CS 171: Intro to ML and DM

Christian Shelton

UC Riverside

Slide Set 4: Linear Regression, II

Slides from CS 171

- From UC Riverside
  - CS 171: Introduction to Machine Learning and Data Mining
  - Professor Christian Shelton
- DO NOT REDISTRIBUTE
  - These slides contain copyrighted material (used with permission) from
    - Elements of Statistical Learning (Hastie, et al.)
    - Pattern Recognition and Machine Learning (Bishop)
    - An Introduction to Machine Learning (Kubat)
    - Machine Learning: A Probabilistic Perspective (Murphy)
  - For use only by enrolled students in the course

Non-linear Regression

Desired: $f(x) = c + bx + ax^2$

Non-linear Regression

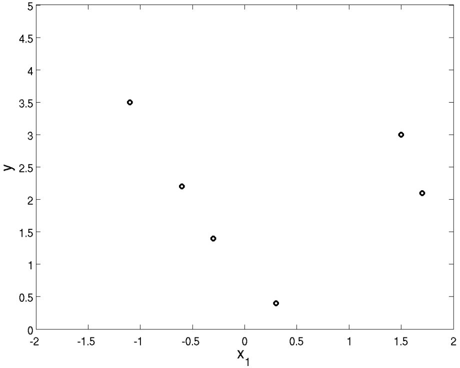
Desired: $f(x) = c + bx + ax^2$
Can be written as $f(x) = w_0 \times 1 + w_1 \times x + w_2 \times x^2$

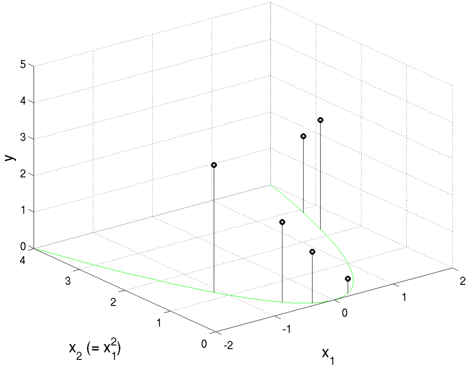
Non-linear Regression

Desired: $f(x) = c + bx + ax^2$
Can be written as $f(x) = w_0 \times 1 + w_1 \times x + w_2 \times x^2$

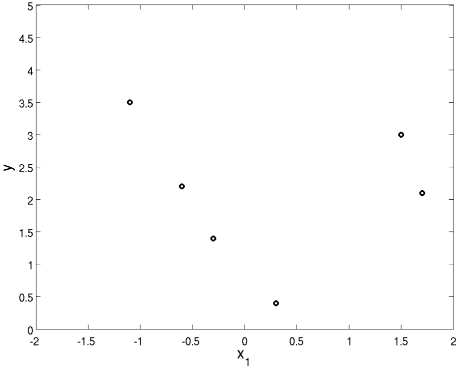
So instead of just adding the "0th" attribute (always 1)
also add other attributes that can be calculated from the given attributes.

