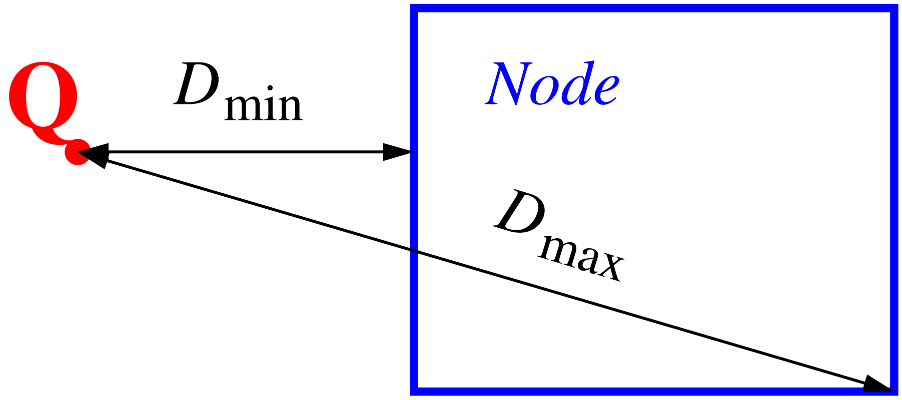
Cutoff Criterion

You're about to compute $\sum_k w_k^2 \mathbf{x}_k^T \mathbf{x}_k$ for a node .



You know that all weights inside the node must lie in the range

$$w_{min} = weight\ (D_{max}) <= w_k <= weight\ (D_{min}) = w_{max}$$

So if w_{max} and w_{min} are very similar , then we know that

$$\sum_k w_k^2 \mathbf{x}_k^T \mathbf{x}_k = (w_{max} + w_{min})/2 \ * \sum_k \mathbf{x}_k^T \mathbf{x}_k$$

That information is inside the current node .

w_{max} and w_{min} are very similar when (1) the weight function is flat or (2) the node is small.

Cutoff Rule

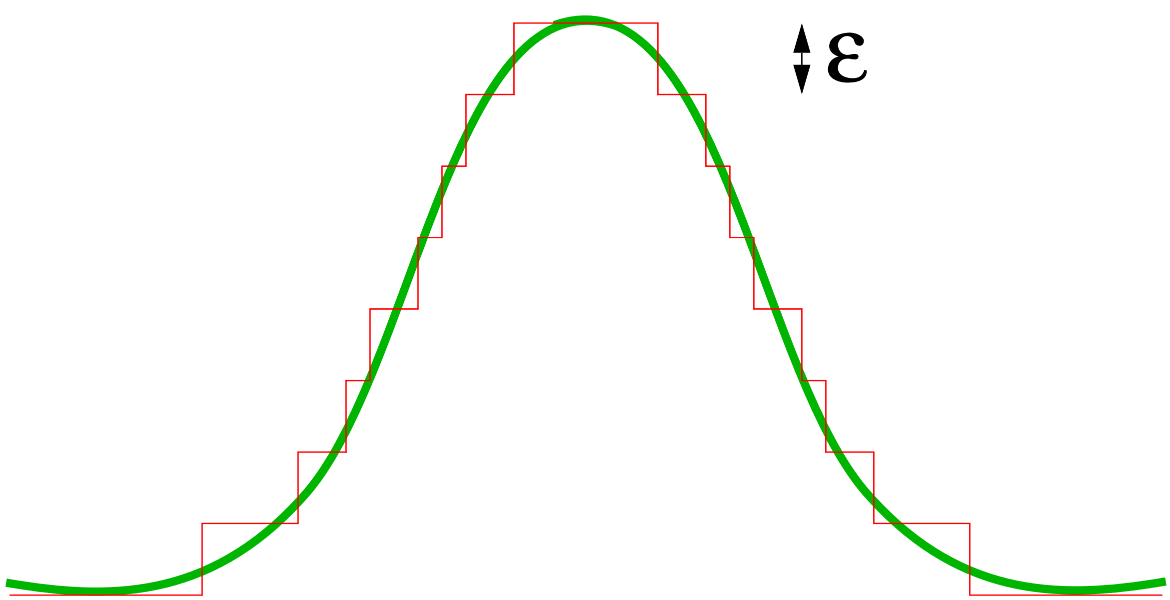
To compute $\sum_k w_k^2 \mathbf{x}_k^T \mathbf{x}_k$ for a non-leaf node:

- **If** $w_{\max} - w_{\min} < \varepsilon$ then return

$$(w_{\max} + w_{\min})/2 \ * \ \sum_k \mathbf{x}_k^T \mathbf{x}_k$$

- **Else**

  recursively compute $\sum_k w_k^2 \mathbf{x}_k^T \mathbf{x}_k$ for left and right children and add them.



