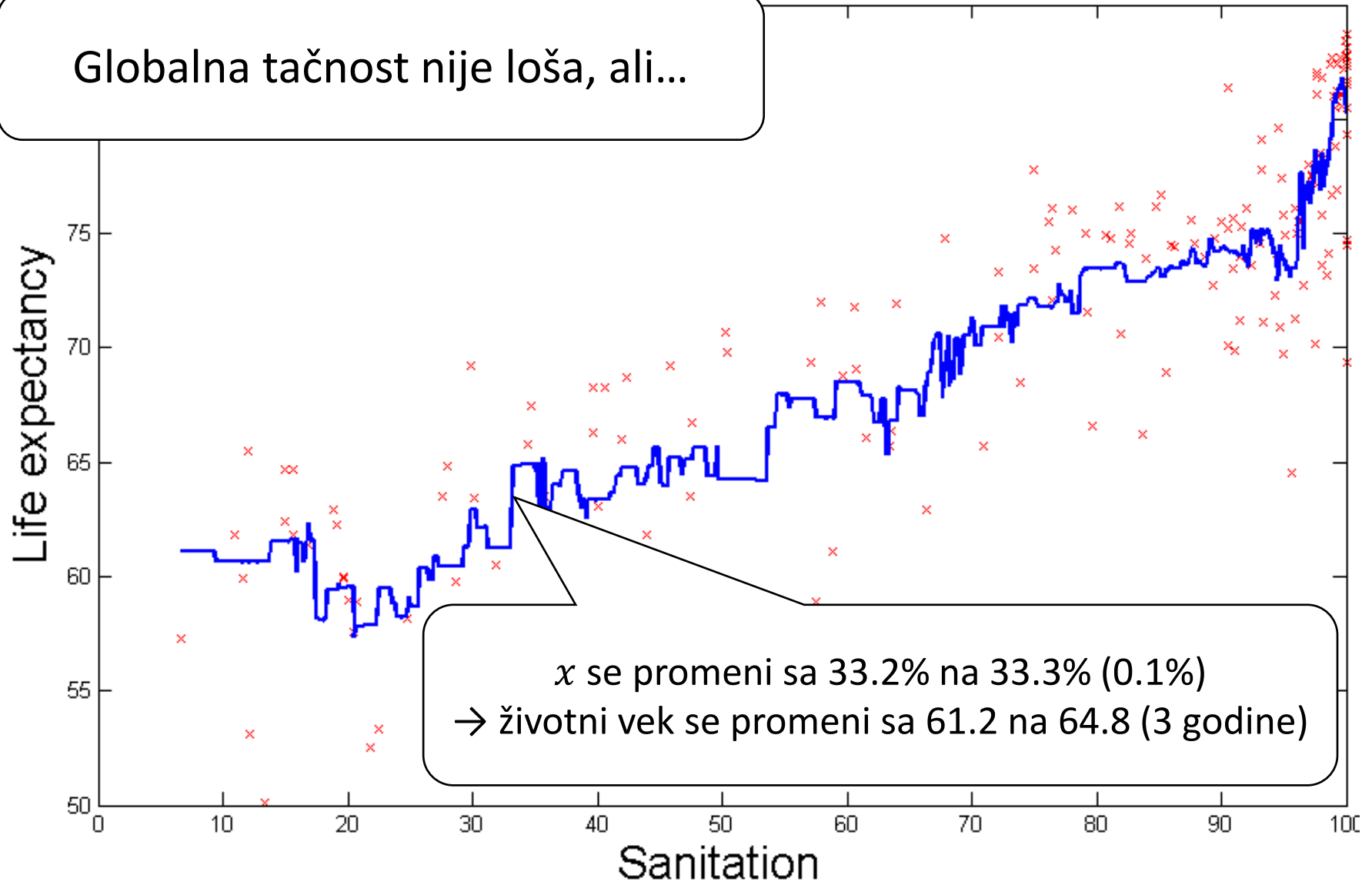


Problem sa diskontinuitetom

Globalna tačnost nije loša, ali...



