Gradient Weights help Nonparametric Regressors

Samory Kpotufe — Toyota Technological Institute, Chicago
Abdeslam Boularias — Max Planck Institute for Intelligent Systems, Tübingen

1 Overview

- Motivation: f does not depend equally on all dimensions, take advantage of this.
- Gradient-weighting applies to any distance-based regressor, is simple and efficient.
- Experiments on real-world problems show significant improvement in performance.

2 Preliminaries

Nonparametric regression setup:
Assume $Y = f(X) + \text{noise}$. Estimate $f(x)$ at x from sample $\{X_i, Y_i\}_1^n$.

Distance based regression:

$$f_n(x) = \sum_{i=1}^{n} w(x, X_i) \cdot Y_i, \text{ where } w(x, X_i) \text{ depends on metric } \rho \text{ (usually Euclidean).}$$

Ex: k-NN, kernel, local polynomial regressors.

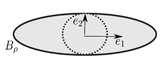
3 A simple idea:

f might not depend equally on all features of X!
Let $f'_i \equiv$ derivative along coordinate i, let $\|f'_i\|_{1,\mu} \equiv \mathbb{E}_X |f'_i(X)|$.
In practice $\left\{\|f'_i\|_{1,\mu}\right\}_{i \le d}$ has small and large elements.

Gradient weighting:
Reweight $x^i \to \sqrt{\mathbf{W}_i} \cdot x^i$, where $\mathbf{W}_i \approx \|f'_i\|_{1,\mu}$.
Equivalently, replace Euclidean distance with $\rho(x, x') = \sqrt{(x - x')^\top \mathbf{W} (x - x')}$.
Now run any local regressor f_n on $(\mathcal{X}, \rho)$.

Similar to metric learning, but cheaper: we estimate a single ρ!
More general than feature selection: f could depend on all $[d]$.

4 Intuition: suppose $\|f'_2\|_{1,\mu} \gg \|f'_1\|_{1,\mu}$.



Lower Var($f_n(x)$): suppose $\|f'_i\|_{1,\mu}$ is large only for $i \in R \subsetneq [d]$.
- Balls B_ρ contain more points relative to Euclidean balls.
- Formally, $\mu(B_\rho(x, \rho(\mathcal{X}) \cdot \epsilon)) \approx \epsilon^{|R|} \gg \epsilon^d$.

Bias of $f_n(x)$ is relatively unaffected:

- No sample is too far from x in any relevant direction.
- Formally, f has the following Lipschitz property:

$$|f(x) - f(x')| \le \left(\sum_{i \in R} \frac{|f'_i|_\infty}{\sqrt{\mathbf{W}_i}} \right) \rho(x, x').$$

5 Gradient norm estimator: estimate $\|f'_i\|_{1,\mu}$ and not f'_i

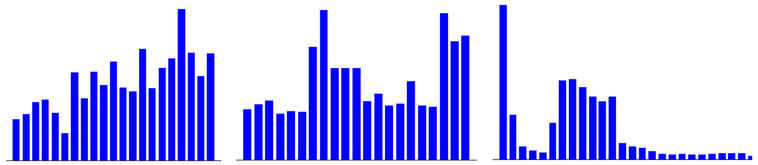
$$\mathbf{W}_i \triangleq \mathbb{E}_n \frac{|f_{n,h}(X + te_i) - f_{n,h}(X - te_i)|}{2t} \cdot 1_{\{A_{n,i}(X)\}},$$

where $A_{n,i}(X) \equiv$ we are confident in both estimates $f_{n,h}(X \pm te_i)$.

- Fast preprocessing, and online: just 2 estimates of $f_{n,h}$ at X. Metric learning optimizes over a space of possible metrics.
- Only $2 \times 2 \times 1$ param. search grid to adapt to d dimensions. KR with bandwidths $\{h_i\}_1^d$ needs $d \times d$ param. search grid.
- General: preprocessing for any distance-based regressor. Other methods apply to particular regressors, e.g. Rodeo for local linear regression, metric learning for KR.

6 Practicality of the approach

On typical real-world data, $\|f'_i\|_{1,\mu}$ varies a lot!



SARCOS robot's joint 7. Parkinson's (UCI repository). Telecom (UCI repository).

7 Consistency of the gradient weights estimator

Distributional assumptions:

- $\mathcal{X}$ has bounded diamter 1 and μ has mass everywhere on $\mathcal{X}$: $\forall x \in \mathcal{X}, \forall h > 0, \mu(B(x, h)) \ge C_\mu h^d$.
- f is continuously differentiable on the τ-envelope $\mathcal{X} + B(0, \tau)$, f'_i is uniformly continuous on $\mathcal{X} + B(0, \tau)$ and $\sup_{x \in \mathcal{X} + B(0,\tau)} |f'_i(x)| \le |f'_i|_{\text{sup}}$.

Theorem 1. *Under general regularity conditions on μ, and smoothness conditions on ∇f, we have with probability $\ge 1 - \delta$:*

$$\left| \mathbf{W}_i - \|f'_i\|_{1,\mu} \right| \le \frac{1}{t} \left(\sqrt{\frac{A(n)}{nh^d}} + h \cdot \sum_{i \in [d]} |f'_i|_{sup} \right) + 2 |f'_i|_{sup} \left(\sqrt{\frac{\ln 2d/\delta}{n}} + \mu\left(\partial_{t,i}(\mathcal{X})\right) \right) + \epsilon_{t,i}.$$

Boundary parameter: $\partial_{t,i}(\mathcal{X}) \triangleq \{x : \{x + te_i, x - te_i\} \not\subseteq \mathcal{X}\}$.
Smoothness parameter: $\epsilon_{t,i} \triangleq \sup_{x \in \mathcal{X},\, s \in [-t,t]} |f'_i(x) - f'_i(x + se_i)|$.