This is equivalent to using some approximate weighting function that never deviates from desired weighting function by more than ε

One final little detail...

- $\sum_k w_k^{\,2}$ can get as small as $O(\varepsilon)$ if we are far from data.
- So approximations of $O(\varepsilon)$ can become dangerously significant.

  Thus we alter our criterion to:

  Cutoff if $w_{\max} - w_{\min} < \tau W_{TOTAL}$

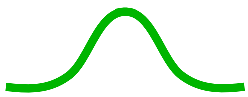
  where τ is a small constant and where W_{TOTAL} is the total sum of squared weights of all datapoints in this query.

- W_{TOTAL} is not known in advance, but has a lower bound W_{SOFAR}. So in actuality ...

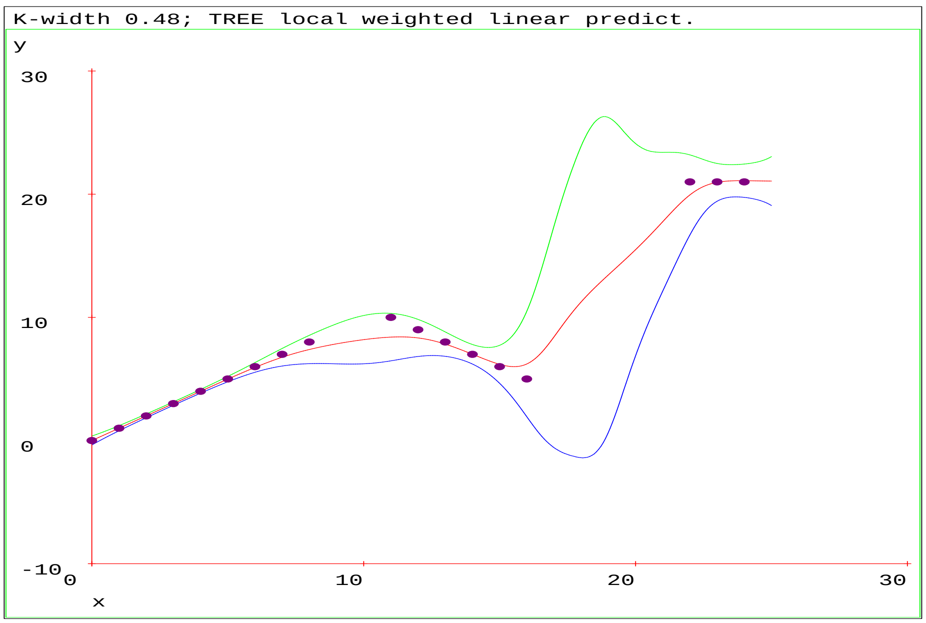
  Cutoff if $w_{\max} - w_{\min} < \tau W_{SOFAR}$

Slight Approximation

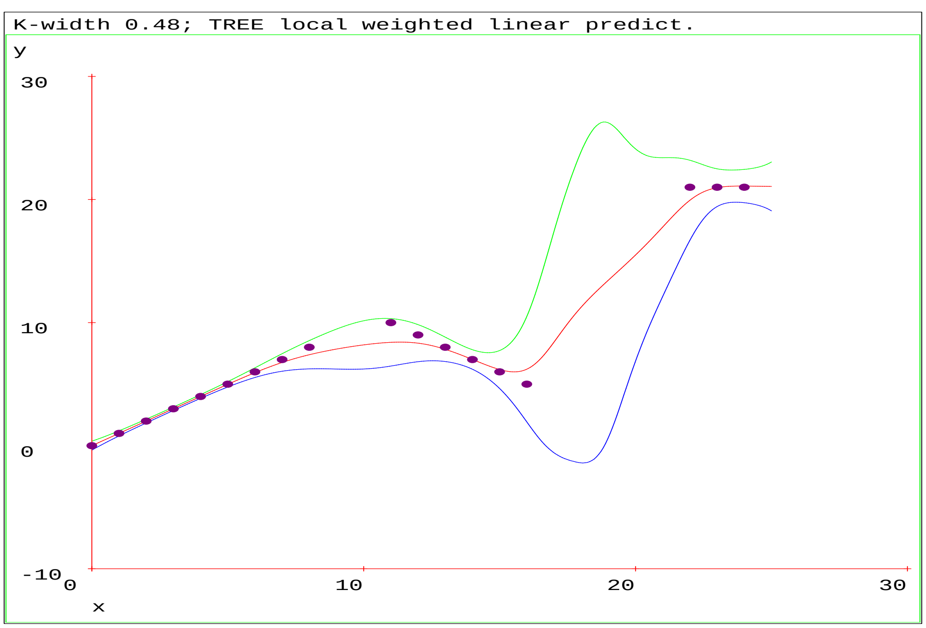
Examine the difference for locally weighted linear regression.