Problem sa diskontinuitetom

- Rešenje je uvođenje težinskih faktora udaljenosti (*weighted k-NN*)
- Ideja: pridružiti veću težinu opservacijama koje su bliže (sličnije) tački $x^{(q)}$ u odnosu na opservacije koje su joj manje slične

$$\hat{y}^{(q)} = \frac{w^{(qNN1)}y^{NN1} + w^{(qNN2)}y^{NN2} + \dots + w^{(qNNk)}y^{NNk}}{\sum_{j=1}^k w^{(qNNj)}}$$

- Efekat: kada jedna od opservacija izađe iz „prozora, vrednost se neće mnogo izmeniti jer je ta opservacija već imala nisku težinu prilikom formiranja predikcije

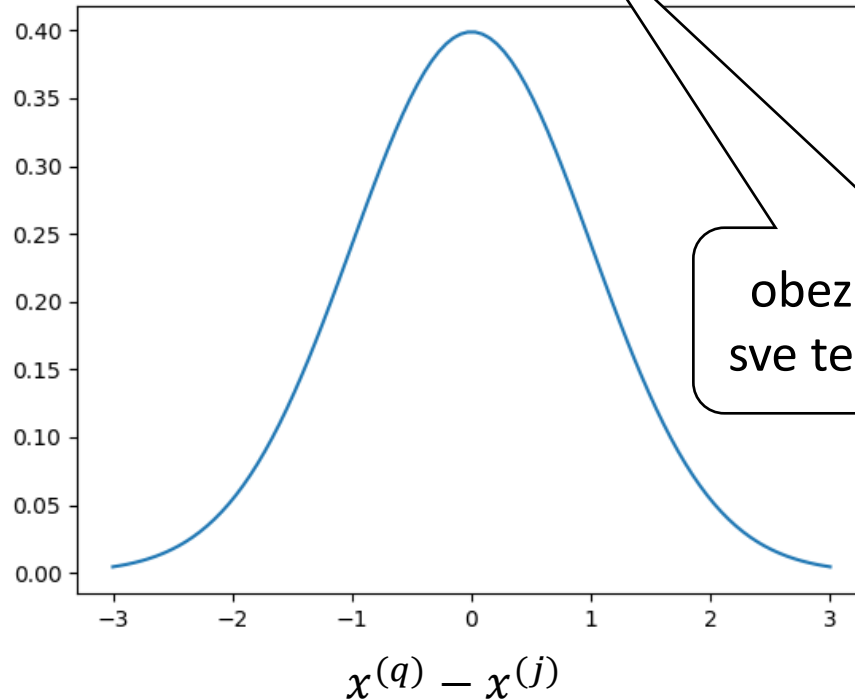
k -NN sa težinskim faktorima udaljenosti

- Kako definisati težine? Želimo:
 - Malo $w^{(qNNj)}$ za malo $distance(x^{(NNj)}, x^q)$
 - Veliko $w^{(qNNj)}$ za veliko $distance(x^{(NNj)}, x^q)$
- Definisaćemo težine pomoću jezgra (eng. *kernel*)
- Jednostavan slučaj *isotropic kernel*-a: jezgra koja su samo funkcija rastojanja bilo koje tačke i tačke x^q
- Različita jezgra definišu koliko brzo se težine smanjuju kao funkcija rastojanja tačke od x^q

Primer: Gausov kernel

kernelom određujemo koliko su opservacije “slične”