Note that, under distributional assumptions, $\mu(\partial_{t,i}(\mathcal{X})) \xrightarrow{t \to 0} 0$ and $\epsilon_{t,i} \xrightarrow{t \to 0} 0$. Thus, if for example $h = C/\log^2 n$, $t = \sqrt{h}$, we have $\mathbf{W}_i \xrightarrow{P} \|f'_i\|_{1,\mu}$ for all $i \in [d]$.

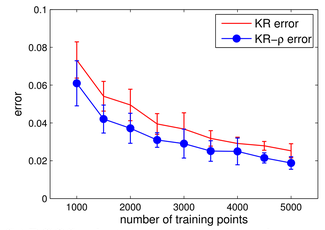
8 Experiments

We tested the performance of both Kernel Regression (KR) and k-NN without using the gradient weights, and with the gradient weights (KR-ρ and k-NN-ρ).

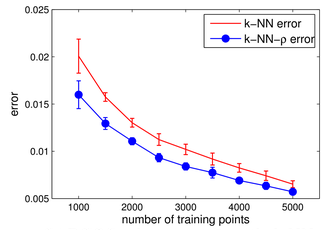
	Barrett joint 1	Barrett joint 5	SARCOS joint 1	SARCOS joint 5	Housing
KR error	0.50 ± 0.02	0.50 ± 0.03	0.16 ± 0.02	0.14 ± 0.02	0.37 ±0.08
KR-ρ error	**0.38**± 0.03	**0.35** ± 0.02	**0.14** ± 0.02	**0.12** ± 0.01	**0.25** ±0.06
KR time	0.39 ± 0.02	0.37 ± 0.01	0.28 ± 0.05	0.23 ± 0.03	0.10±0.01
KR-ρ time	0.41 ± 0.03	0.38 ± 0.02	0.32 ± 0.05	0.23 ± 0.02	0.11±0.01
	Concrete Strength	Wine Quality	Telecom	Ailerons	Parkinson's
KR error	0.42 ± 0.05	**0.75** ± 0.03	0.30±0.02	0.40±0.02	0.38±0.03
KR-ρ error	**0.37** ± 0.03	**0.75** ± 0.02	**0.23**±0.02	**0.39**±0.02	**0.34**±0.03
KR time	0.14 ± 0.02	0.19 ± 0.02	0.15±0.01	0.20±0.01	0.30±0.03
KR-ρ time	0.14 ± 0.01	0.19 ± 0.02	0.16±0.01	0.21±0.01	0.30±0.03

	Barrett joint 1	Barrett joint 5	SARCOS joint 1	SARCOS joint 5	Housing
k-NN error	0.41 ± 0.02	0.40 ± 0.02	0.08 ± 0.01	0.08 ± 0.01	0.28 ±0.09
k-NN-ρ error	**0.29** ± 0.01	**0.30** ± 0.02	**0.07** ± 0.01	**0.07** ± 0.01	**0.22**±0.06
k-NN time	0.21 ± 0.04	0.16 ± 0.03	0.13 ± 0.01	0.13 ± 0.01	0.08 ±0.01
k-NN-ρ time	0.13 ± 0.04	0.16 ± 0.03	0.14 ± 0.01	0.13 ± 0.01	0.08 ±0.01
	Concrete Strength	Wine Quality	Telecom	Ailerons	Parkinson's
k-NN error	0.40 ± 0.04	0.73 ± 0.04	**0.13**±0.02	0.37±0.01	0.22±0.01
k-NN-ρ error	**0.38** ± 0.03	**0.72** ± 0.03	0.17±0.02	**0.34**±0.01	**0.20**±0.01
k-NN time	0.10 ± 0.01	0.15 ± 0.01	0.16±0.02	0.12±0.01	0.14±0.01
k-NN-ρ time	0.11 ± 0.01	0.15 ± 0.01	0.15±0.01	0.11±0.01	0.15±0.01

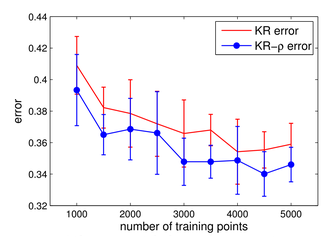
Normalized mean square prediction errors and average prediction time per point (in milliseconds).



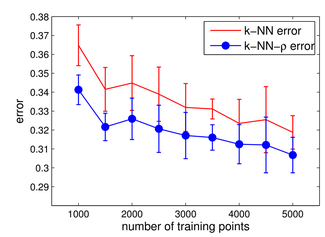
SARCOS robot, joint 7, with kernel regression



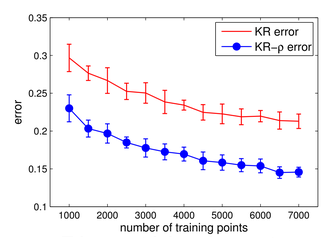
SARCOS robot, joint 7, with k-NN



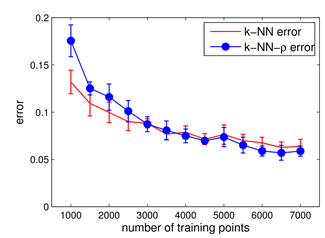
Ailerons with kernel regression



Ailerons with k-NN



Telecom with kernel regression



Telecom with k-NN

Normalized mean square prediction error over 2000 points for varying training sizes.

NIPS 2012, Lake Tahoe, December 2012