KW = 4



The linear , exact, method.

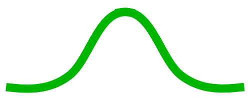


Multires , using $\tau = 10^{-7}$

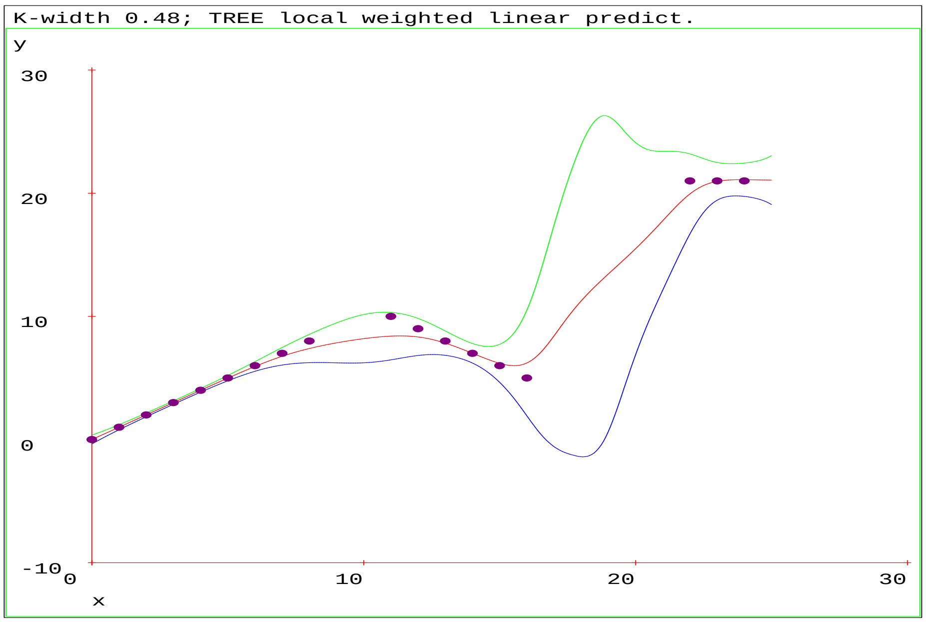
..it's identical

Medium Approximation

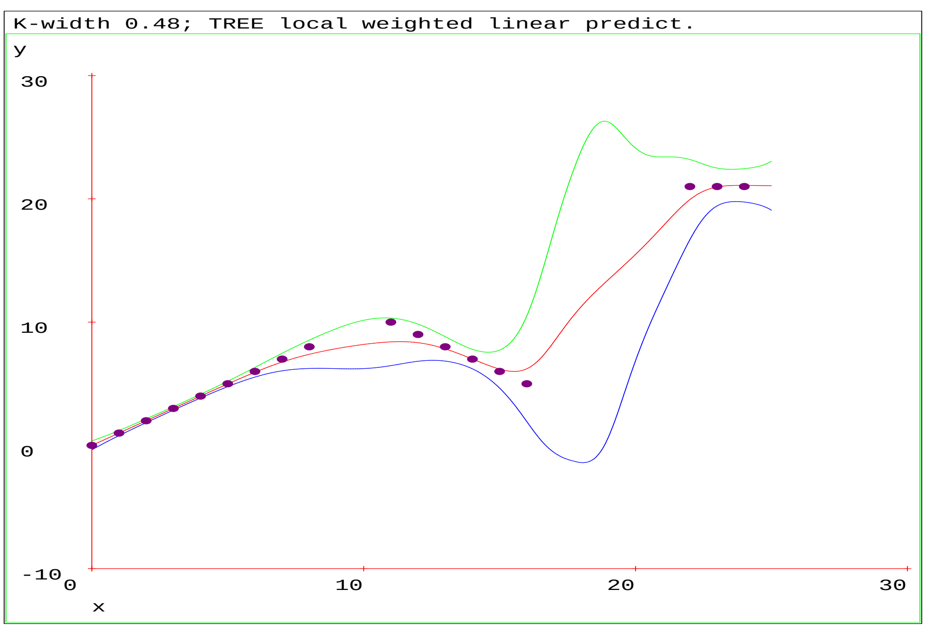
Examine the difference for locally weighted linear regression.



