Gradient weights help nonparametric regression

Samory Kpotufe, Abdeslam Boularias

Toyota Technological Institute - Chicago
and Max Planck Institute for Intelligent Systems

Preliminaries

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

Preliminaries

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

A common way: local regression.

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

Usually ρ is the Euclidean distance in $\mathbb{R}^d$.
Ex: k-NN, kernel, local polynomial regressors.

Preliminaries

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

A common way: local regression.

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

Usually ρ is the Euclidean distance in $\mathbb{R}^d$.
Ex: k-NN, kernel, local polynomial regressors.

We present a simple way to significantly improve performance!

Motivation behind the approach:

f often varies more in some coordinates than in others.

Motivation behind the approach:

f often varies more in some coordinates than in others.

Define $f_i' \equiv$ derivative of f along coordinate $i \in [d]$.
The gradient norm $\|f_i'\|_1 \equiv \mathbb{E}_X |f_i'(X)|$ captures how relevant i is.

Motivation behind the approach:

f often varies more in some coordinates than in others.

Define $f_i' \equiv$ derivative of f along coordinate $i \in [d]$.
The gradient norm $\|f_i'\|_1 \equiv \mathbb{E}_X |f_i'(X)|$ captures how relevant i is.

In practice the norms $\|f_i'\|_1$ would differ accross coordinates $i \in [d]$.

Motivation behind the approach:

f often varies more in some coordinates than in others.

Define $f_i' \equiv$ derivative of f along coordinate $i \in [d]$.
The gradient norm $\|f_i'\|_1 \equiv \mathbb{E}_X |f_i'(X)|$ captures how relevant i is.

In practice the norms $\|f_i'\|_1$ would differ accross coordinates $i \in [d]$.

Gradient weighting:

Weigh each coordinate i according to unknown $\|f_i'\|_1$

Motivation behind the approach:

f often varies more in some coordinates than in others.

Define $f_i' \equiv$ derivative of f along coordinate $i \in [d]$.
The gradient norm $\|f_i'\|_1 \equiv \mathbb{E}_X |f_i'(X)|$ captures how relevant i is.

In practice the norms $\|f_i'\|_1$ would differ accross coordinates $i \in [d]$.

Gradient weighting:

Weigh each coordinate i according to unknown $\|f_i'\|_1$

... i.e., use a metric $\rho(x, x') = \sqrt{(x - x')^\top \mathbf{W}(x - x')}$, where the diagonal $\mathbf{W}_i \approx \|f_i'\|_1$.

Motivation behind the approach:
f often varies more in some coordinates than in others.

Define $f_i' \equiv$ derivative of f along coordinate $i \in [d]$.
The gradient norm $\|f_i'\|_1 \equiv \mathbb{E}_X |f_i'(X)|$ captures how relevant i is.

In practice the norms $\|f_i'\|_1$ would differ accross coordinates $i \in [d]$.

Gradient weighting:

Weigh each coordinate i according to unknown $\|f_i'\|_1$

... i.e., use a metric $\rho(x, x') = \sqrt{(x - x')^\top \mathbf{W}(x - x')}$, where the diagonal $\mathbf{W}_i \approx \|f_i'\|_1$.

Now run any local regressor f_n on $(\mathcal{X}, \rho)$.

Gradient weighting:

Gradient weighting:

Do regression on $(\mathcal{X}, \rho)$, where ρ depends on $\mathbf{W}_i \approx \|f_i'\|_1$.

Gradient weighting:

Do regression on $(\mathcal{X}, \rho)$, where ρ depends on $\mathbf{W}_i \approx \|f_i'\|_1$.

Similar to metric learning, but cheaper to estimate:

Avoid searching a space of possible metrics, just estimate ρ!

Gradient weighting:

Do regression on $(\mathcal{X}, \rho)$, where ρ depends on $\mathbf{W}_i \approx \|f_i'\|_1$.

Similar to metric learning, but cheaper to estimate:

Avoid searching a space of possible metrics, just estimate ρ!

Similar to feature selection, but works more generally:

Significant gains even when f depends on all features!
Little to no loss when all features are similarly relevant!

Why it works: local regression on $(\mathcal{X}, \rho)$.

Performance depends on regression variance and bias.

Why it works: local regression on $(\mathcal{X}, \rho)$.

Performance depends on regression variance and bias.
Variance decreases while bias is relatively unaffected.

