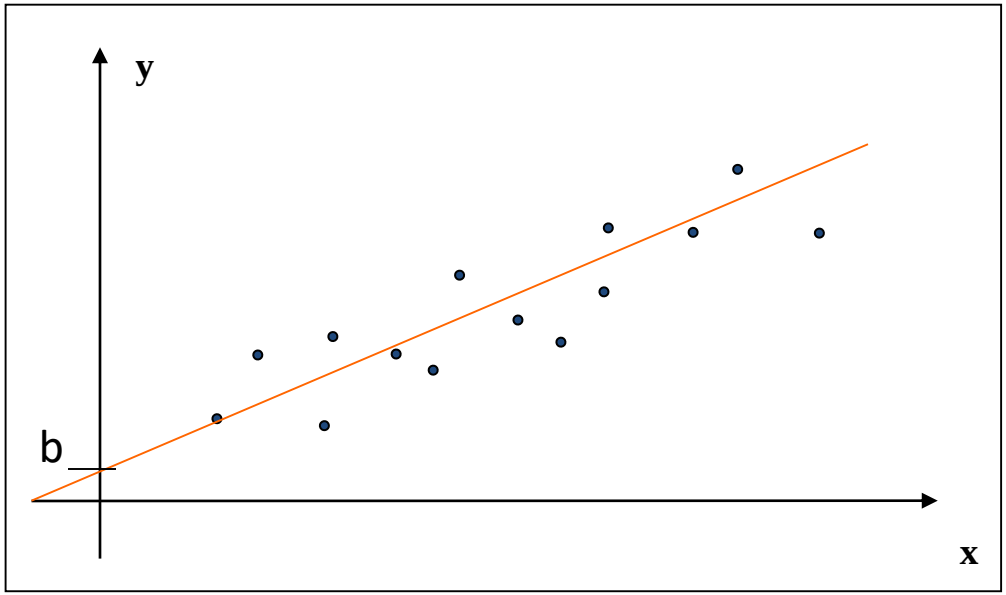
From Linear Regression to Weighted Nonlinear Regression

Linear Regression

Linear regression searches a linear mapping between input *x* and output *y,* parametrized by the slope vector *w* and *intercept b.*

$$y = f\left(x; w, b\right) = w^T x + b$$



Linear Regression

Linear regression searches a linear mapping between input *x* and output *y,* parametrized by the slope vector *w* and *intercept b.*

$$y = f\left(x; w, b\right) = w^T x + b$$

One can omit the intercept by centering the data:

$$y' = y - \bar{y} \text{ and } x' = x - \bar{x}, \quad \bar{x}, \bar{y}: \text{ mean on } x \text{ and } y$$

$$y' = w^T x' + b'$$

$$\text{with } b' = b + w^T \bar{x} - \bar{y}$$

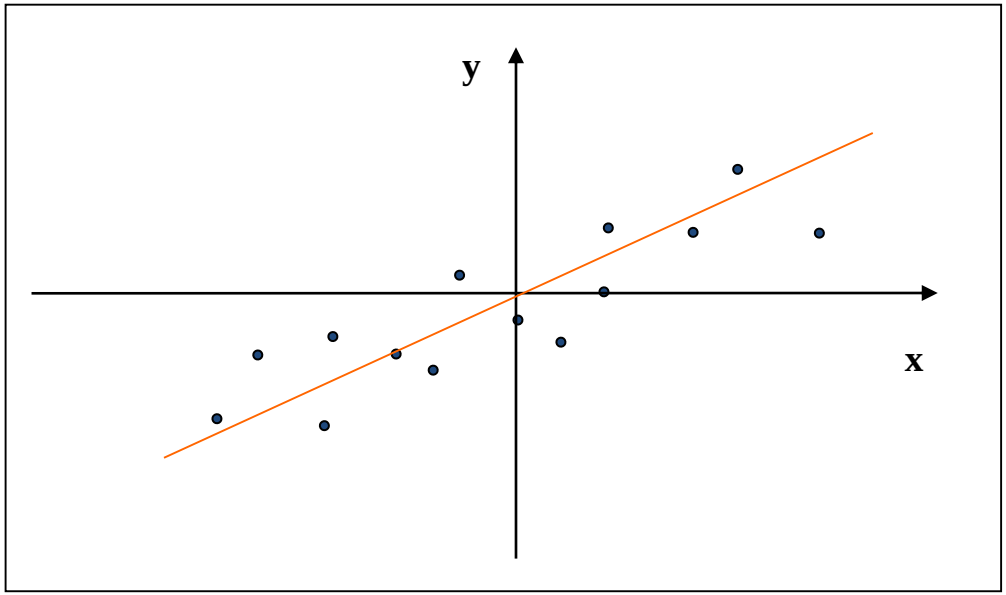
$$\text{Least-square estimate of } \left(b'\right)^* = \bar{y}' - w^T \bar{x}' = 0$$

$$\Rightarrow y' = w^T x'.$$

Linear Regression

Linear regression searches a linear mapping between input *x* and output *y,* parametrized by the slope vector *w.*

$$y = f\left(x; w\right) = w^T x$$



Linear Regression

Pair of M training points $X = [x^1 \; x^2 \; \ldots \; x^M]$ and $y = [y^1 \; y^2 \; \ldots \; y^M]^T$

$x^i \in \mathbb{R}^N, y^i \in \mathbb{R}.$

Find the optimal parameter w through least-square regression:

$$w^* = \min_w \left(\sum_{i=1}^{M} \frac{1}{2} \left(w^T x^i - y^i \right)^2 \right)$$

Find the analytical solution through partial differentiation:

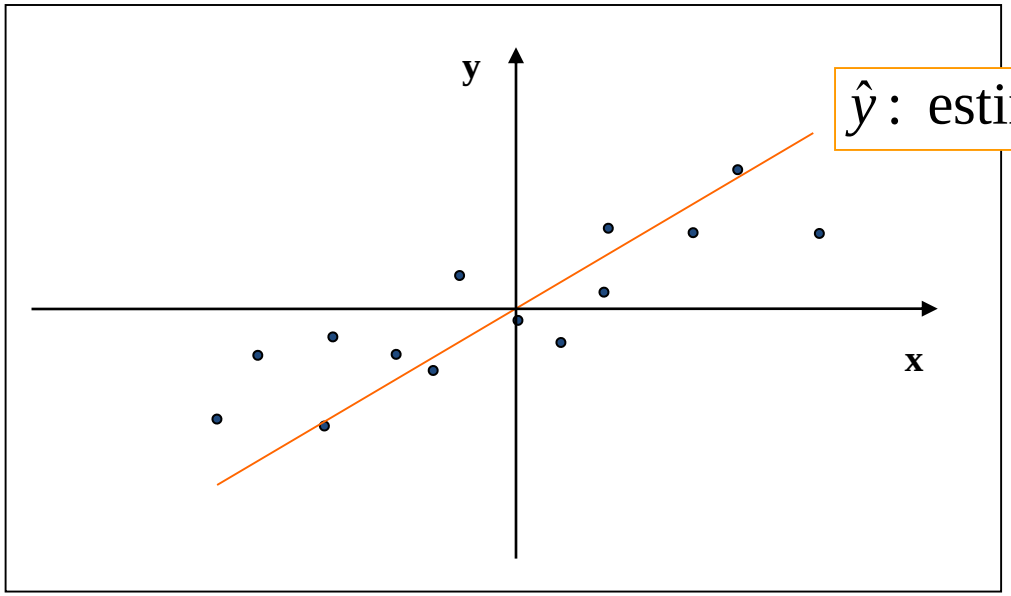
$$w^* = \left(XX^T \right)^{-1} Xy$$

Weighted Linear Regression

Regression through weighted Least Square

$$w^* = \min_w \left(\sum_{i=1}^{M} \frac{1}{2} \beta_i \left(w^T x^i - y^i \right)^2 \right), \qquad \beta_i \in \mathbb{R} \qquad \beta_1 = \beta_2 \ldots = \beta_M$$

→ Standard linear regression



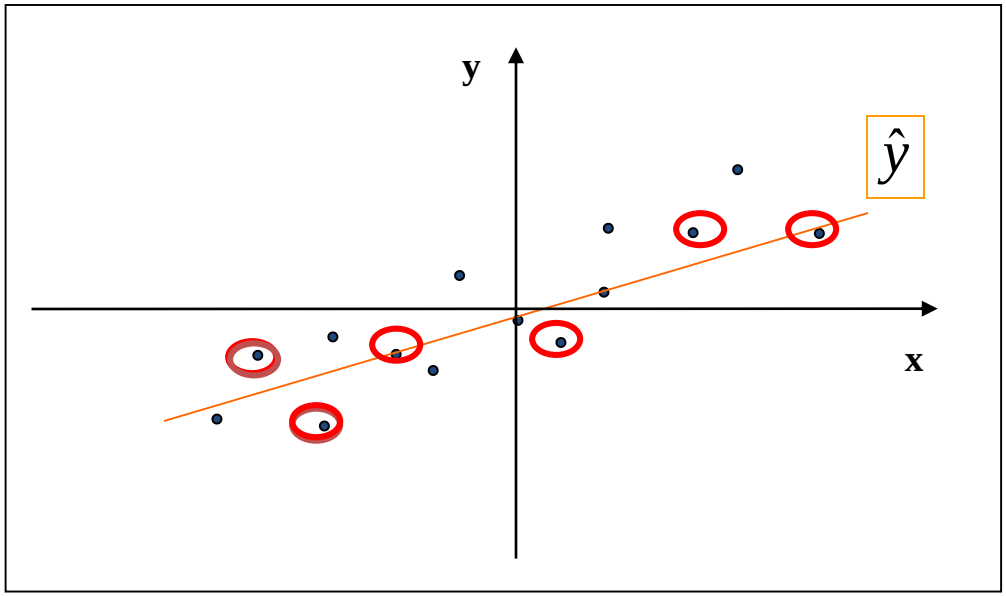
$\hat{y}$: estimator

All points have equal weight.

Weighted Linear Regression

Regression through weighted Least Square

$$w^* = \min_w \left(\sum_{i=1}^{M} \frac{1}{2} \beta_i \left(w^T x^i - y^i \right)^2 \right), \qquad \beta_i \in \mathbb{R}$$