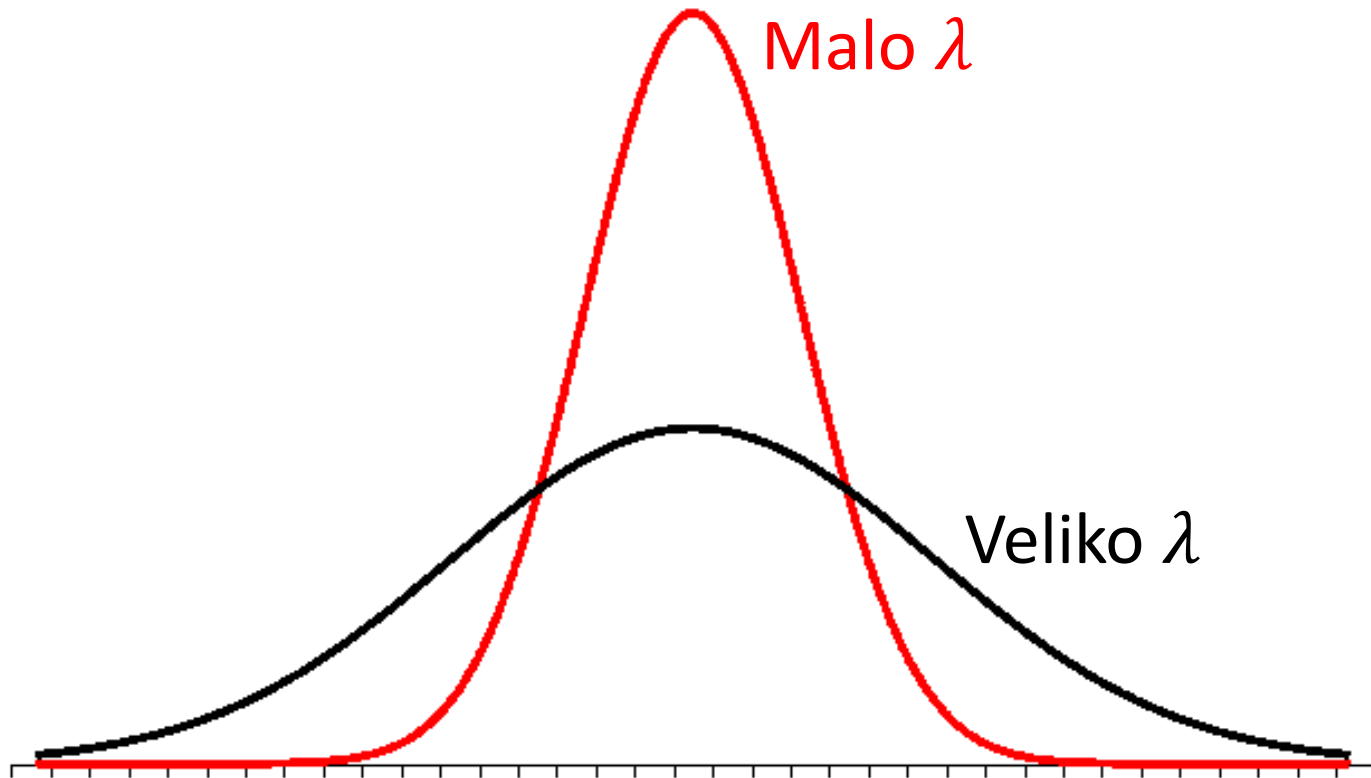
$$\text{Kernel}_\lambda(x^{(q)} - x^{(j)}) = \frac{1}{\sqrt{2\pi}} \exp \left\{ \frac{-(x^{(q)} - x^{(j)})^2}{\lambda} \right\}$$