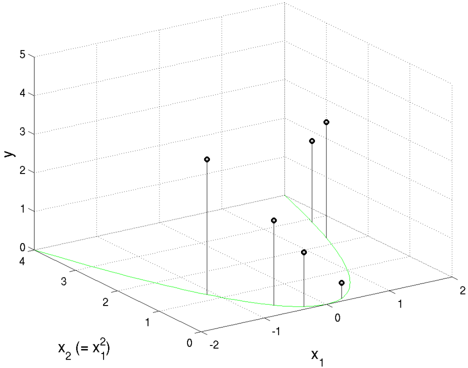
Non-linear regression



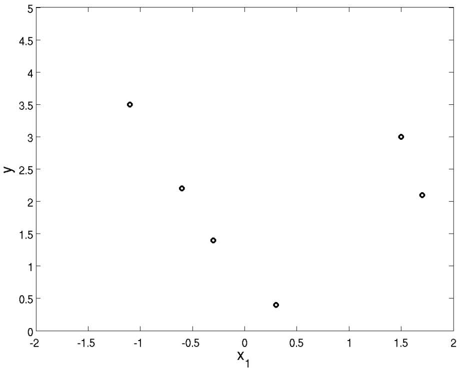


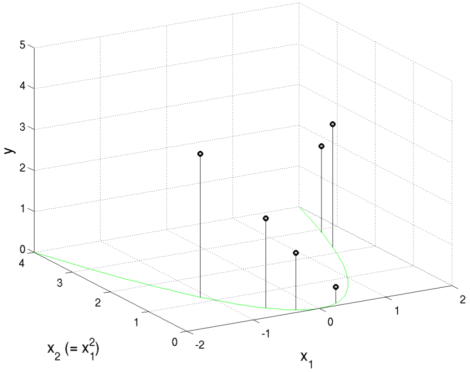
Non-linear regression



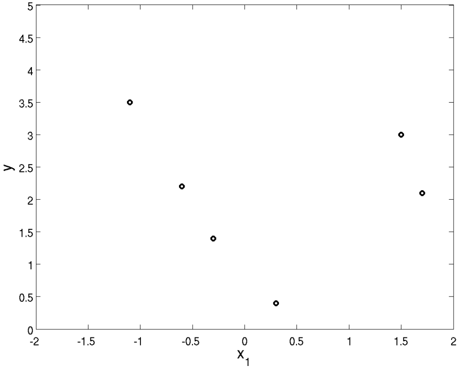


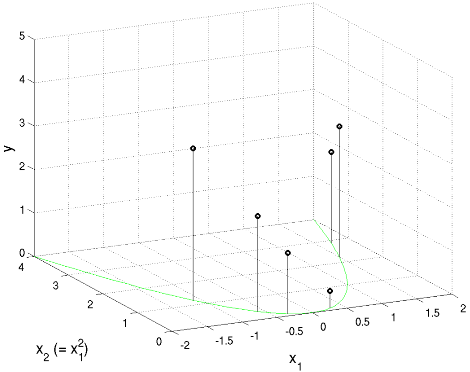
Non-linear regression



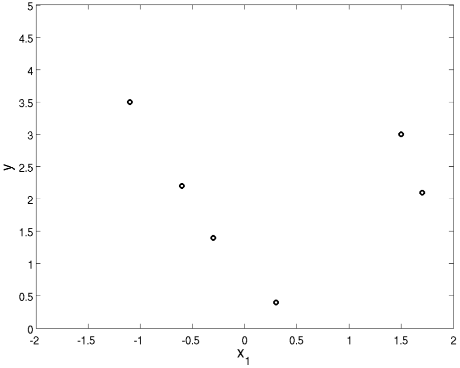


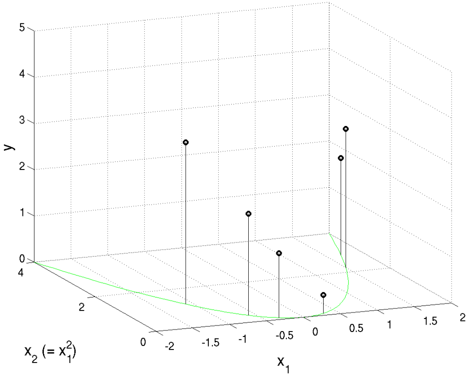
Non-linear regression



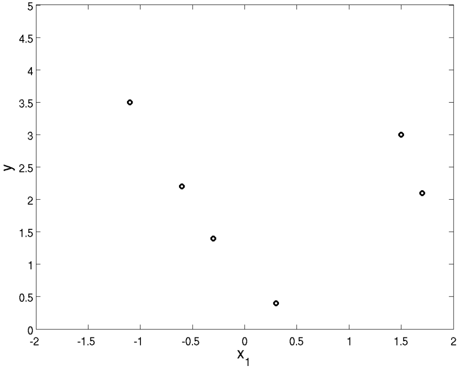


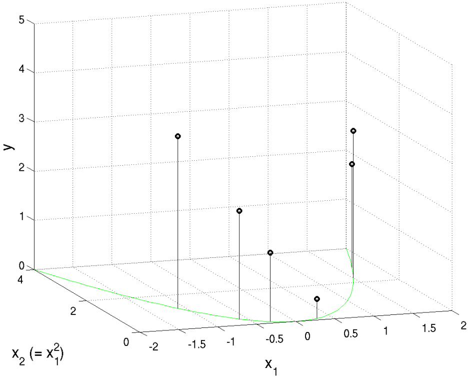
Non-linear regression



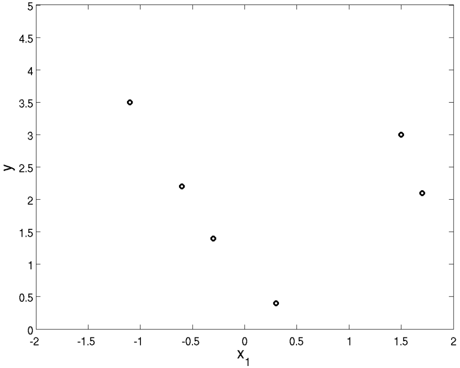


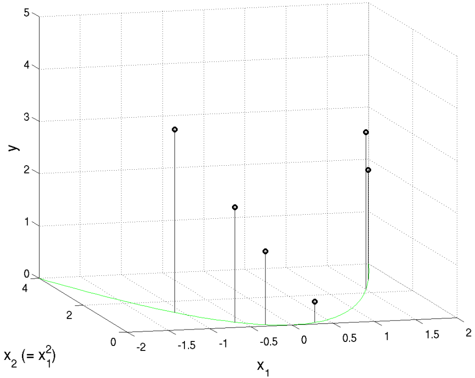
Non-linear regression



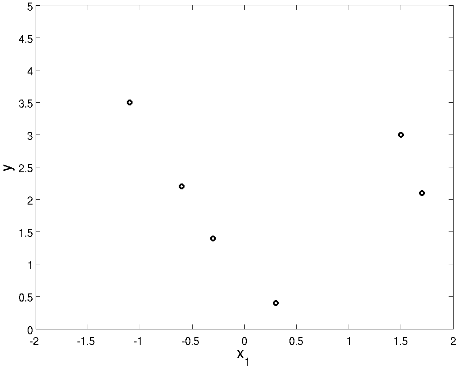


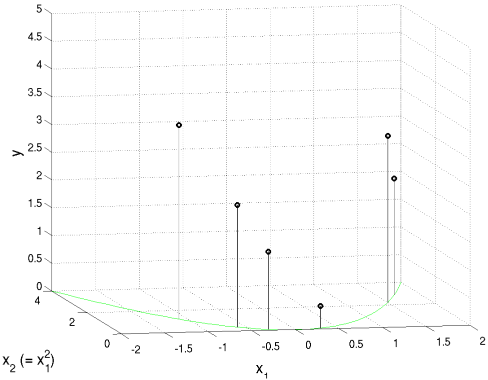
Non-linear regression



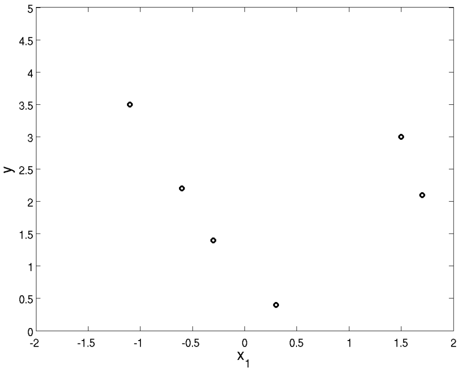


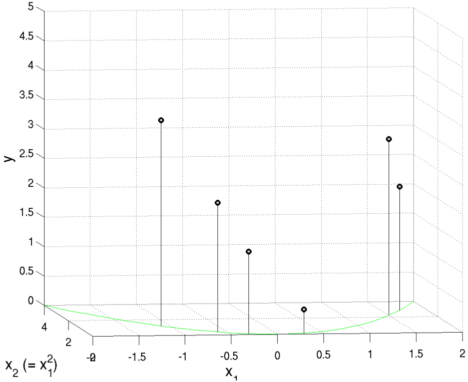
Non-linear regression



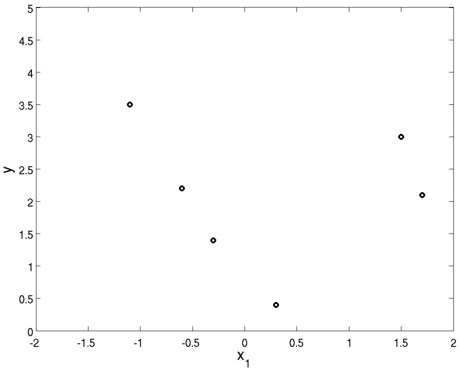


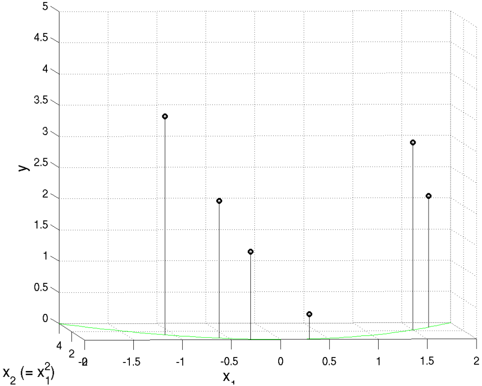
Non-linear regression



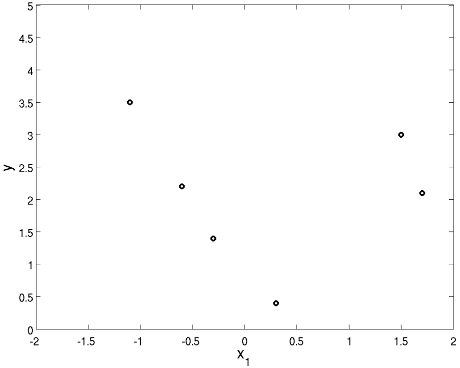


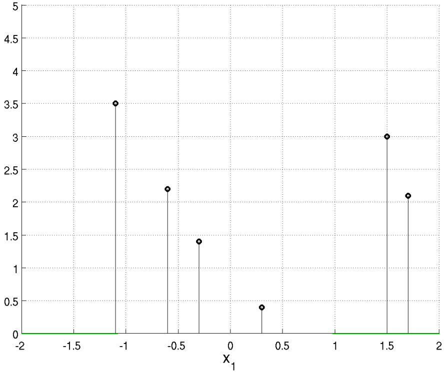
Non-linear regression



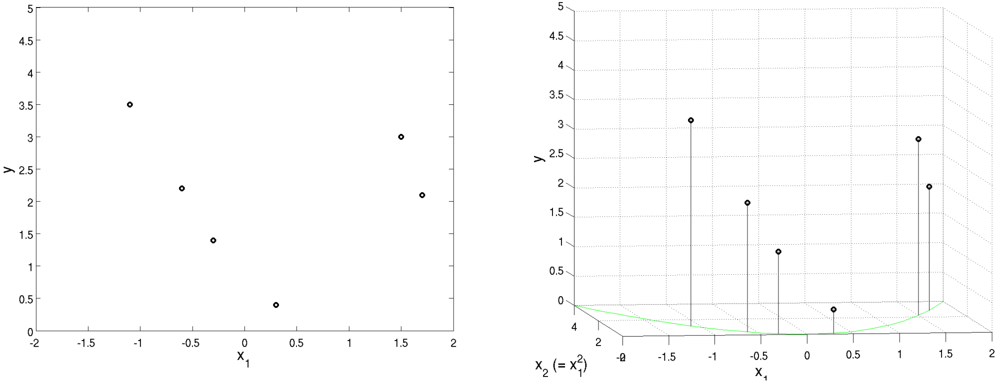


Non-linear regression

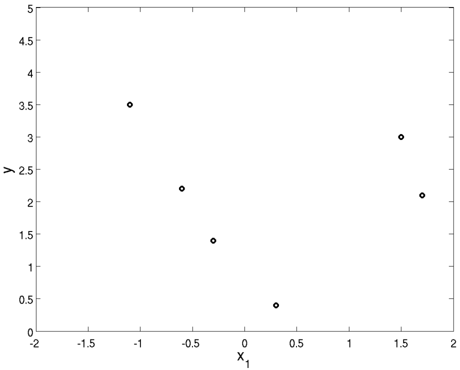


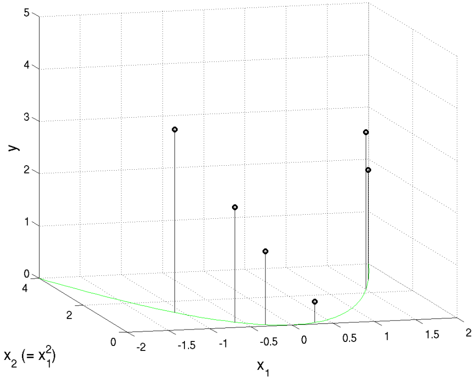


Non-linear regression

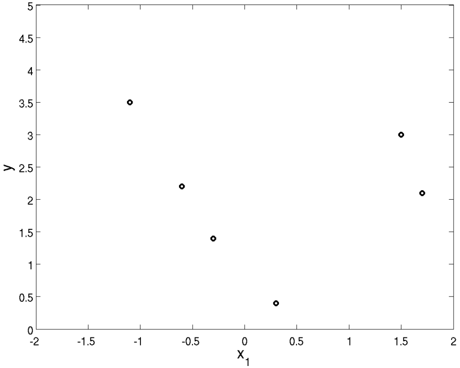


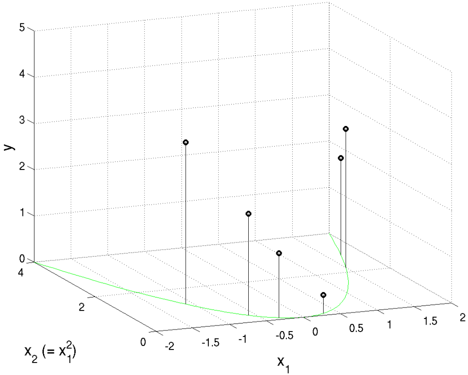