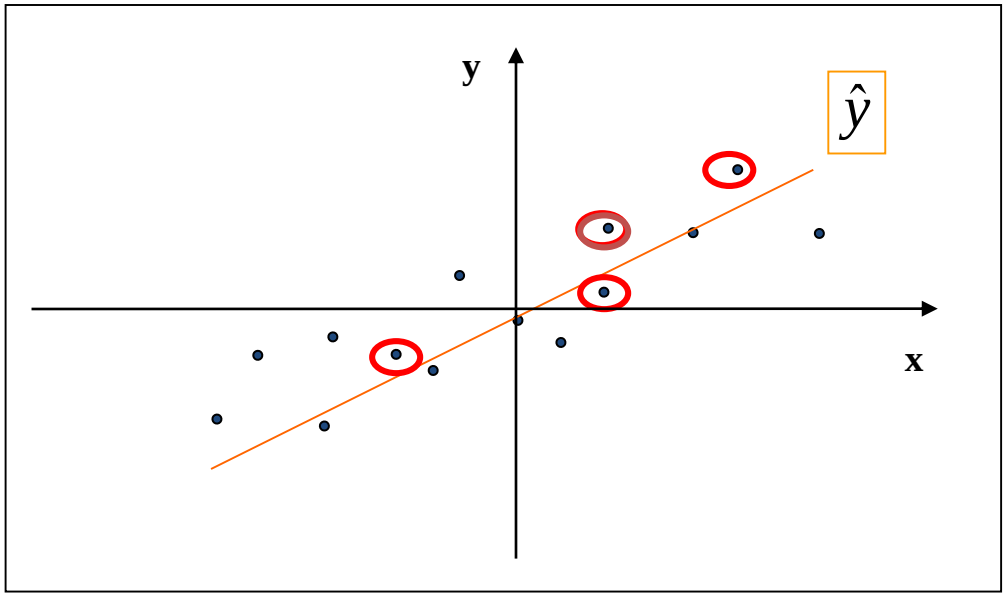


Points in red have large weights.

Weighted Linear Regression

Regression through weighted Least Square

$$w^* = \min_w \left(\sum_{i=1}^{M} \frac{1}{2} \beta_i \left(w^T x^i - y^i \right)^2 \right), \qquad \beta_i \in \mathbb{R}$$



Points in red have large weights.

Computing the optimal regressor

We start from the loss function: $\sum_{i=1}^{M} \frac{1}{2} \beta_i \left(w^T x^i - y^i \right)^2$

We set B a diagonal matrix with entries β_i, $B = \begin{bmatrix} \beta_1 & & \\ & \beta_2 & \\ \ldots\ldots & & \\ \ldots\ldots\ldots\ldots\ldots & & \beta_M \end{bmatrix}$

Change of variable: $Z = XB^{1/2}$ and $v = B^{1/2} y \quad \Rightarrow$ Loss: $w^T Z - v$

Minimizing for loss, one gets a closed-form best estimator for w:

$\hat{y} = w^T x = \left(ZZ^T \right)^{-1} Zv \, x$

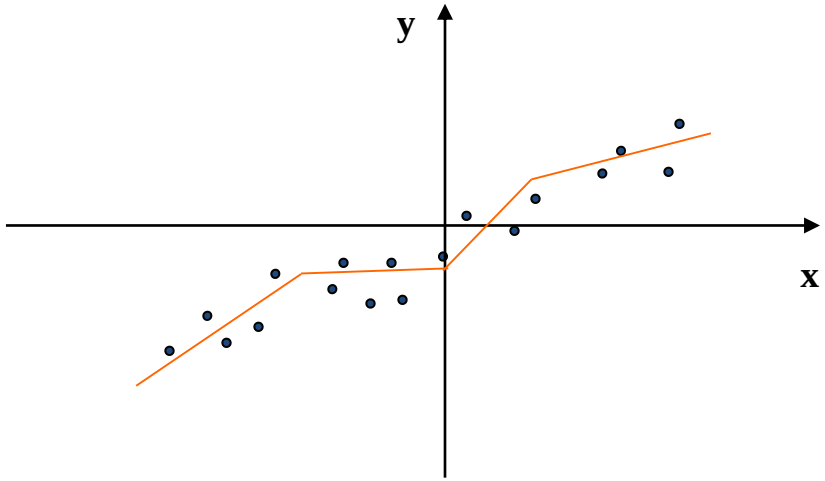
$$w^* = \left(ZZ^T \right)^{-1} Zv$$

Limitations of Linear Regression

$$w^* = \min_w \left(\sum_{i=1}^{M} \frac{1}{2} \beta_i \left(w^T x^i - y^i \right)^2 \right), \qquad \beta_i \in \mathbb{R} \text{ constant}$$

Assumes that a single linear dependency applies everywhere.

Not true for data sets with local dependencies.



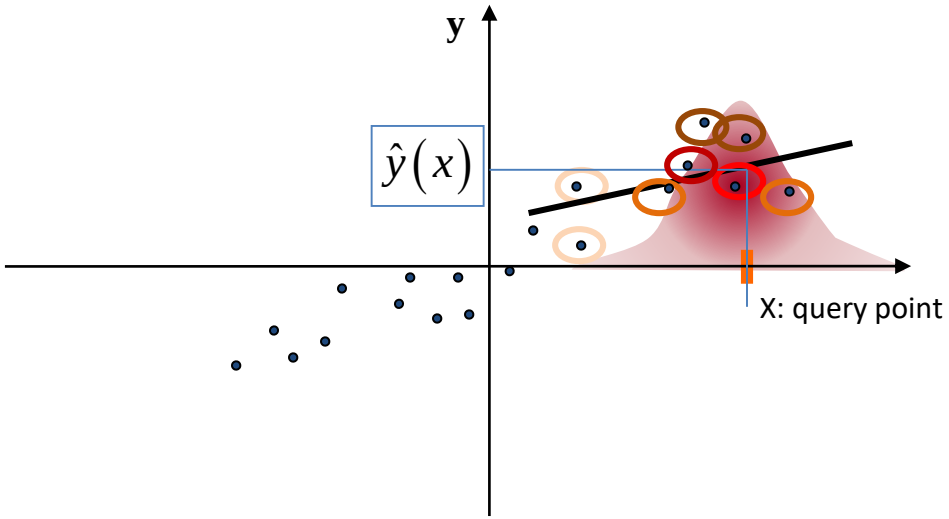
Need a regression method to estimate local linear dependencies