KW = 4



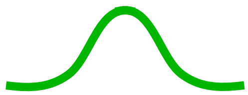
The linear , exact, method.



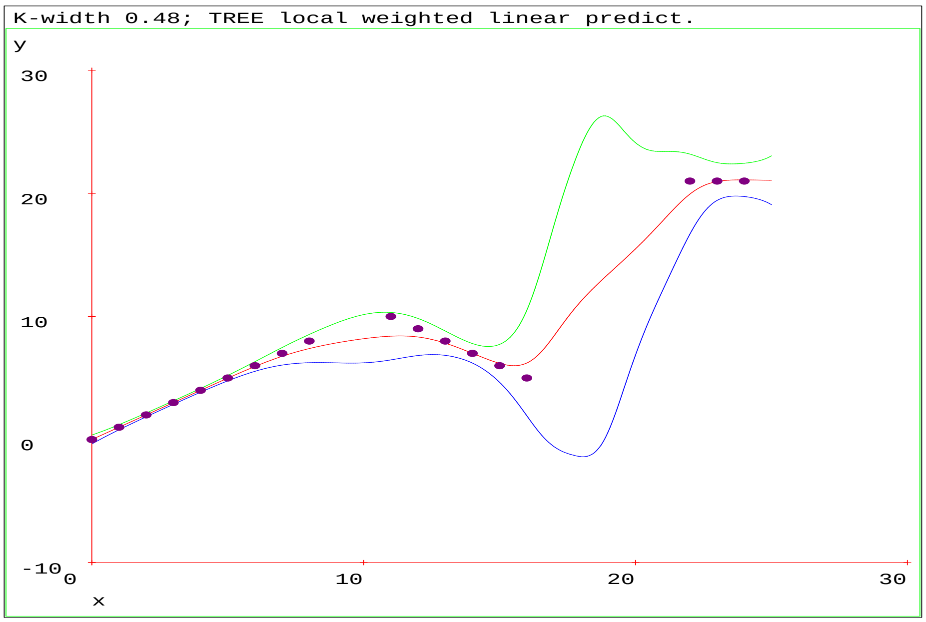
Multires , using $\tau = 0.05$

..it's indistinguishable

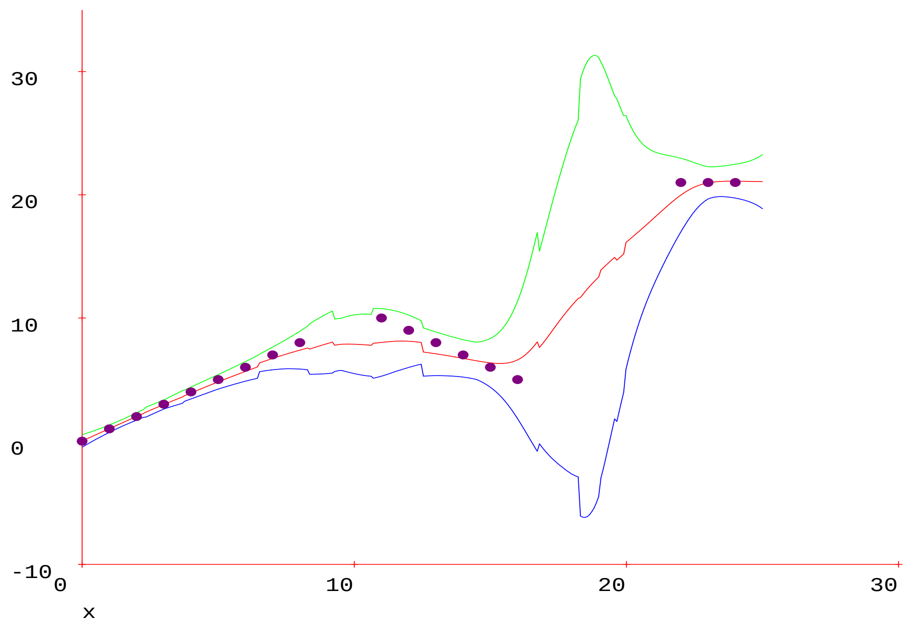
Gross Approximation



KW = 4