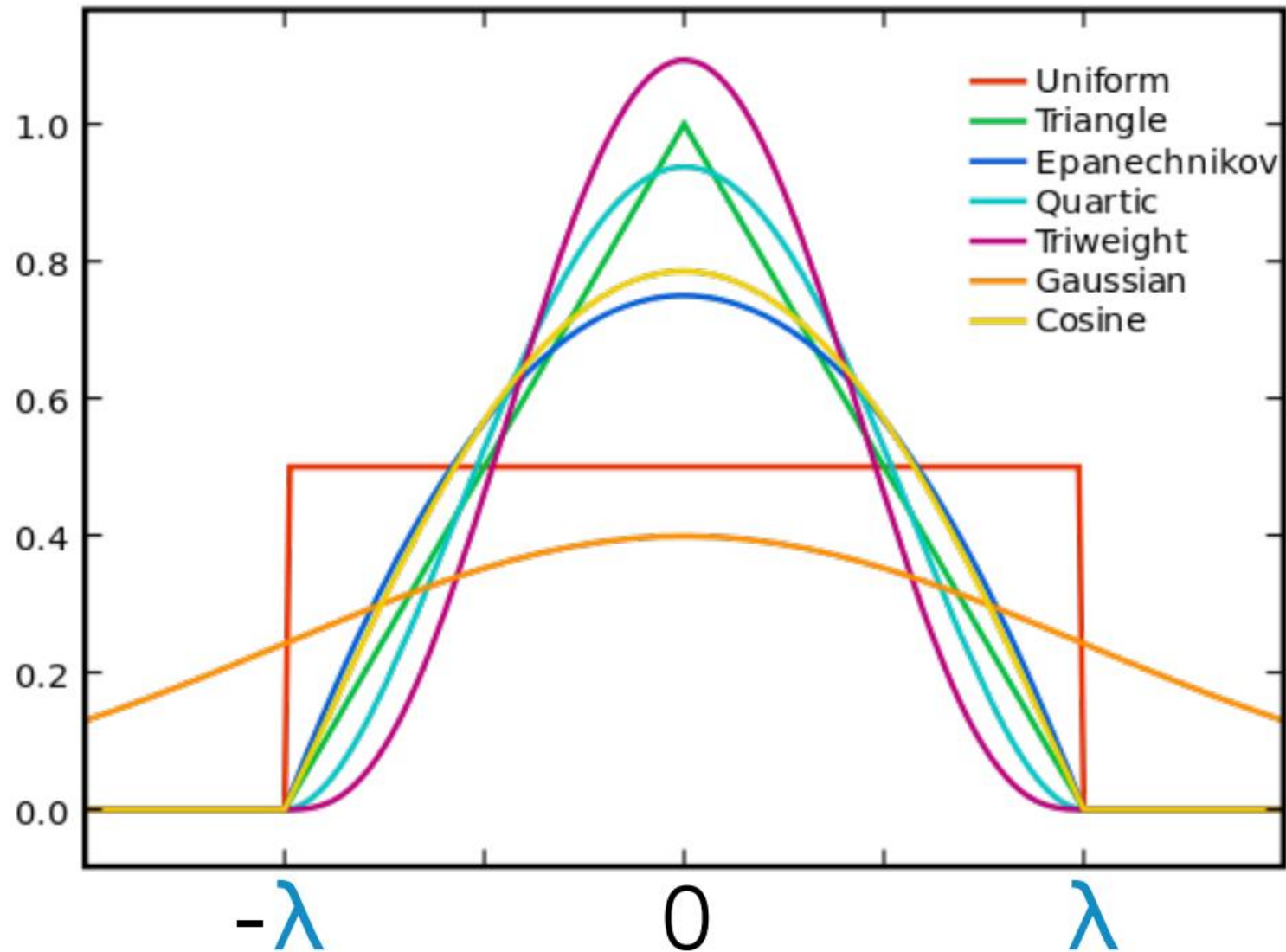
definiše koliko brzo
funkcija opada

obezbeđuje da su
sve težine pozitivne

Primer: Gausov kernel



Različita Jezgra (kerneli)