Locally Weighted Regression

Estimate is determined through local influence of each group of datapoints

$$\hat{y}(x) = \sum_{i=1}^{M} \beta_i(x)\, y^i \;/\; \sum_{j=1}^{M} \beta_j(x) \qquad \beta_i(x) \in \mathbb{R}: \text{ weights function of x}$$

$$\beta_i(x) = K\left(d\left(x^i, x\right)\right), \quad \text{with } K\left(d\left(x^i, x\right)\right) = e^{-d\left(x^i, x\right)},\ d\left(x^i, x\right): \text{norm-2}$$

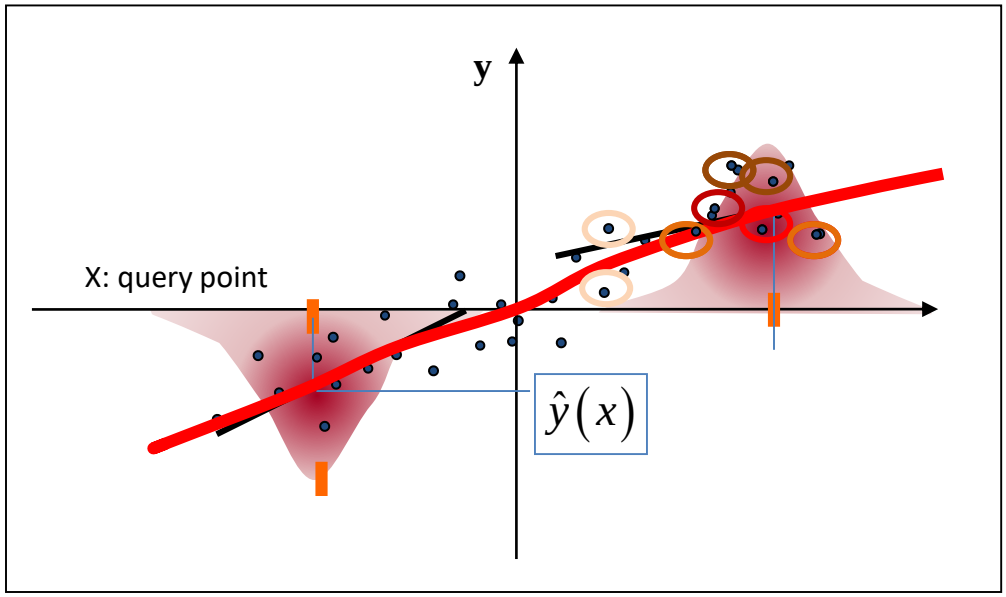


Locally Weighted Regression

Estimate is determined through local influence of each group of datapoints

$$\hat{y}(x) = \sum_{i=1}^{M} \beta_i(x)\, y^i \Big/ \sum_{j=1}^{M} \beta_j(x) \qquad \beta_i(x) \in \mathbb{R}: \text{ weights function of x}$$

$$\beta_i(x) = K\left(d\left(x^i, x\right)\right), \quad \text{with } K\left(d\left(x^i, x\right)\right) = e^{-d\left(x^i, x\right)}, \; d\left(x^i, x\right): \text{norm-2}$$



Generates a smooth function y(x)

Locally Weighted Regression

Estimate is determined through local influence of each group of datapoints

$$\hat{y}(x) = \sum_{i=1}^{M} \beta_i(x)\, y^i \;/\; \sum_{j=1}^{M} \beta_j(x)$$

$\beta_i(x) \in \mathbb{R}$: weights function of x

$$\beta_i(x) = K\left(d\left(x^i, x\right)\right)$$

Model-free regression!
No longer explicit model of the form $y = w^T x$

Regression computed at each query point.
Depends on training points.

Locally Weighted Regression

Estimate is determined through local influence of each group of datapoints

$$\hat{y}(x) = \sum_{i=1}^{M} \beta_i(x)\, y^i \Big/ \sum_{j=1}^{M} \beta_j(x) \qquad \beta_i(x) \in \mathbb{R}: \text{ weights function of x}$$