The linear , exact, method.



Multires , using $\tau = 0.5$

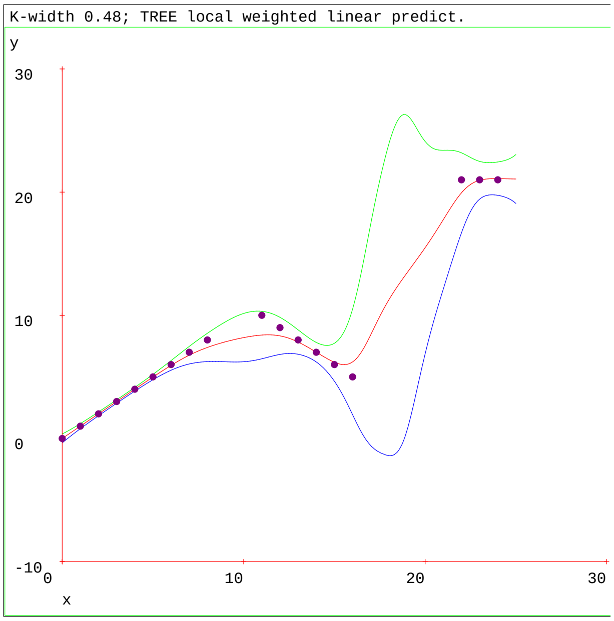
..it's close but with tiny discont- inuities

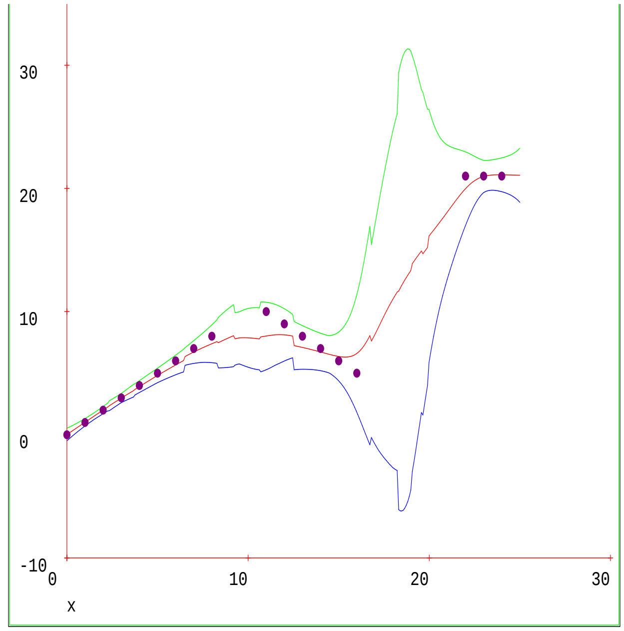
Do the discontinuities pose a danger for algorithms that compute derivatives?