Non-linear regression



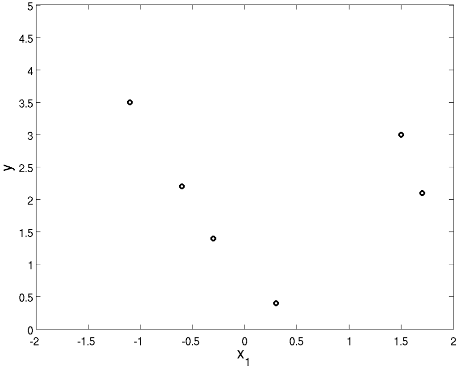


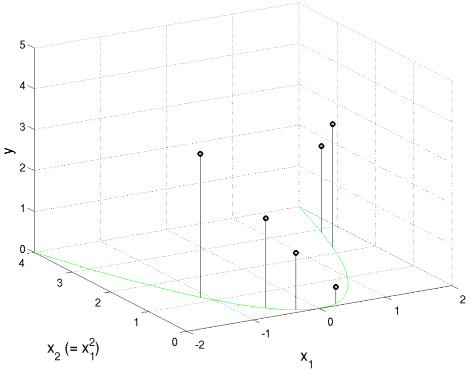
Non-linear regression



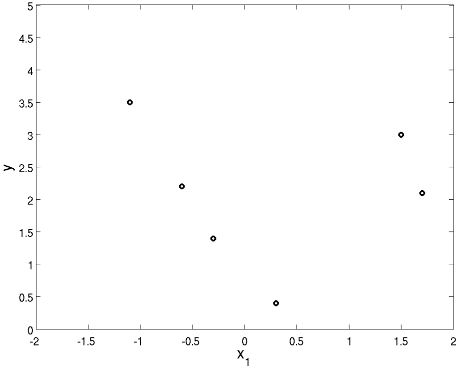


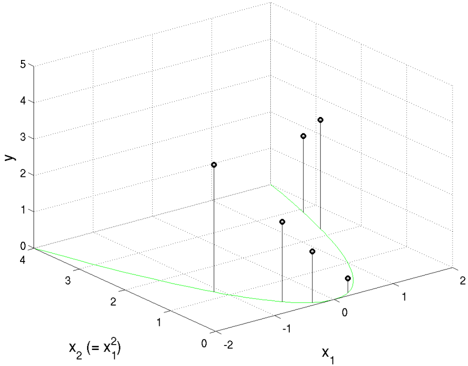
Non-linear regression



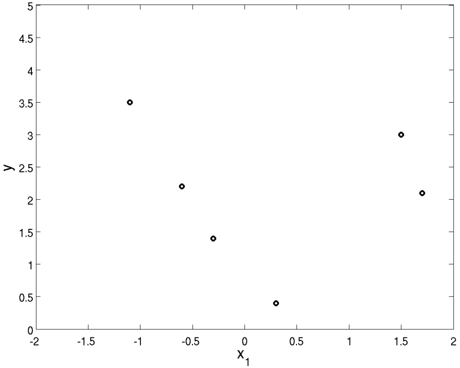


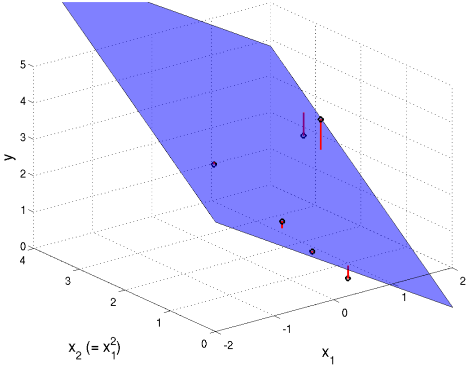
Non-linear regression



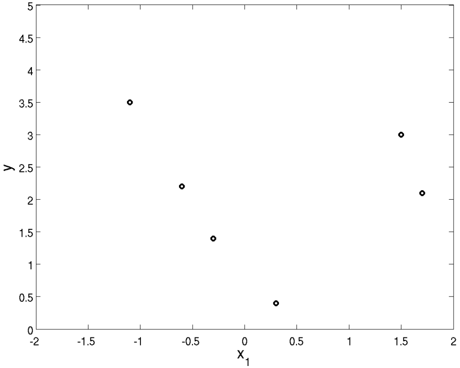


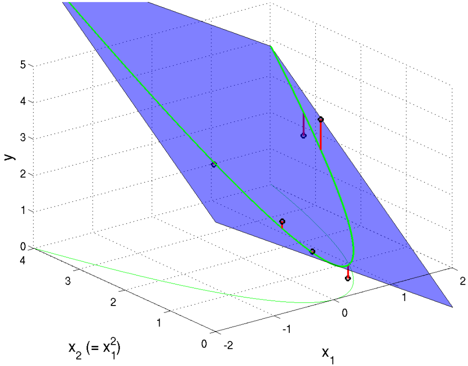
Non-linear regression



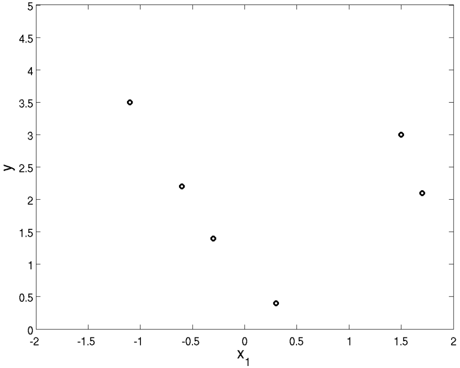


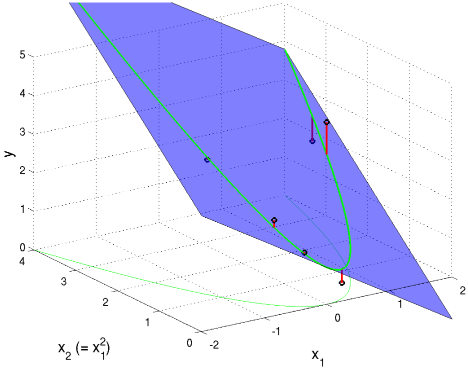
Non-linear regression



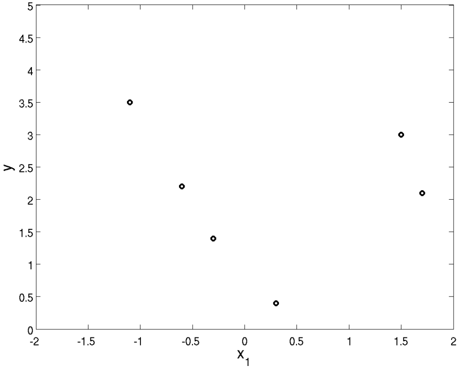


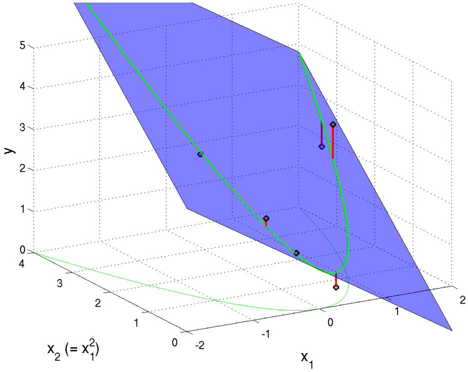
Non-linear regression



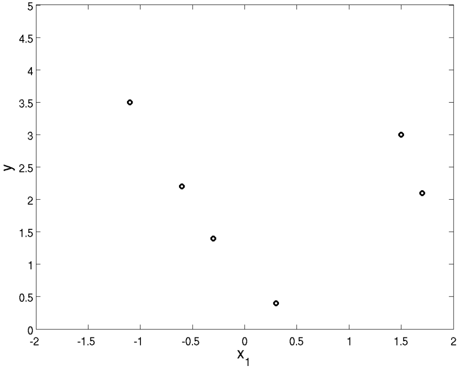


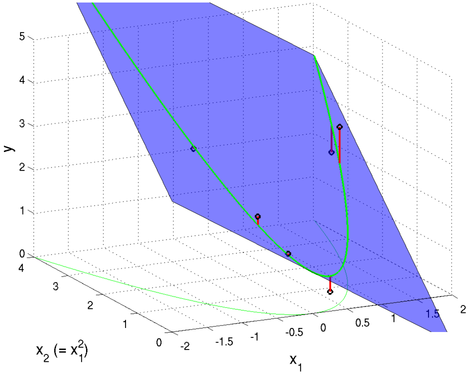
Non-linear regression



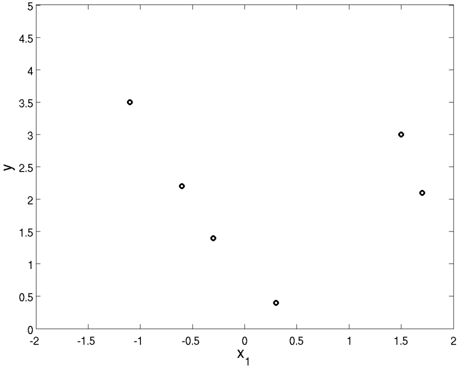


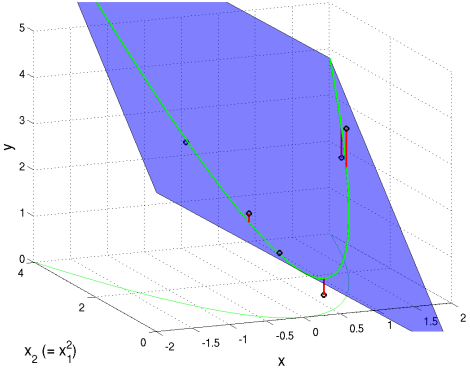
Non-linear regression



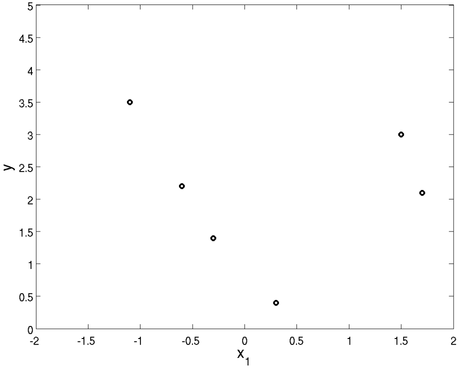


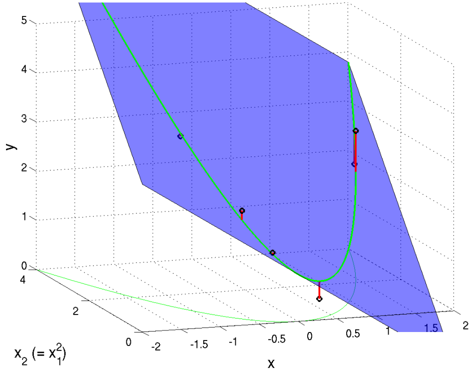
Non-linear regression



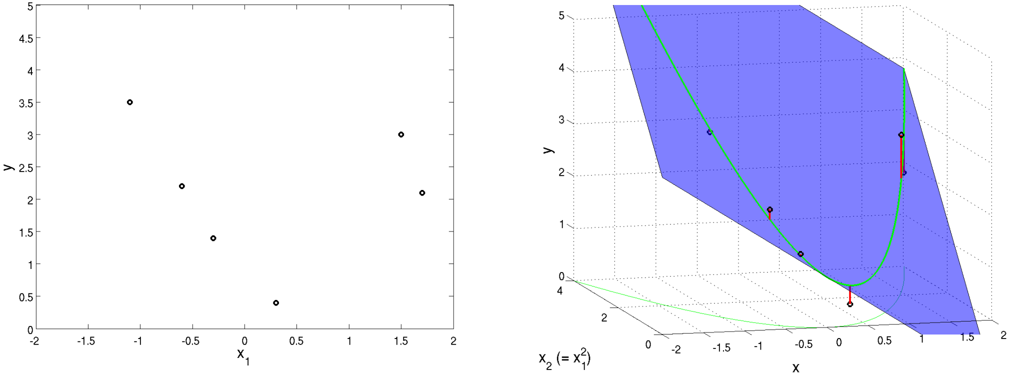


Non-linear regression

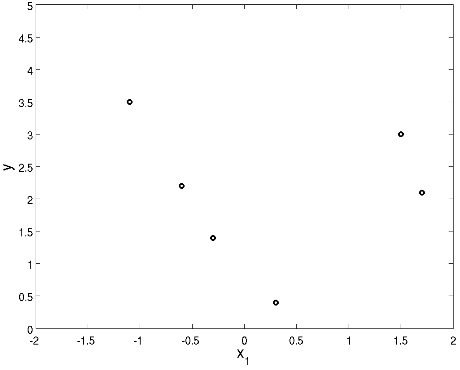


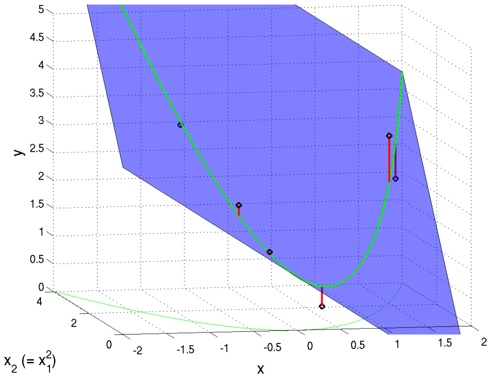


Non-linear regression

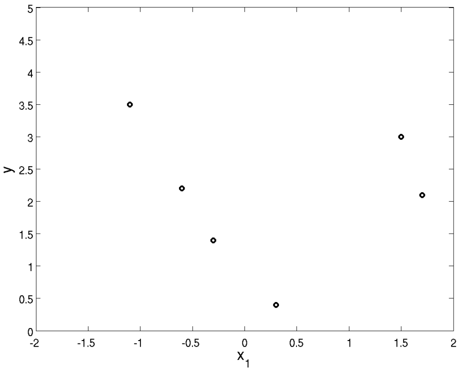


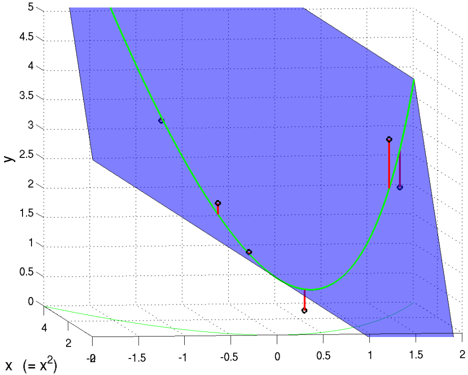