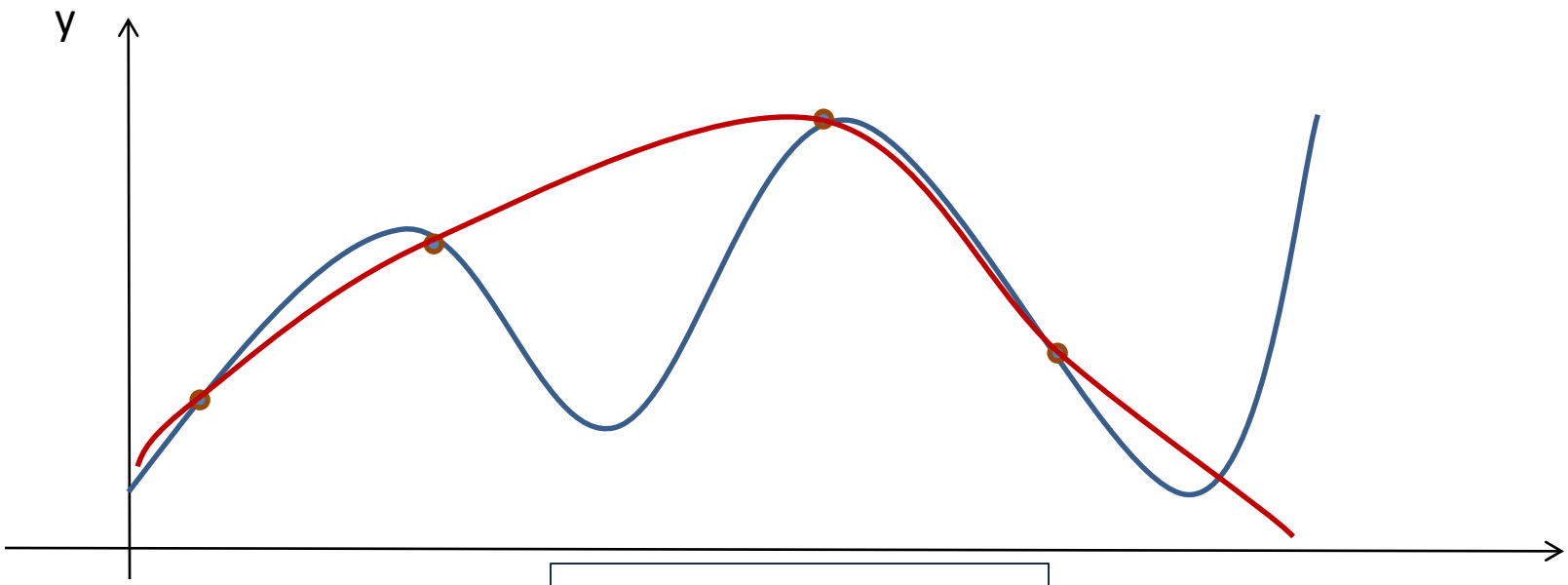
Which training points?

$$\beta_i(x) = K\left(d\left(x^i, x\right)\right)$$

Which kernel?

Data-driven Regression

Good prediction depends on the choice of datapoints.

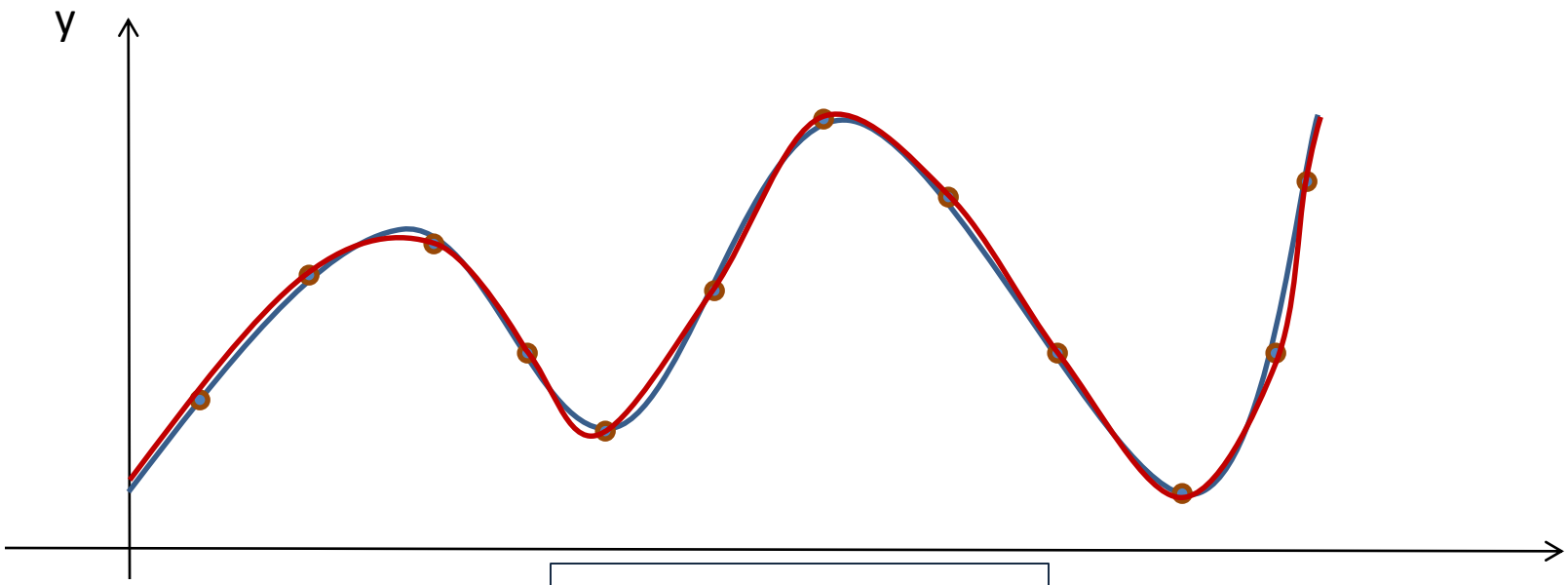


Data-driven Regression

Good prediction depends on the choice of datapoints.

The more datapoints, the better the fit.