Gradient Estimation

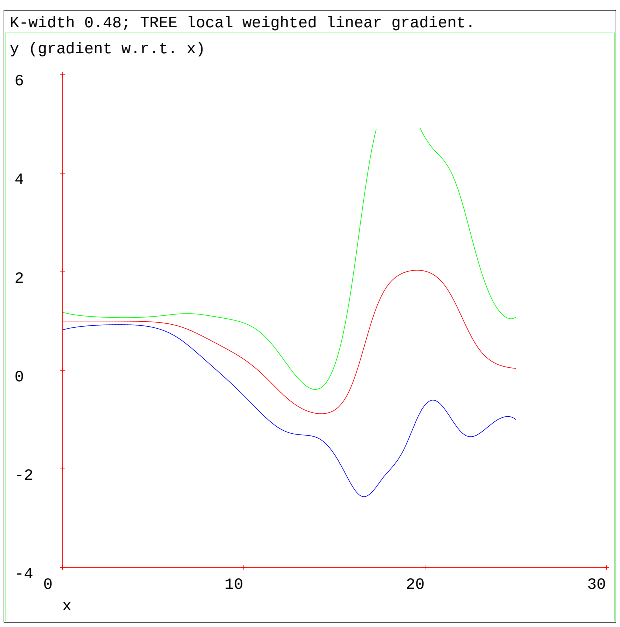
Exact Method | **τ =0.5 fast method**

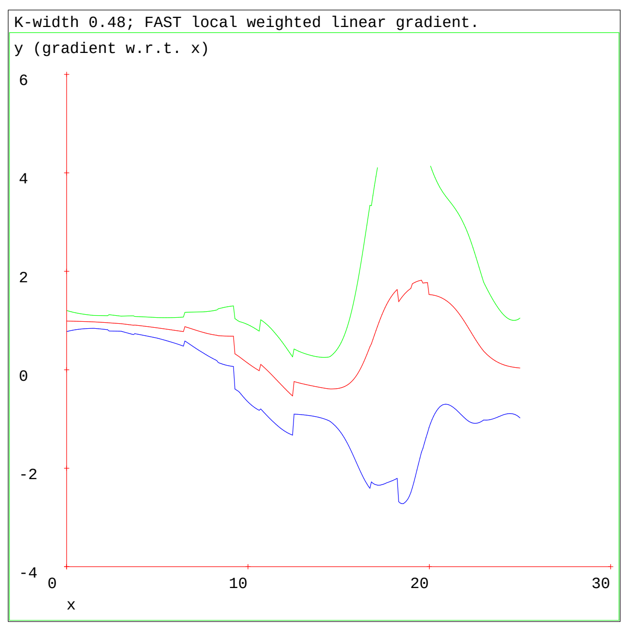
Predicted value





Predicted gradient





Gradients are estimated from regression. Not by numerical differencing .

Speed and Accuracy

Using locally weighted regression with kernel width 1/32 in unit-scaled coordinates .

		Expense	Error
heart.mbl	Regular	42.16	0.27
L40:{9}	RegZero	32.95	0.28
13 inputs	Tree	21.23	0.28
170 datapoints	Approx	18.93	0.28
StdError 0.43	Fast	14.12	0.28
pool.mbl	Regular	34.65	0.63
L40:{9}	RegZero	33.45	0.63
3 inputs	Tree	22.33	0.63
153 datapoints	Approx	4.41	0.63
StdError 2.2140	Fast	0.80	0.62
energy.mbl	Regular	535.87	11.93
L40:{9}	RegZero	484.30	11.93
5 inputs	Tree	323.37	11.93
2344 datapoints	Approx	5.11	15.15
StdError 286.07	Fast	1.10	21.60
abalone.mbl	Regular	964.00	1.65
L40:{9}	RegZero	806.00	1.65
10 inputs	Tree	469.00	1.65
4077 datapoints	Approx	5.80	1.67
StdError 2.66	Fast	1.70	1.71
mpg.mbl	Regular	70.10	1.92
L40:{9}	RegZero	55.18	1.92
9 inputs	Tree	34.35	1.92
292 datapoints	Approx	11.61	1.92
StdError 6.8151	Fast	2.00	1.93
breast-all.mbl	Regular	143.40	0.03
L40:{9}	RegZero	126.18	0.03
9 inputs	Tree	59.88	0.03
599 datapoints	Approx	13.82	0.03
StdError 0.3	Fast	6.21	0.02

Regular : standard linear method. RegZero: Linear method, but ignores zero-weights . Tree: Multires with $\tau = 1\text{e-}7$. Approx: $\tau = 0.05$. Fast: $\tau = 0.5$. Expense is seconds for 100 predictions .