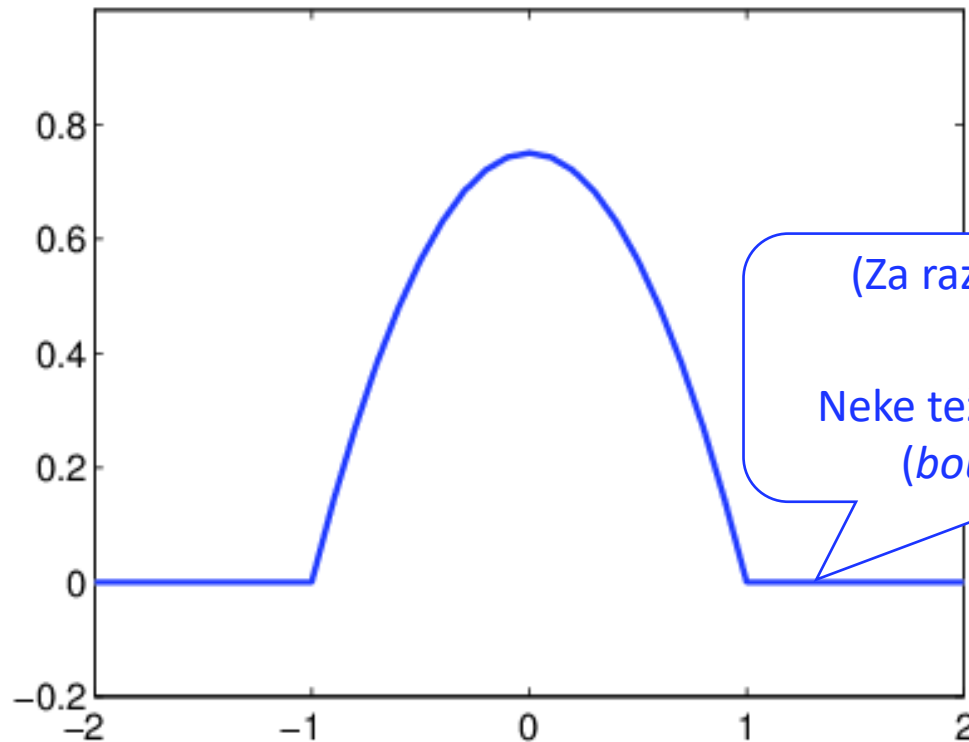
Kernel regresija

- Pristup sličan k -NN sa težinskim faktorima udaljenosti
- Umesto da pridružujemo težine samo najbližim susedima, dodelićemo težinu *svim* opservacijama u trening skupu
- Predikcija *Nadaraya-Watson kernel weighted average*:

$$\hat{y}^{(q)} = \frac{\sum_{i=1}^N K_{\lambda} \left(\frac{x^{(q)} - x^{(i)}}{\lambda} \right) y^{(i)}}{\sum_{i=1}^N K_{\lambda} \left(\frac{x^{(q)} - x^{(i)}}{\lambda} \right)}$$

Epanechnikov kernel

$$K_{\lambda}(x, y) = D\left(\frac{|x - y|}{\lambda}\right) \quad D(t) = \begin{cases} \frac{3}{4}(1 - t^2), & t < 1 \\ 0, & \text{inače} \end{cases}$$



(Za razliku od Gausovog
kernela)

Neke težine su eksplicitno 0
(*bounded support*)

Epanechnikov kernel – primer

$$x^q = 3.5, \quad y^q = ?$$

$$\lambda = 0.1$$

$$\hat{y}^{(q)} = \frac{\sum_{i=1}^N K_\lambda \left(\frac{x^{(q)} - x^{(i)}}{\lambda} \right) y^{(i)}}{\sum_{i=1}^N K_\lambda \left(\frac{x^{(q)} - x^{(i)}}{\lambda} \right)}$$