Why it works: local regression on $(\mathcal{X}, \rho)$.

Performance depends on regression variance and bias.
Variance decreases while bias is relatively unaffected.

- Variance is controlled by the mass of balls on $(\mathcal{X}, \rho)$:
  We show that balls in $(\mathcal{X}, \rho)$ have higher mass.

Why it works: local regression on $(\mathcal{X}, \rho)$.

Performance depends on regression variance and bias.
Variance decreases while bias is relatively unaffected.

- Variance is controlled by the mass of balls on $(\mathcal{X}, \rho)$:
  We show that balls in $(\mathcal{X}, \rho)$ have higher mass.
- Bias is controlled by the smoothness of f in $(\mathcal{X}, \rho)$:
  We show that smoothness properties of f are maintained.

Efficient estimation of $\|f_i'\|_1$: don't estimate f_i'.

Take average of differences of $f_{n,h} \equiv$ initial rough estimate of f:

Efficient estimation of $\|f_i'\|_1$: don't estimate f_i'.

Take average of differences of $f_{n,h} \equiv$ initial rough estimate of f:

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

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

Efficient estimation of $\|f_i'\|_1$: don't estimate f_i'.

Take average of differences of $f_{n,h} \equiv$ initial rough estimate of f:

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

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.

Efficient estimation of $\|f_i'\|_1$: don't estimate f_i'.

Take average of differences of $f_{n,h} \equiv$ initial rough estimate of f:

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

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.
- Just two parameters t and h allow tuning to d dimensions.

Efficient estimation of $\|f_i'\|_1$: don't estimate f_i'.

Take average of differences of $f_{n,h} \equiv$ initial rough estimate of f:

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

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.
- Just two parameters t and h allow tuning to d dimensions.
- General: preprocessing for any distance-based regressor.

$\mathbf{W}_i$ consistently estimates $\|f_i'\|_1$

Theorem

Under general regularity conditions on μ, and uniform continuity of ∇f,

$$\mathbf{W}_i \xrightarrow{P} \|f_i'\|_1 \text{ provided } t \to 0,\ h/t \to 0 \text{ and } nh^d t^2 \to \infty.$$

$\mathbf{W}_i$ consistently estimates $\|f_i'\|_1$

Theorem

Under general regularity conditions on μ, and uniform continuity of ∇f,

$$\mathbf{W}_i \xrightarrow{P} \|f_i'\|_1 \text{ provided } t \to 0,\ h/t \to 0 \text{ and } nh^d t^2 \to \infty.$$

Finite sample bounds provide guidance into tuning t and h.

$\mathbf{W}_i$ consistently estimates $\|f_i'\|_1$

Theorem

Under general regularity conditions on μ, and uniform continuity of ∇f,

$$\mathbf{W}_i \xrightarrow{P} \|f_i'\|_1 \text{ provided } t \to 0,\ h/t \to 0 \text{ and } nh^d t^2 \to \infty.$$

Finite sample bounds provide guidance into tuning t and h.

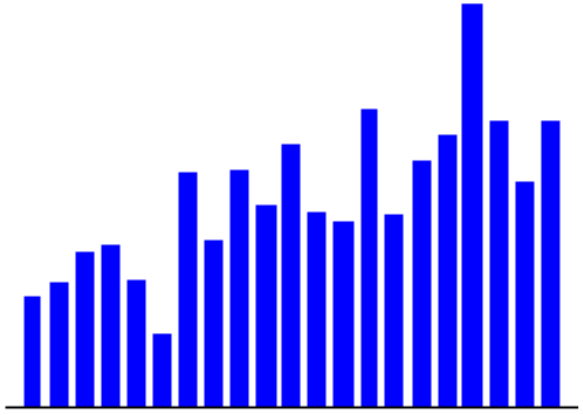
Main technical hurdle: Behavior of $\mathbf{W}_i$ at boundary of $\mathcal{X}$.

Significant performance improvement in practice ...

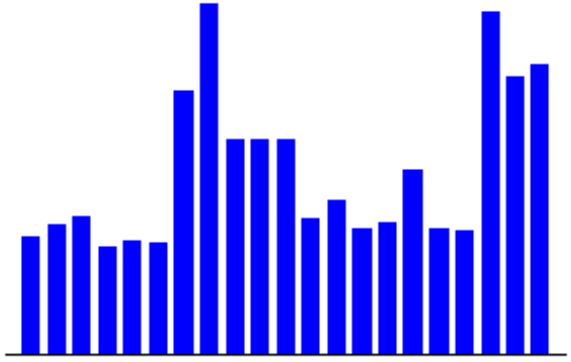
Significant performance improvement in practice ...