Non-linear regression



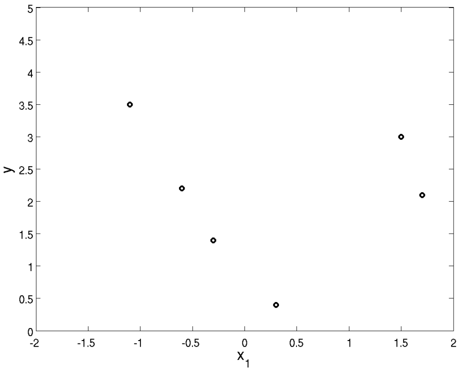


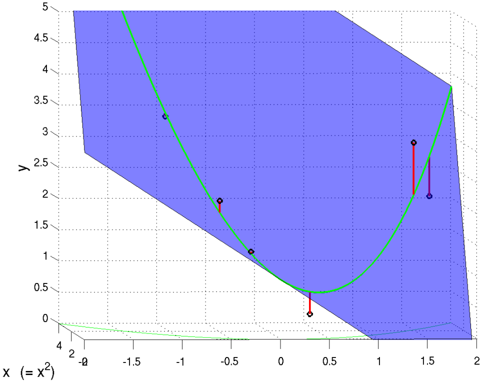
Non-linear regression



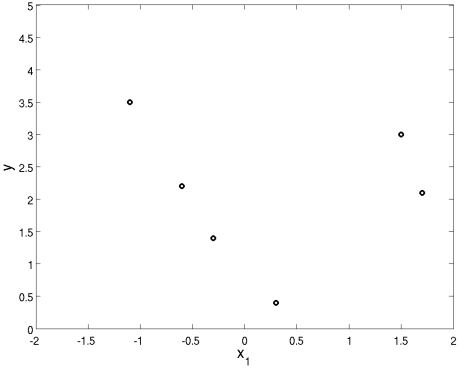


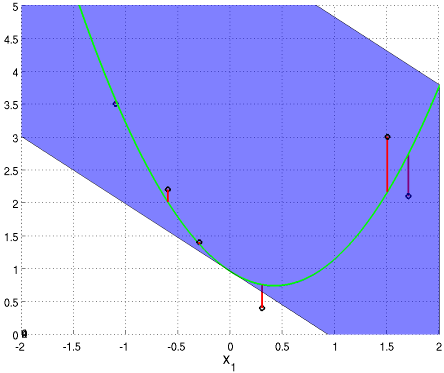
Non-linear regression



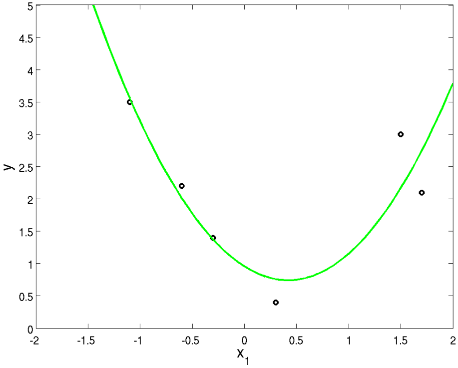


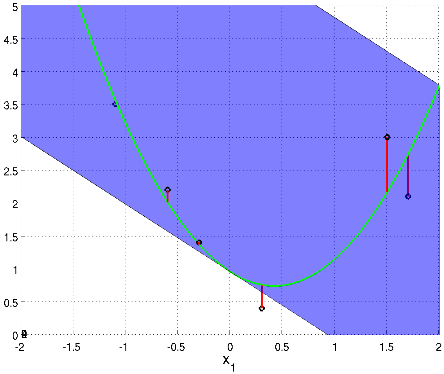
Non-linear regression



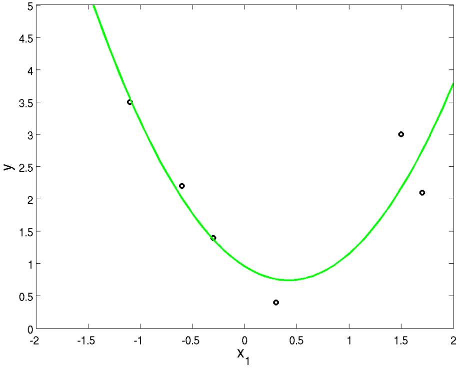


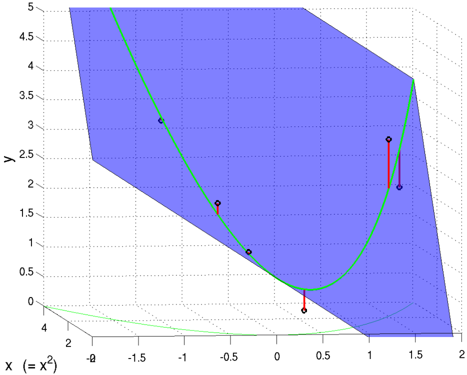
Non-linear regression



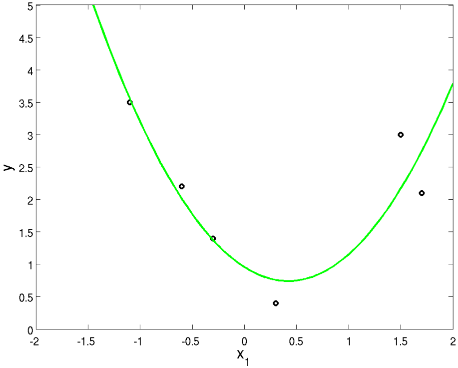


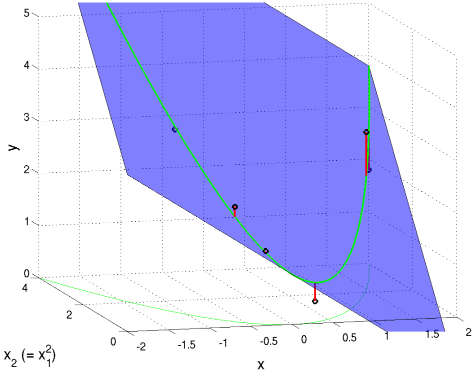
Non-linear regression



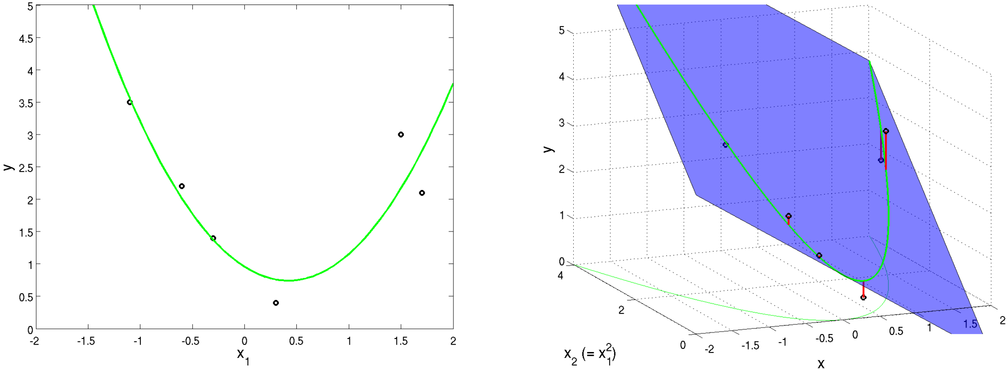


Non-linear regression

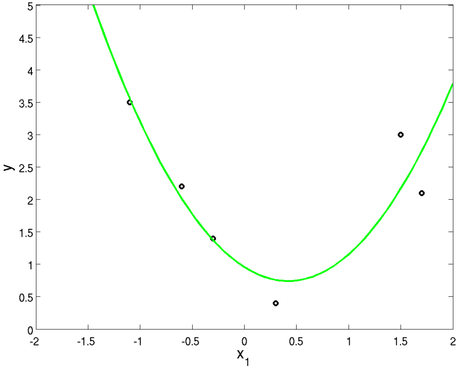


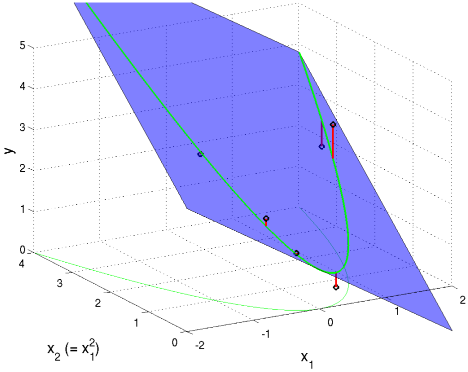


Non-linear regression

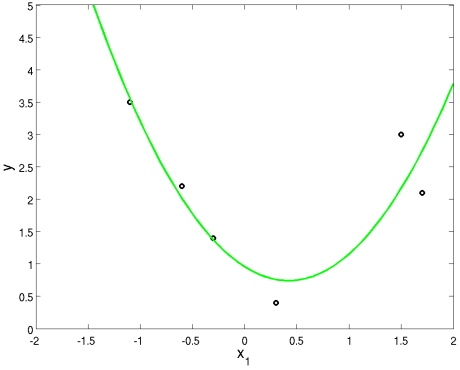


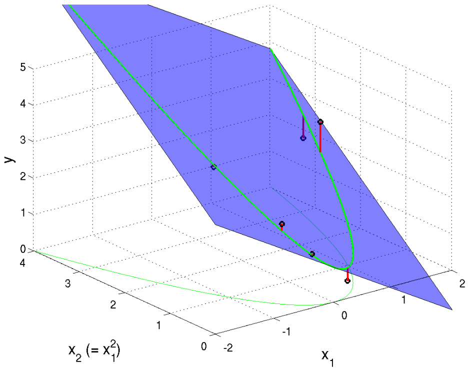
Non-linear regression



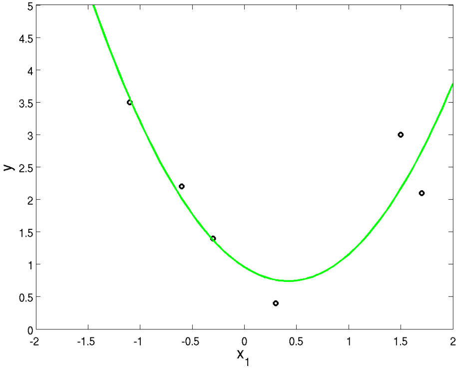


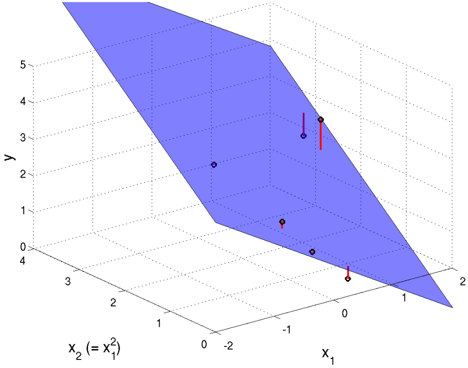
Non-linear regression





Non-linear regression





Feature Mapping

The function that takes the raw attributes and creates features from them is often written $\phi(x)$. For instance

$$\phi(x) = \begin{bmatrix}1 & x_1 & x_2\end{bmatrix} \qquad \phi(x) = \begin{bmatrix}1 & x_1 & x_2 & x_1^2 & x_2^2 & x_1x_2\end{bmatrix}$$