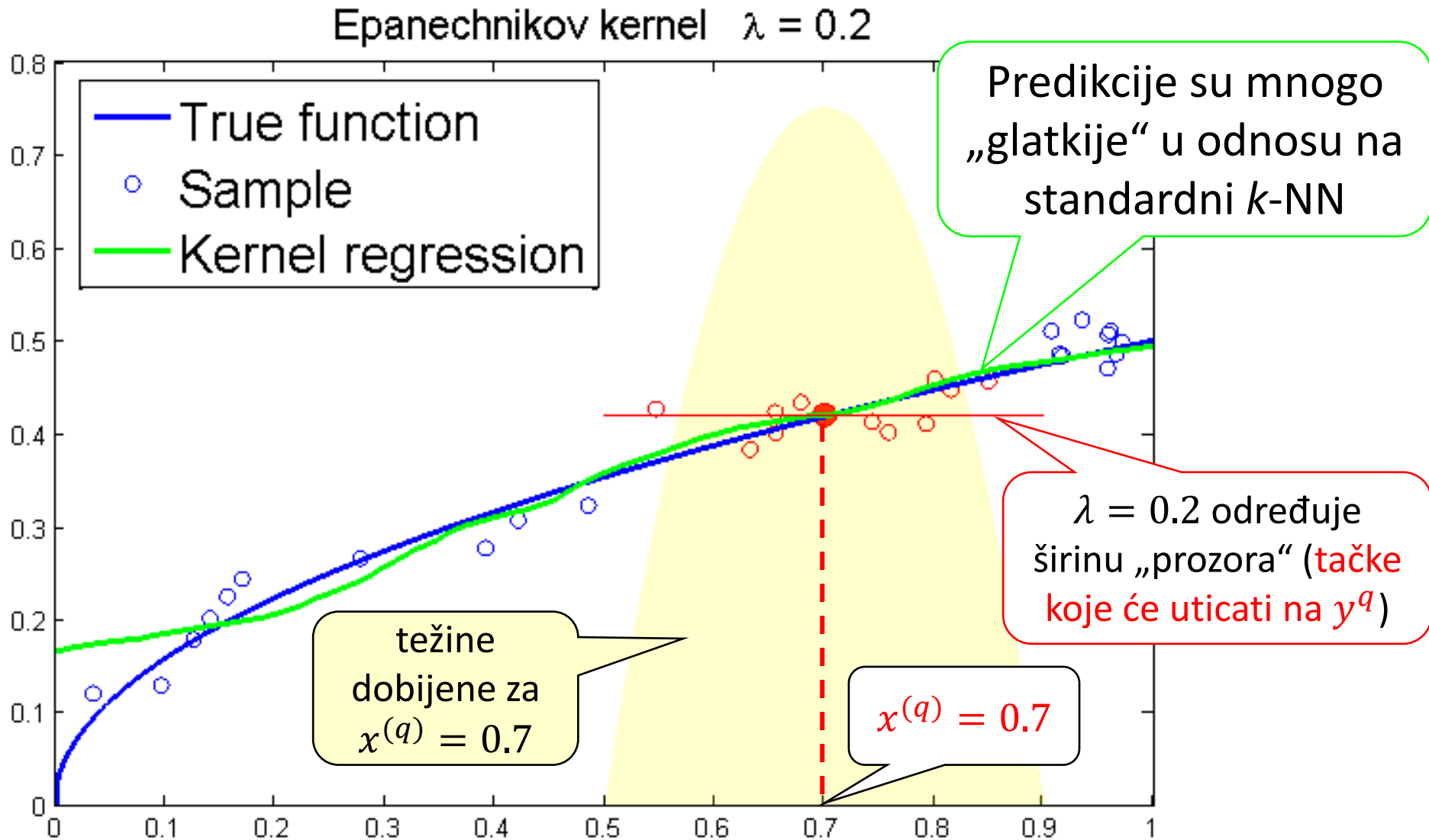
$$K_\lambda(x, y) = D \left(\frac{|x - y|}{\lambda} \right)$$

$$D(t) = \begin{cases} \frac{3}{4}(1 - t^2), & t < 1 \\ 0, & \text{inače} \end{cases}$$

$$\hat{y}^{(q)} = \frac{8.4943}{1.6099} = 5.2762$$

x^j	y^j	$\frac{ x^q - x^j }{\lambda}$	$K_\lambda = D \left(\frac{ x - y }{\lambda} \right)$	$K_\lambda \cdot y^j$
3.0975	5.8787	4.0249	0	0
3.1269	5.8232	3.7309	0	0
3.1418	5.8827	3.5819	0	0
3.1576	5.8613	3.4238	0	0
3.2784	5.5232	2.2157	0	0
3.4217	5.3962	0.7826	0.2907	1.5686
3.4853	5.2560	0.1465	0.7339	3.8574
3.5469	5.2418	0.4685	0.5853	3.0683
3.6324	5.1205	1.3236	0	0
3.7923	4.7350	2.9228	0	0
3.8003	4.4386	3.0028	0	0
3.8148	4.5322	3.1478	0	0
3.9058	4.1041	4.0579	0	0
3.9134	4.0967	4.1339	0	0
3.9158	4.2446	4.1579	0	0
3.9572	4.3408	4.5720	0	0
3.9576	4.0179	4.5760	0	0
3.9595	4.1718	4.5950	0	0
3.9649	4.0726	4.6490	0	0
3.9706	4.2435	4.7060	0	0
Σ			1.6099	8.4943