Computational costs increase dramatically with number of datapoints

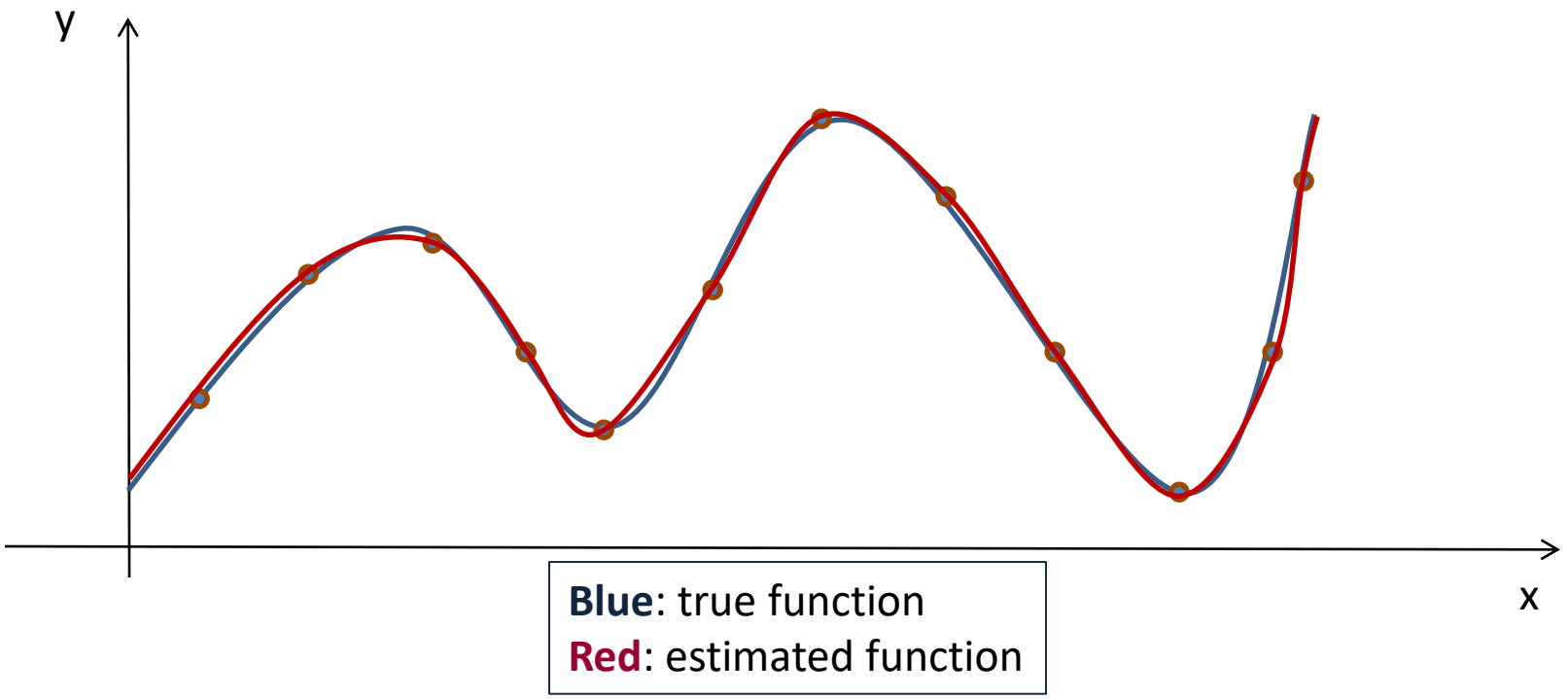


Data-driven Regression

Several methods in ML for performing non-linear regression.

Differ in the objective function, in the amount of parameters.

K-nearest neighbors (KNN) uses all datapoints (model-free)