$$\phi(x) = \begin{bmatrix}1 & x_1 & x_1^2\end{bmatrix} \qquad \phi(x) = \begin{bmatrix}1 & x_1 & x_2 & \sin(x_1) & \sin(x_2)\end{bmatrix}$$

Feature Mapping

The function that takes the raw attributes and creates features from them is often written $\phi(x)$. For instance

$$\phi(x) = \begin{bmatrix}1 & x_1 & x_2\end{bmatrix} \qquad \phi(x) = \begin{bmatrix}1 & x_1 & x_2 & x_1^2 & x_2^2 & x_1x_2\end{bmatrix}$$

$$\phi(x) = \begin{bmatrix}1 & x_1 & x_1^2\end{bmatrix} \qquad \phi(x) = \begin{bmatrix}1 & x_1 & x_2 & \sin(x_1) & \sin(x_2)\end{bmatrix}$$

Instead of X, we sometimes call the new matrix Φ:

$$X = \begin{bmatrix} \text{---}x_1\text{---} \\ \text{---}x_2\text{---} \\ \vdots \\ \text{---}x_m\text{---} \end{bmatrix} \qquad \Phi = \begin{bmatrix} \text{---}\phi(x_1)\text{---} \\ \text{---}\phi(x_2)\text{---} \\ \vdots \\ \text{---}\phi(x_m)\text{---} \end{bmatrix}$$

Feature Mapping

The function that takes the raw attributes and creates features from them is often written $\phi(x)$. For instance

$$\phi(x) = \begin{bmatrix} 1 & x_1 & x_2 \end{bmatrix} \qquad \phi(x) = \begin{bmatrix} 1 & x_1 & x_2 & x_1^2 & x_2^2 & x_1 x_2 \end{bmatrix}$$

$$\phi(x) = \begin{bmatrix} 1 & x_1 & x_1^2 \end{bmatrix} \qquad \phi(x) = \begin{bmatrix} 1 & x_1 & x_2 & \sin(x_1) & \sin(x_2) \end{bmatrix}$$