Datasets: from robotics, material science, agriculture, telecommunication, medicine, ...

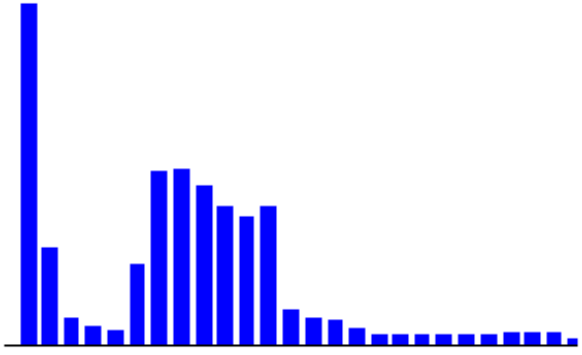
On real-world data, $\|f_i'\|_1$ varies accross $i \in [d]$!



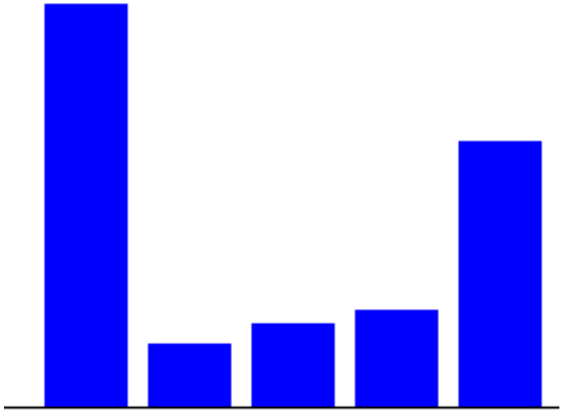
SRCS robot joint7.



Parkinson's.



Telecom.



Ailerons.

Results on real world datasets, kernel regression.

	Barrett joint 1	Barrett joint 5	SARCOS joint 1	SARCOS joint 5	Housing
KR error	0.50 $\pm$ 0.02	0.50 $\pm$ 0.03	0.16 $\pm$ 0.02	0.14 $\pm$ 0.02	0.37 $\pm$0.08
KR-ρ error	**0.38**$\pm$ 0.03	**0.35** $\pm$ 0.02	**0.14** $\pm$ 0.02	**0.12** $\pm$ 0.01	**0.25** $\pm$0.06

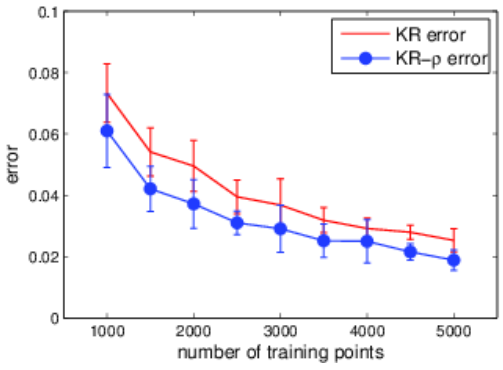
	Concrete Strength	Wine Quality	Telecom	Ailerons	Parkinson's
KR error	0.42 $\pm$ 0.05	**0.75** $\pm$ 0.03	0.30$\pm$0.02	0.40$\pm$0.02	0.38$\pm$0.03
KR-ρ error	**0.37** $\pm$ 0.03	**0.75** $\pm$ 0.02	**0.23**$\pm$0.02	**0.39**$\pm$0.02	**0.34**$\pm$0.03

Results on real world datasets, k-NN regression.