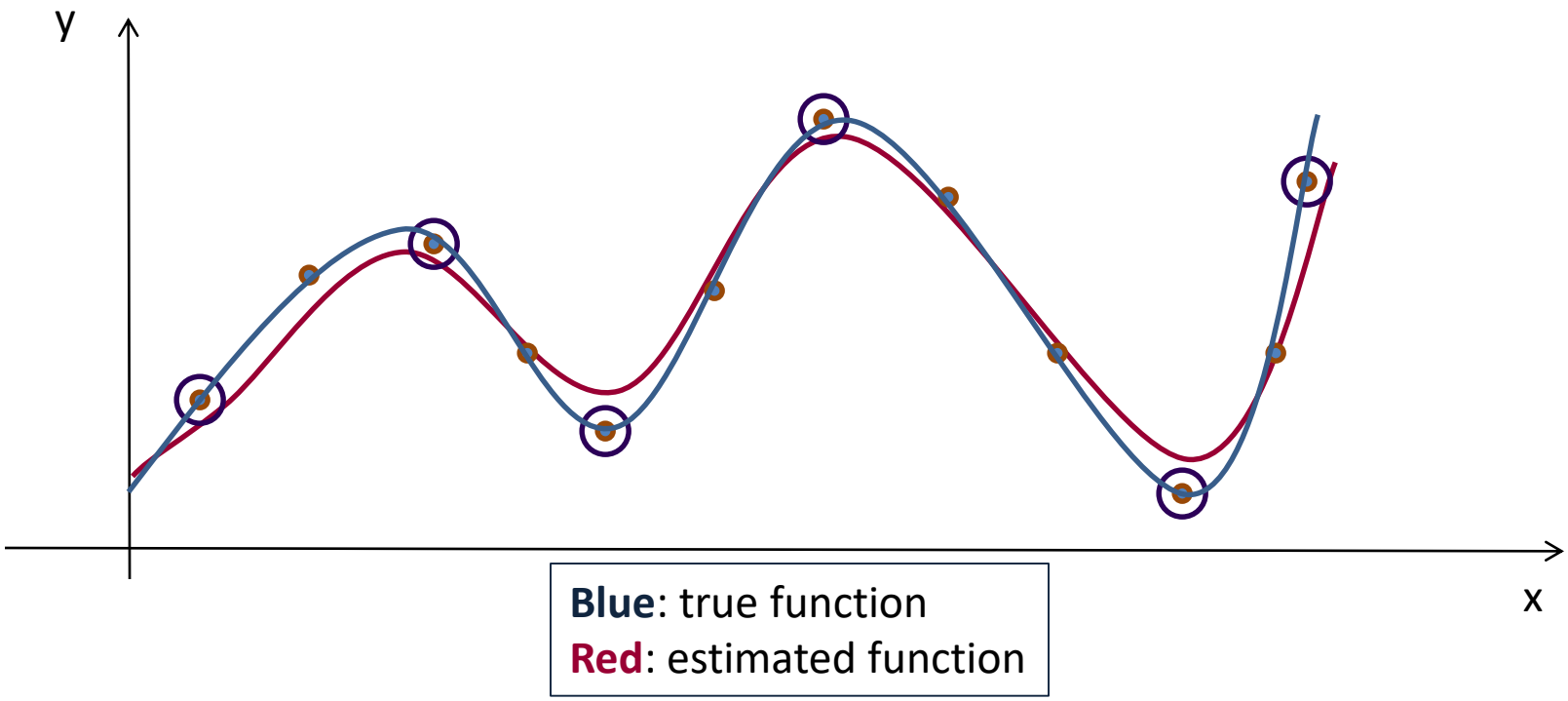


Data-driven Regression

Several methods in ML for performing non-linear regression.

Differ in the objective function, in the amount of parameters.

K-nearest neighbors (KNN) uses all datapoints (model-free)
Support Vector Regression (SVR) picks a subset of datapoints (support vectors)

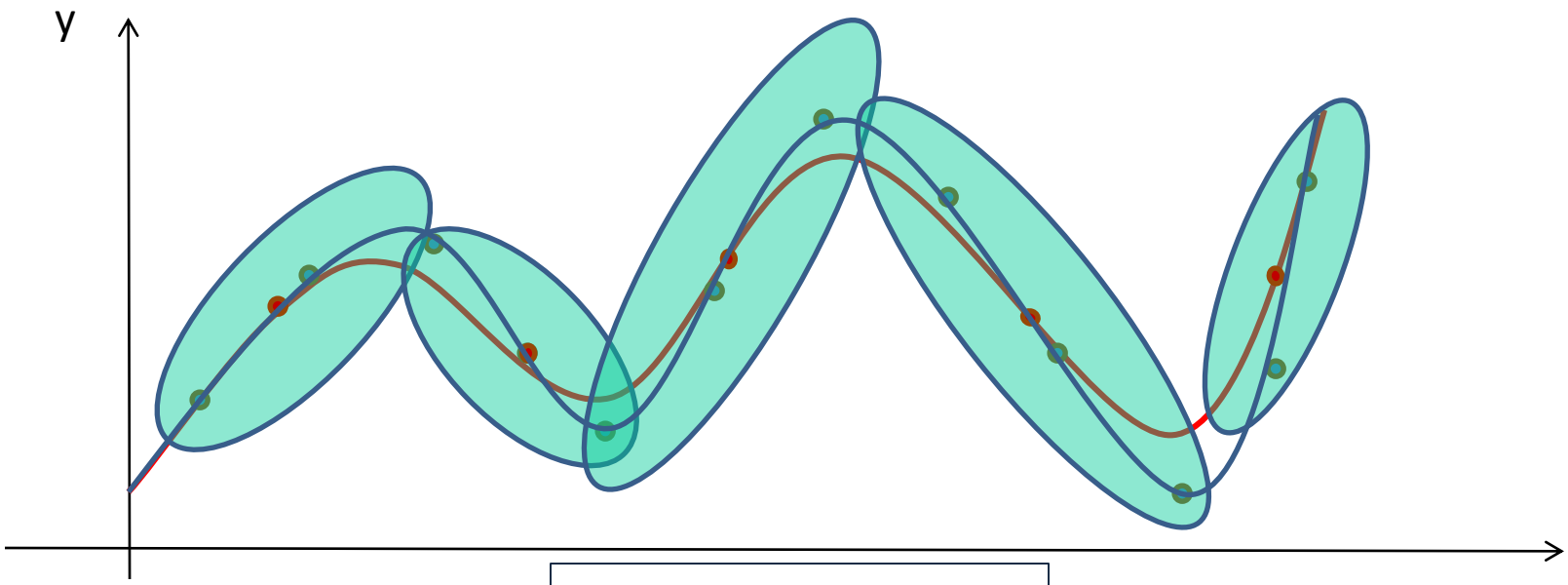


Data-driven Regression

Several methods in ML for performing non-linear regression.