Instead of X, we sometimes call the new matrix Φ:

$$X = \begin{bmatrix} \text{---} x_1 \text{---} \\ \text{---} x_2 \text{---} \\ \vdots \\ \text{---} x_m \text{---} \end{bmatrix} \qquad \Phi = \begin{bmatrix} \text{---} \phi(x_1) \text{---} \\ \text{---} \phi(x_2) \text{---} \\ \vdots \\ \text{---} \phi(x_m) \text{---} \end{bmatrix}$$

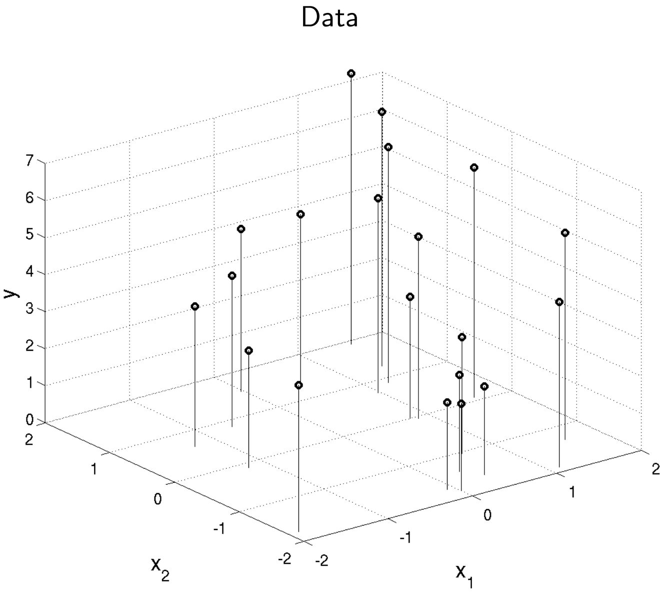
The learning equation correspondingly changes notation:

$$\hat{w} = (X^\top X)^{-1}(X^\top Y) \qquad \hat{w} = (\Phi^\top \Phi)^{-1}(\Phi^\top Y)$$

And of course the resulting function changes equation too:

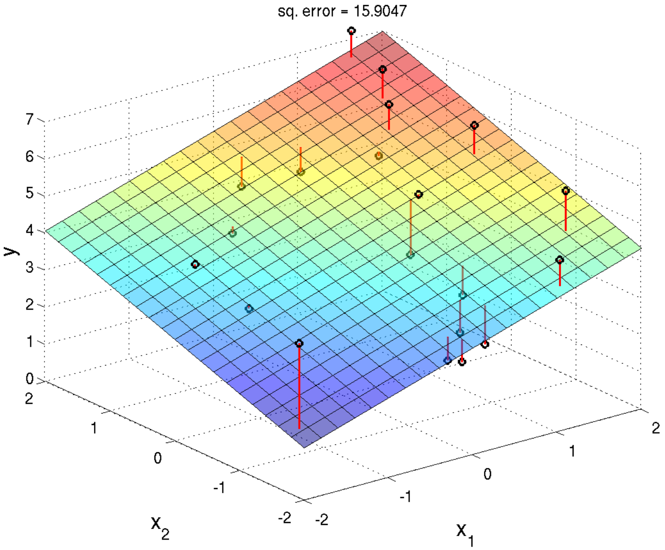
$$f(x) = x^\top \hat{w} \qquad f(x) = \phi(x)^\top \hat{w}$$

Overfitting



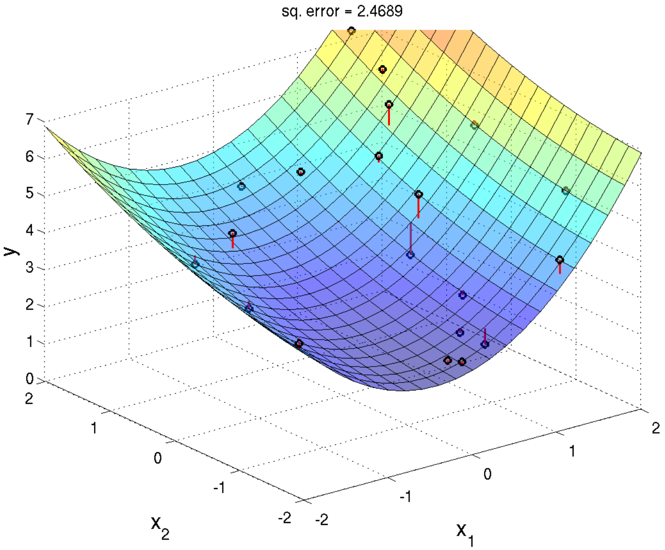
Overfitting

Linear: $\phi(x) = \begin{bmatrix} 1 & x_1 & x_2 \end{bmatrix}$



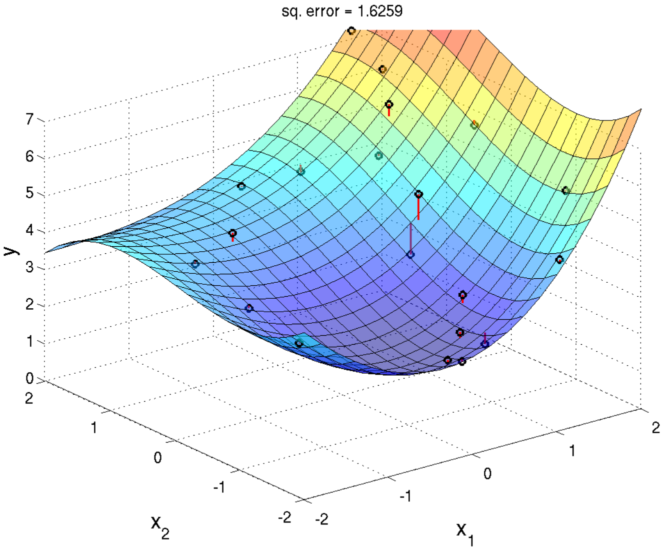
Overfitting

2nd order: $\phi(x) = \begin{bmatrix}1 & x_1 & x_2 & x_1^2 & x_1 x_2 & x_2^2\end{bmatrix}$



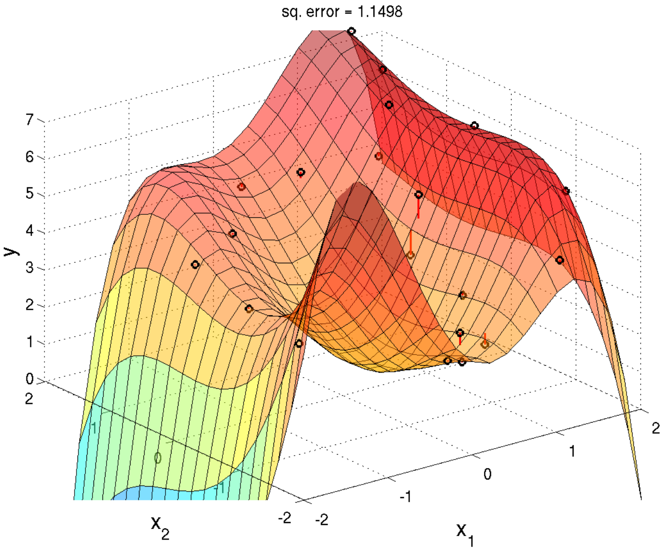
Overfitting

3rd order: $\phi(x) = [1 \quad x_1 \quad x_2 \quad x_1^2 \quad x_1x_2 \quad x_2^2 \quad x_1^3 \quad x_1^2x_2 \quad x_1x_2^2 \quad x_2^3]$



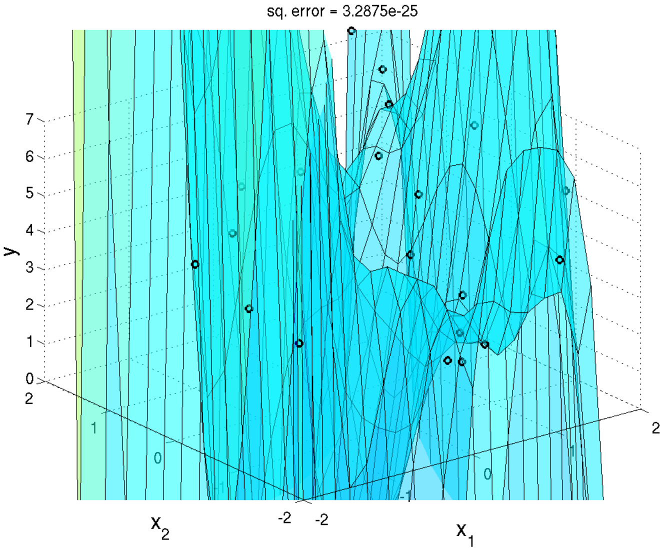
Overfitting

4th order: $\phi(x) = \begin{bmatrix} 1 & x_1 & x_2 & x_1^2 & x_1x_2 & x_2^2 & x_1^3 & x_1^2x_2 & x_1x_2^2 & x_2^3 & x_1^4 & x_1^3x_2 & x_1^2x_2^2 & x_1x_2^3 & x_2^4 \end{bmatrix}$