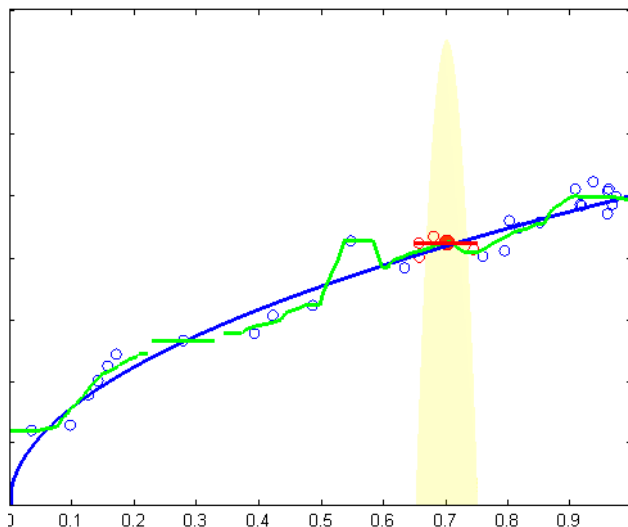
Epanechnikov kernel



Izbor λ

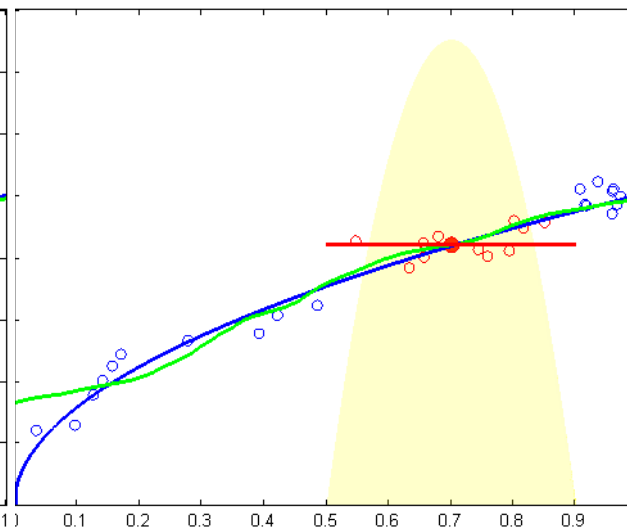
Često, izbor kernela manje utiče na rešenje od izbora λ