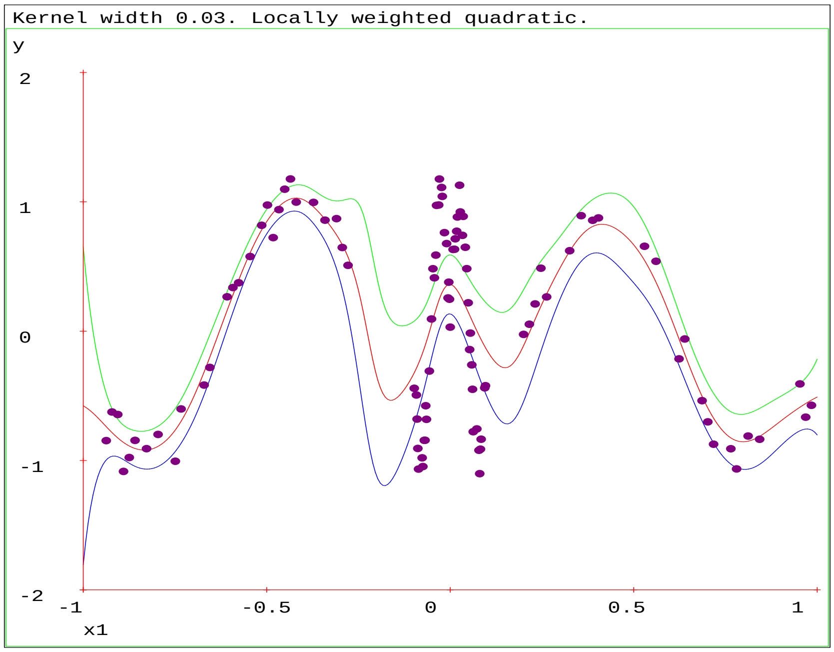
Speed and Accuracy

Using dataset-specific L WPR model selected by cross-validation.

		Expense	Error
heart.mbl	Regular	37.86	0.22
A23:{9}-[5]	RegZero	25.84	0.22
13 inputs	Tree	14.32	0.22
170 datapoints	Approx	13.42	0.22
StdError 0.43	Fast	0.50	0.24
pool.mbl	Regular	36.05	0.63
Q50:999	RegZero	35.95	0.63
3 inputs	Tree	25.43	0.63
153 datapoints	Approx	8.12	0.63
StdError 2.2140	Fast	1.20	0.62
energy.mbl	Regular	546.48	6.12
E29:{9}	RegZero	356.12	6.12
5 inputs	Tree	202.29	6.12
2344 datapoints	Approx	25.53	6.03
StdError 286.07	Fast	1.60	7.50
abalone.mbl	Regular	958.90	1.33
L30:999900090-	RegZero	717.34	1.33
10 inputs	Tree	203.91	1.33
4077 datapoints	Approx	2.35	1.33
StdError 2.6611	Fast	1.40	1.34
mpg.mbl	Regular	66.79	1.95
L40:900009090	RegZero	54.18	1.95
9 inputs	Tree	8.41	1.95
292 datapoints	Approx	1.70	1.94
StdError 6.8151	Fast	1.20	1.92
breast-all.mbl	Regular	44.06	0.01
A01:99--99-9-	RegZero	43.96	0.01
9 inputs	Tree	2.20	0.01
599 datapoints	Approx	2.20	0.01
StdError 0.3	Fast	0.50	0.02

Regular : standard linear method. RegZero: Linear method, but ignores zero-weights . Tree: Multires with $\tau = 1e\text{-}7$. Approx: $\tau = 0.05$. Fast: $\tau = 0.5$. Expense is seconds for 100 predictions .