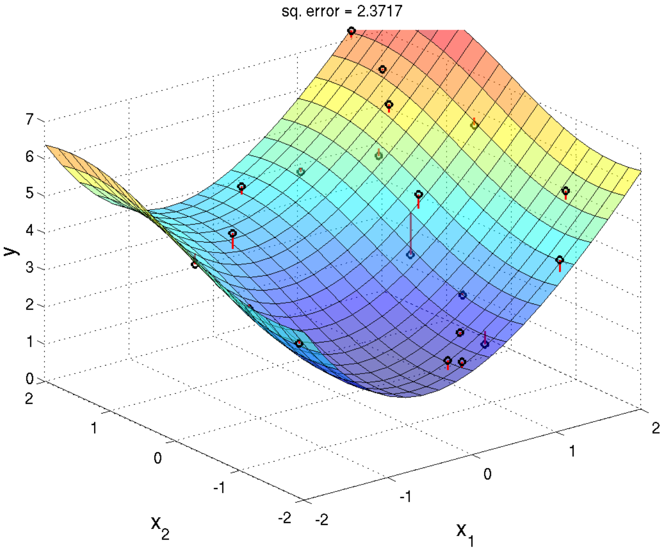
Overfitting

5th order: $\phi(x) = \begin{bmatrix} 1 & x_1 & x_2 & x_1^2 & x_1 x_2 & x_2^2 & x_1^3 & x_1^2 x_2 & \ldots & x_2^5 \end{bmatrix}$



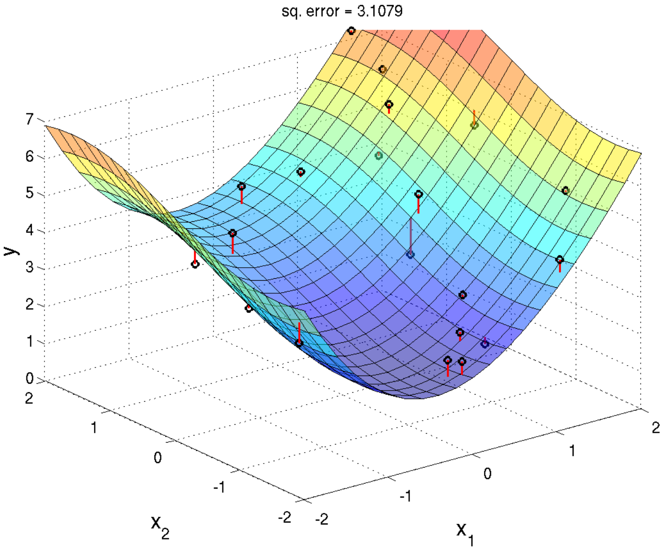
Overfitting

"correct": $\phi(x) = \begin{bmatrix} 1 & \sin(x_1) & \sin(x_2) & x_1^2 \end{bmatrix}$

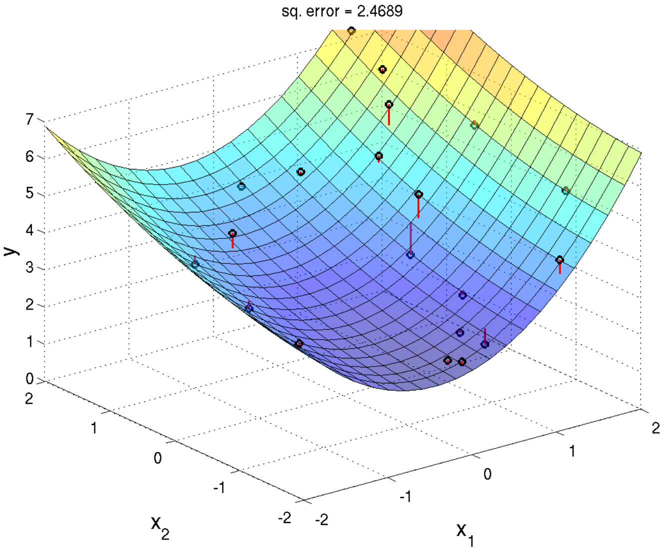


Overfitting

really "correct": $\phi(x) = \left[3 + \sin(x_1) + \sin(x_2) + x_1^2\right]$

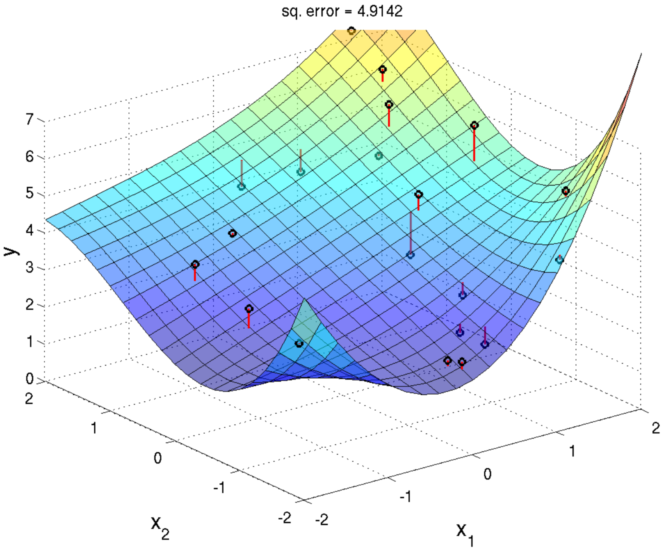


2nd order: $\phi(x) = \begin{bmatrix} 1 & x_1 & x_2 & x_1^2 & x_1x_2 & x_2^2 \end{bmatrix}$



Overfitting

some 5th order: $\phi(x) = \begin{bmatrix} 1 & x_1 & x_2 & x_1^2 x_2 & x_1^3 x_2^2 & x_1^2 x_2^3 \end{bmatrix}$



Regularization

Need a way to discourage the learning algorithm from using all of the features.

Regularization

Need a way to discourage the learning algorithm from using all of the features.

Or at least using them "as much."

Regularization

Need a way to discourage the learning algorithm from using all of the features.
Or at least using them "as much."

$$L = \sum_{i=1}^{m} \left(y_i - \sum_{j=1}^{n} w_j x_{i,j} \right)^2$$

Regularization

Need a way to discourage the learning algorithm from using all of the features.
Or at least using them "as much."

$$L = \sum_{i=1}^{m} \left(y_i - \sum_{j=1}^{n} w_j x_{i,j} \right)^2 + \lambda \sum_{j=1}^{n} w_j^2$$

Regularization

Need a way to discourage the learning algorithm from using all of the features.
Or at least using them "as much."

$$L = \sum_{i=1}^{m} \left(y_i - \sum_{j=1}^{n} w_j x_{i,j} \right)^2 + \lambda \sum_{j=1}^{n} w_j^2$$

This is called "ridge regression" or "LLS with L_2 regularization."

Regularization

Need a way to discourage the learning algorithm from using all of the features.
Or at least using them "as much."

$$L = \sum_{i=1}^{m} \left(y_i - \sum_{j=0}^{n} w_j x_{i,j} \right)^2 + \lambda \sum_{j=1}^{n} w_j^2$$

This is called "ridge regression" or "LLS with L_2 regularization."
If we have a constant feature, we generally do not include it in the regularization.

Ridge Regression

$$L = \sum_{i=1}^{m} \left(y_i - \sum_{j=1}^{n} w_j x_{i,j} \right)^2$$

Ridge Regression

$$L = \sum_{i=1}^{m} \left(y_i - \sum_{j=1}^{n} w_j x_{i,j} \right)^2$$

$$w = \left(X^\top X \right)^{-1} X^\top Y$$

Ridge Regression

$$L = \sum_{i=1}^{m} \left(y_i - \sum_{j=1}^{n} w_j x_{i,j} \right)^2 + \lambda \sum_{j=1}^{n} w_j^2$$

$$w = \left(X^\top X + \lambda I \right)^{-1} X^\top Y$$

Ridge Regression

$$L = \sum_{i=1}^{m} \left(y_i - \sum_{j=1}^{n} w_j x_{i,j} \right)^2 + \lambda \sum_{j=1}^{n} w_j^2$$

$$w = \left(X^\top X + \lambda \begin{bmatrix} 1 & 0 & & 0 \\ 0 & 1 & & 0 \\ & & \ddots & \\ 0 & 0 & & 1 \end{bmatrix} \right)^{-1} X^\top Y$$

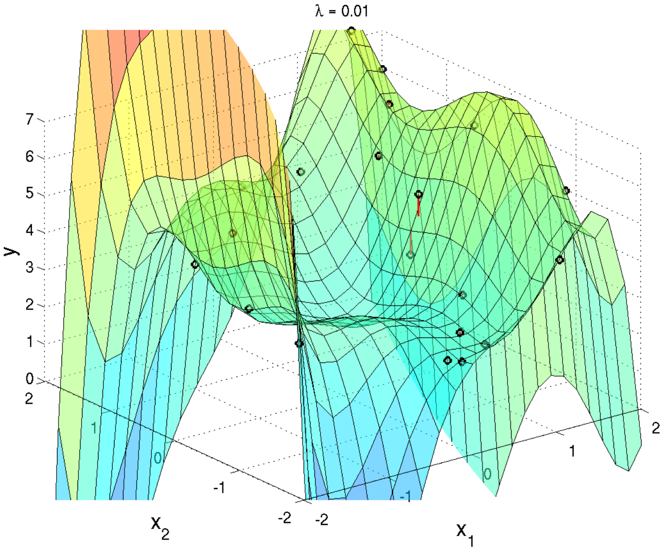
Ridge Regression

$$L = \sum_{i=1}^{m} \left(y_i - \sum_{j=0}^{n} w_j x_{i,j} \right)^2 + \lambda \sum_{j=1}^{n} w_j^2$$

$$w = \left(X^\top X + \lambda \begin{bmatrix} 0 & 0 & & 0 \\ 0 & 1 & & 0 \\ & & \ddots & \\ 0 & 0 & & 1 \end{bmatrix} \right)^{-1} X^\top Y$$