Epanechnikov kernel $\lambda=0.05$



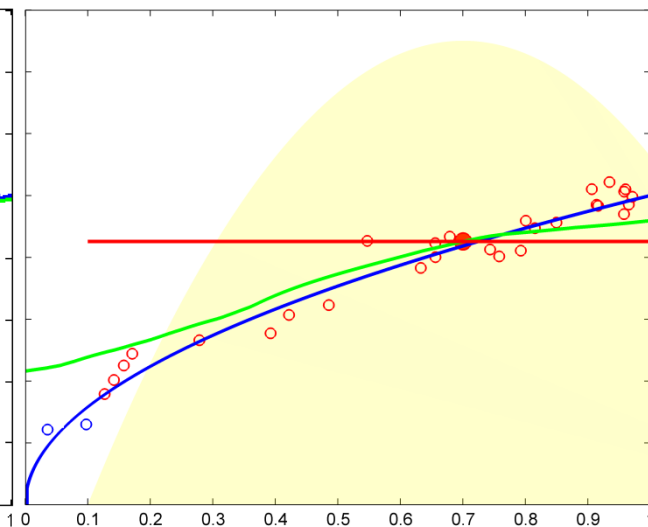
Overfitting
velika varijansa

Epanechnikov kernel $\lambda=0.20$



Dobra aproksimacija

Epanechnikov kernel $\lambda=0.60$

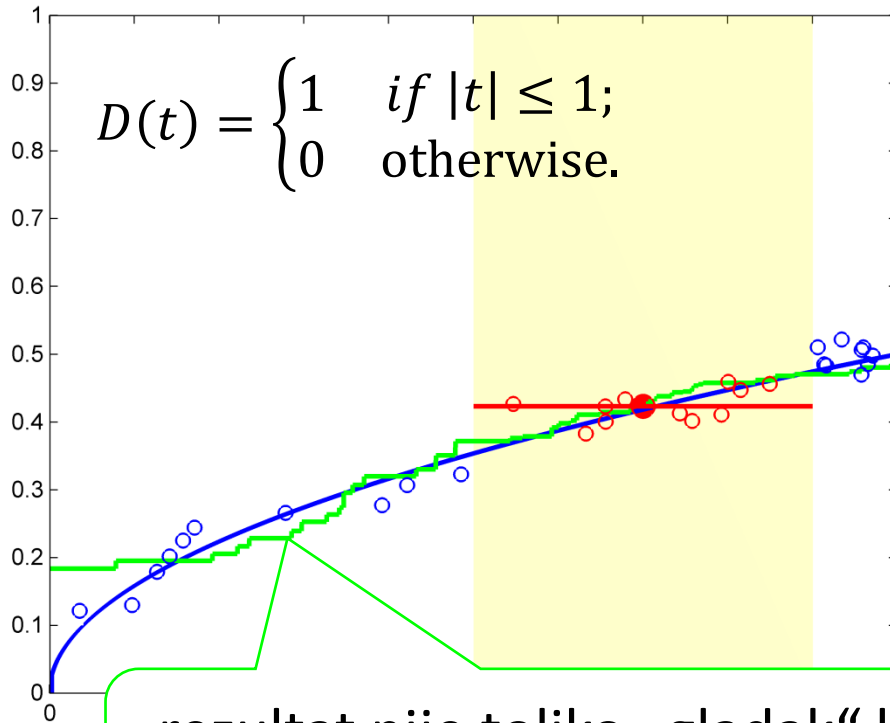


Oversmoothed fit
veliko sistematsko odstupanje

λ biramo unakrsnom validacijom ili trening/validacioni/test skup

Izbor kernela

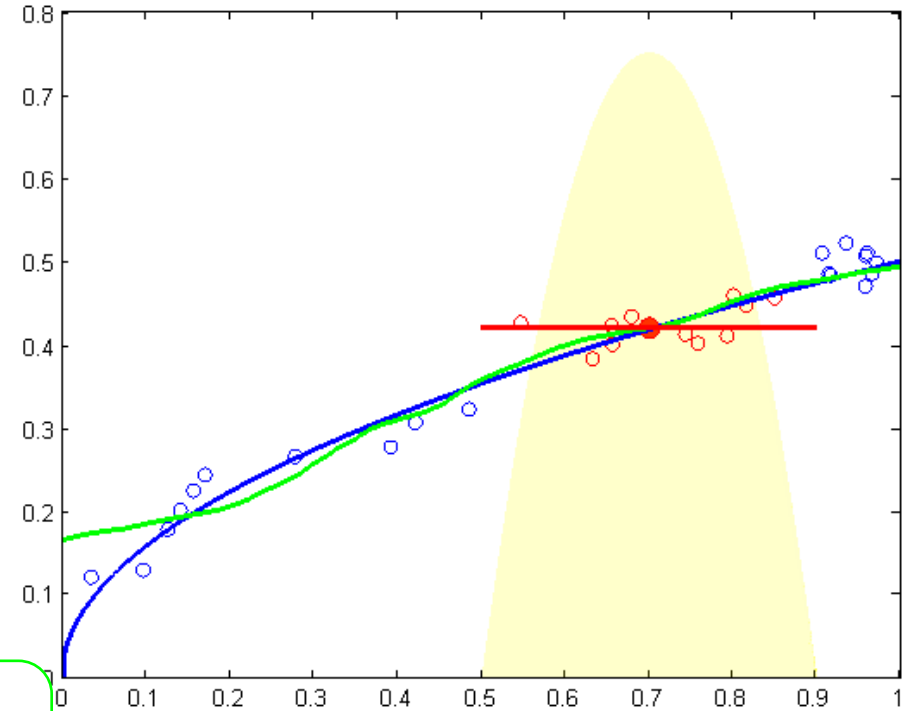
Boxcar kernel $\lambda=0.20$



rezultat nije toliko „gladak“ kao kod Epanechnikov kernela

međutim, rezultujuće krive su prilično slične

Epanechnikov kernel $\lambda=0.20$



Izbor kernela manje utiče na rešenje od izbora λ