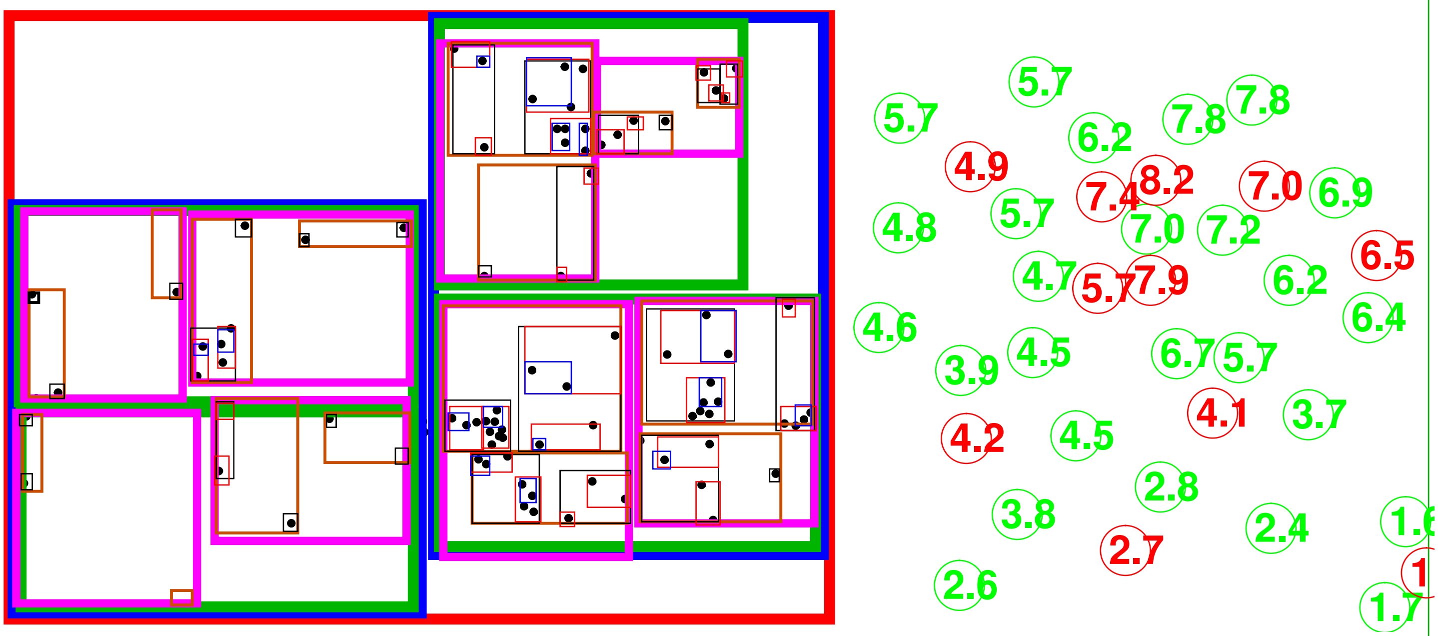
Variable kernel-widths



Using Fixed Kernel width	Using Variable K-width	Using Variable K-width Goal Weight 8.0

K-width	Mean Error	Goal Weight	Mean Error	Method	Mean Error	MSec / Predict
0.25000	0.41	64	0.19			
0.12500	0.24	32	0.13	Linear	0.104	1400
0.06250	0.24	16	0.11	$\tau=10^{-7}$	0.104	395
0.03125	0.22	8	0.10	$\tau=0.05$	0.103	181
0.01562	0.29	4	0.10	$\tau=0.5$	0.107	165
0.00781	0.37	2	0.11			
0.00391	0.41	1	0.15			
0.00195	0.51	0.5	0.51			

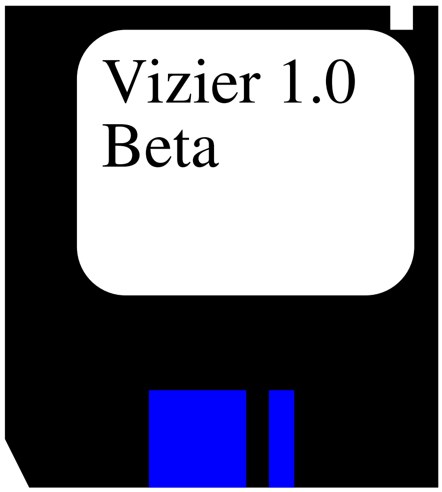
Multiresolution versus Prototypes



They are really not that different.

Nodes in tree can be seen as prototypes summarizing all the instances below them.

Please try this out!



(GUI runs under WindowsNT, Windows95)

See Andrew Moore or Jeff Schneider for a beta test copy

Vizier 1.0 is an experimental Process Data Analysis System.

- Implementations of a range of locally weighted and memory based algorithms.
- Cross-validation-based tuning and model selection.
- Locally weighted Bayesian regression analysis.
- All based on the multiresolution-tree method.

Comments

- Multiresolution trees make LWPR much faster without significant sacrifices.
- But predictions are still not free!
- This work extends [Deng+Moore 1995]'s kernel regression work.
- Similar multiresolution tricks are used in computer graphics and galaxy simulations
- Ongoing work plays similar tricks for local classification algorithms:
  e.g. Locally weighted logistic regression [Deng+Moore, 1997]
- Note that with large datasets, the leaves of the tree are rarely visited...

Idea: Place the bottom levels of the tree in secondary memory (disk array) and maybe cope with billions of examples?