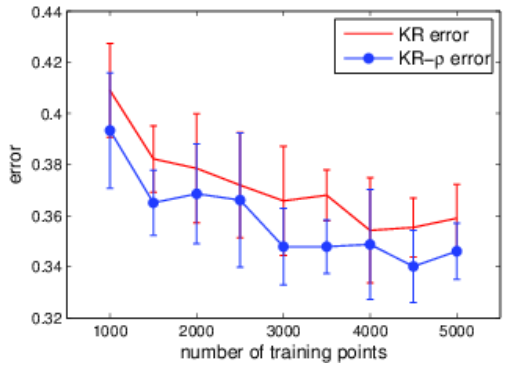
	Barrett joint 1	Barrett joint 5	SARCOS joint 1	SARCOS joint 5	Housing
k-NN error	0.41 $\pm$ 0.02	0.40 $\pm$ 0.02	0.08 $\pm$ 0.01	0.08 $\pm$ 0.01	0.28 $\pm$0.09
k-NN-ρ error	**0.29** $\pm$ 0.01	**0.30** $\pm$ 0.02	**0.07** $\pm$ 0.01	**0.07** $\pm$ 0.01	**0.22**$\pm$0.06

	Concrete Strength	Wine Quality	Telecom	Ailerons	Parkinson's
k-NN error	0.40 $\pm$ 0.04	0.73 $\pm$ 0.04	**0.13**$\pm$0.02	0.37$\pm$0.01	0.22$\pm$0.01
k-NN-ρ error	**0.38** $\pm$ 0.03	**0.72** $\pm$ 0.03	0.17$\pm$0.02	**0.34**$\pm$0.01	**0.20**$\pm$0.01

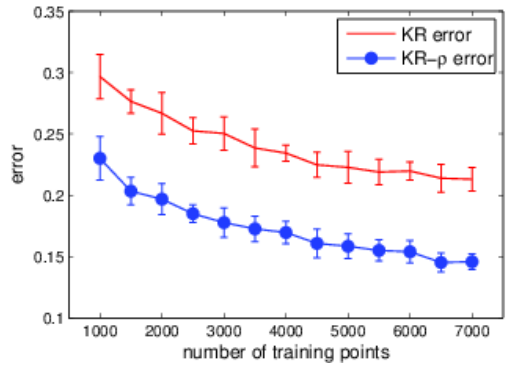
MSE, varying training size, kernel regression



SRCS joint 7, KR

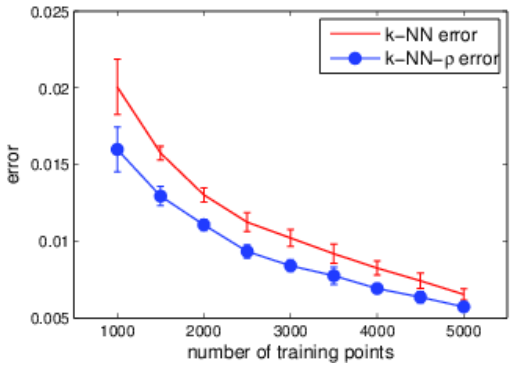


Ailerons, KR

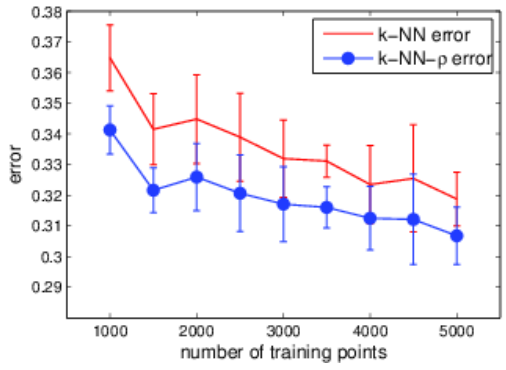


Telecom with KR

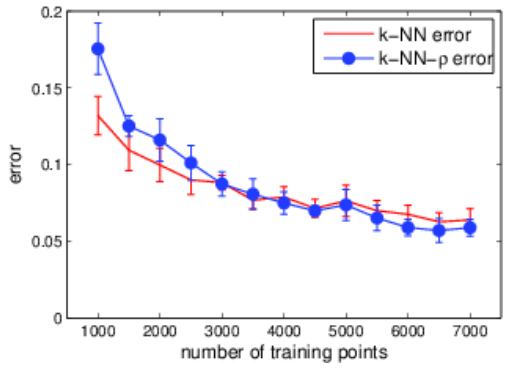
MSE, varying training size, k-NN regression



SRCS joint 7, k-NN



Ailerons, k-NN



Telecom, k-NN

Take home message:

Gradient weights help nonparametric regressors!

Take home message:

Gradient weights help nonparametric regressors!

Simple idea $\Longrightarrow$ much room for improvement!

Take home message:

Gradient weights help nonparametric regressors!

Simple idea $\Longrightarrow$ much room for improvement!

Easy to implement, so I hope you try it! 

Thanks!