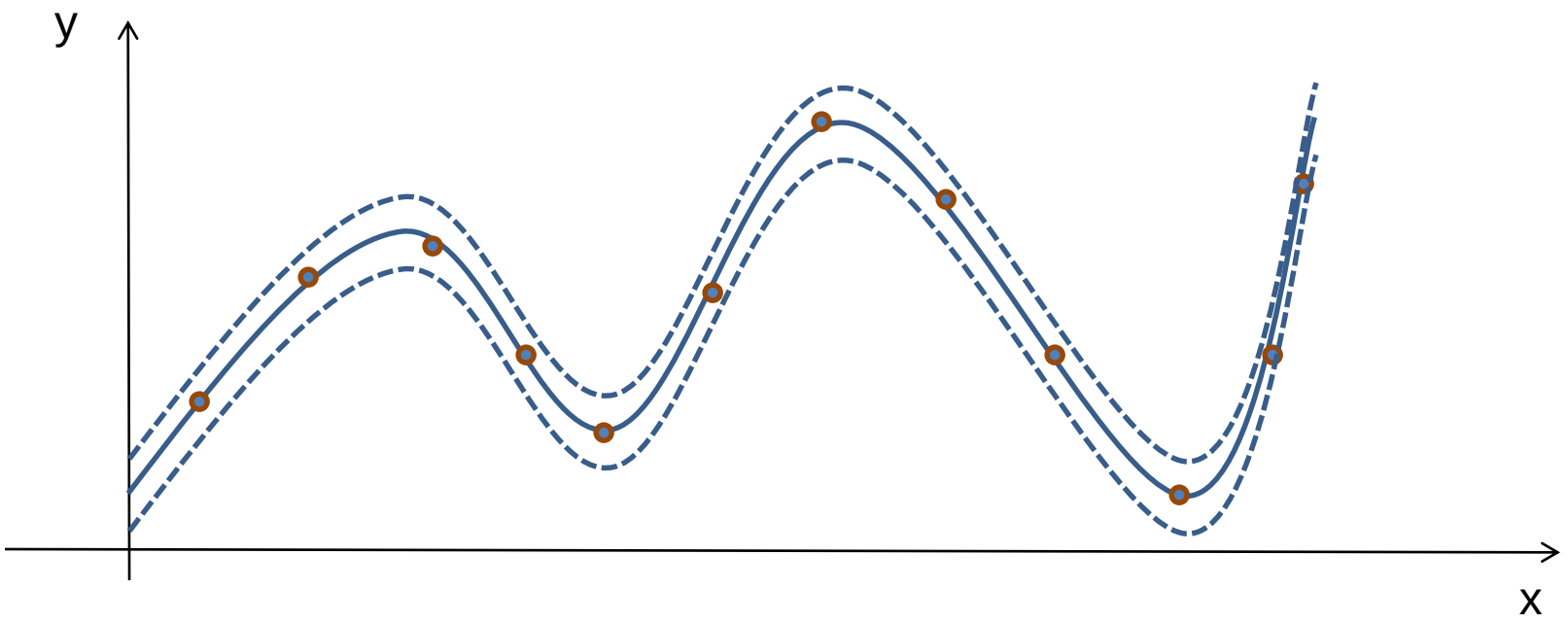
Differ in the objective function, in the amount of parameters.

K-nearest neighbors (KNN) uses all datapoints (model-free)
Support Vector Regression (SVR) picks a subset of datapoints (support vectors)
Gaussian Mixture Regression (GMR) generates a new set of datapoints (centers of Gaussian functions)



Data-driven Regression

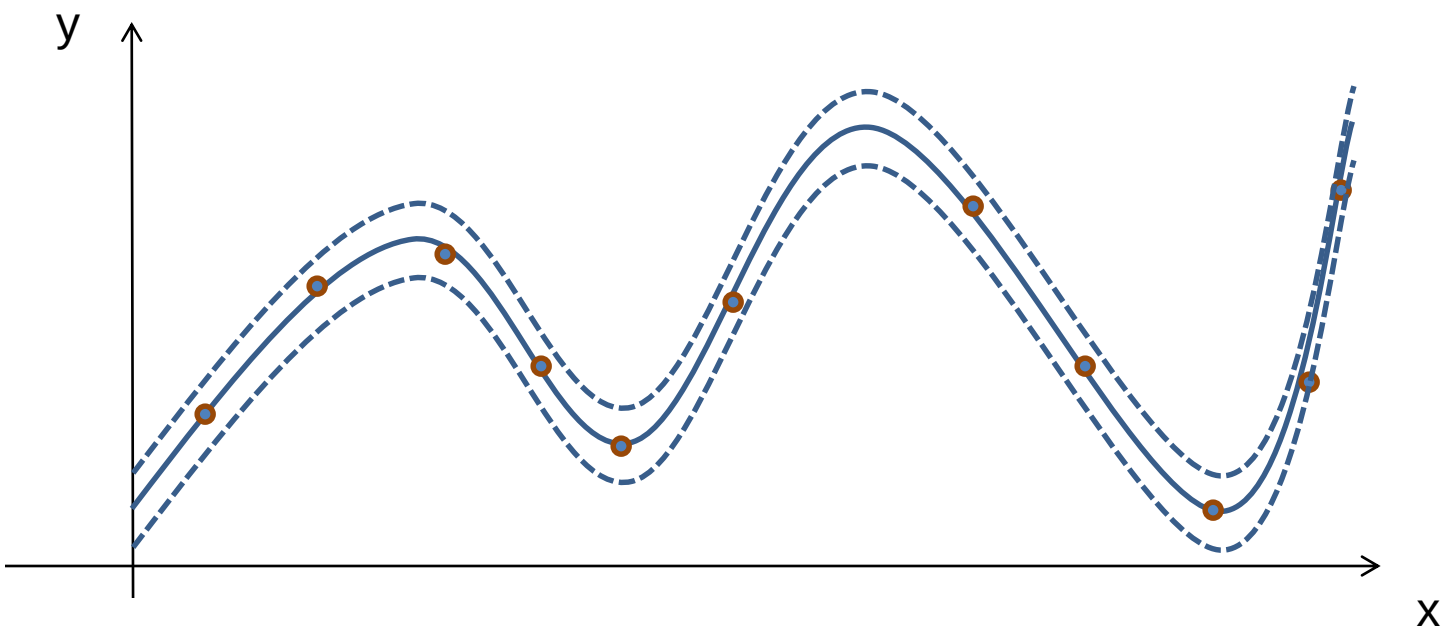
Estimate of the noise is important to measure goodness of fit.



Data-driven Regression

Estimate of the noise is important to measure goodness of fit.

Support Vector Regression (SVR) assumes an estimate of the noise model (ε-tube) and then compute *f* directly within a noise-tolerance.



Data-driven Regression

Estimate of the noise is important to measure goodness of fit.

Gaussian Mixture Regression (GMR) builds a local estimate of the noise model through the variance of the system.