 1210.5840 [cs.LG], 2012.**