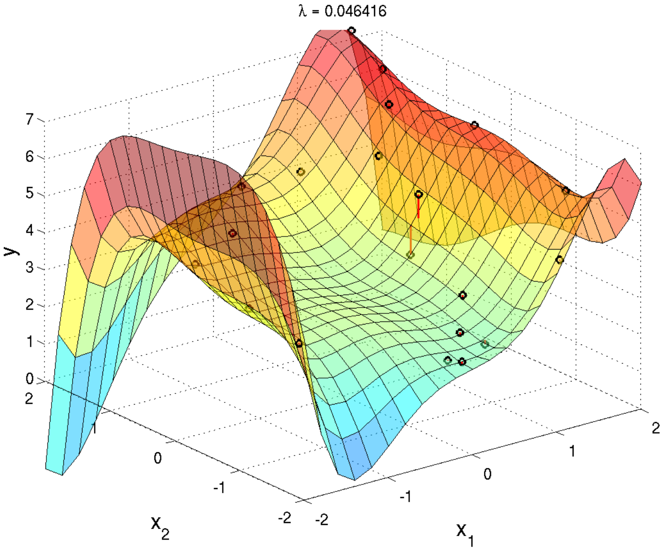
Regularization

5th order polynomial



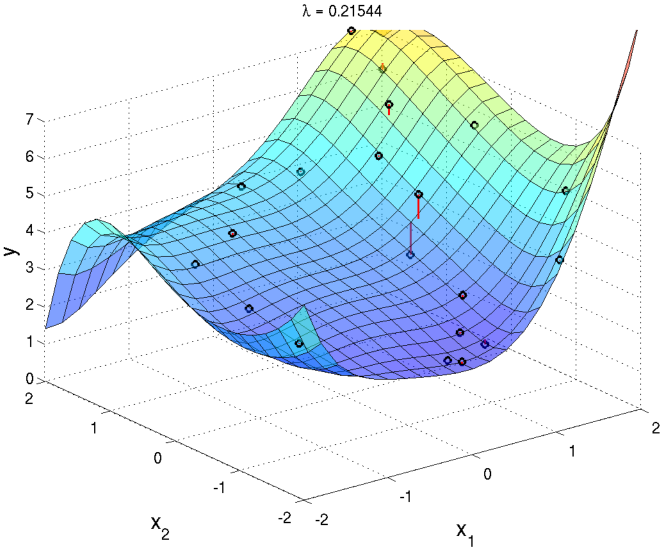
Regularization

5th order polynomial



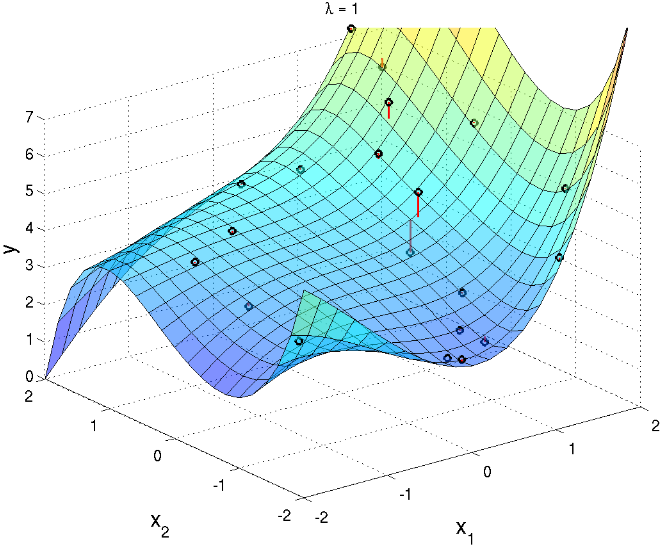
Regularization

5th order polynomial

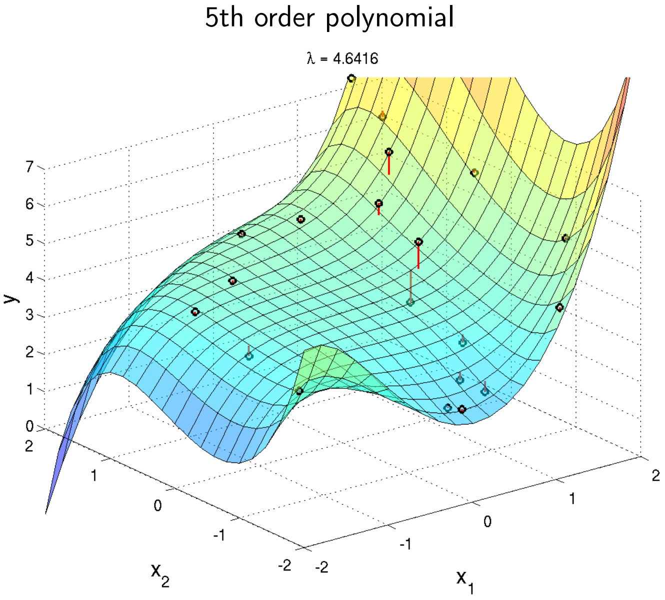


Regularization

5th order polynomial

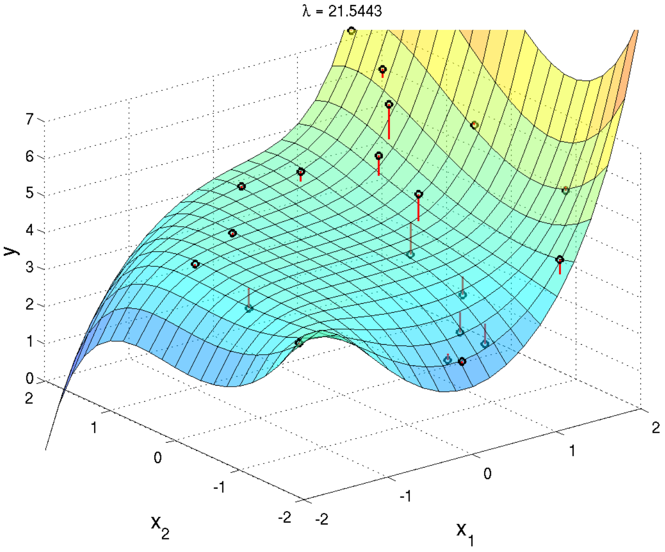


Regularization



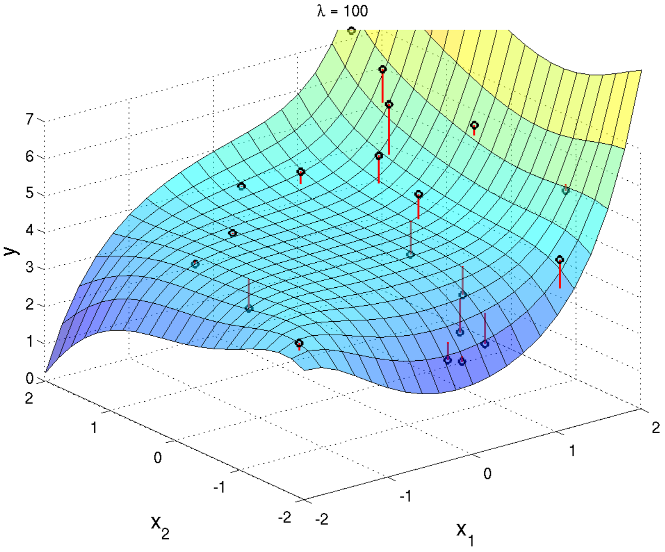
Regularization

5th order polynomial



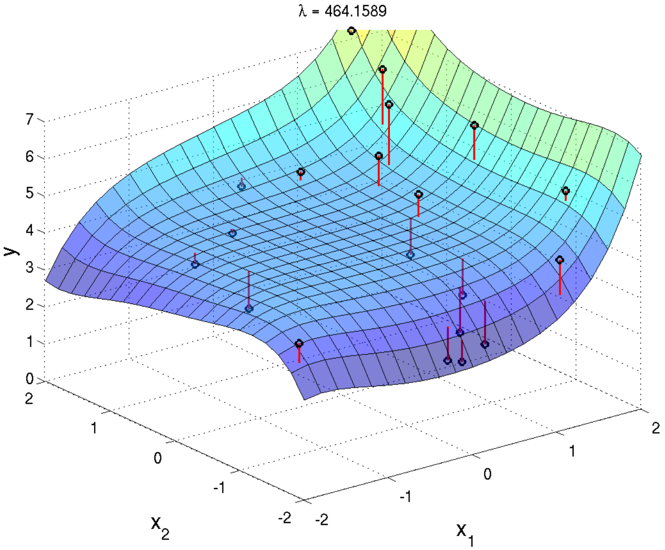
Regularization

5th order polynomial

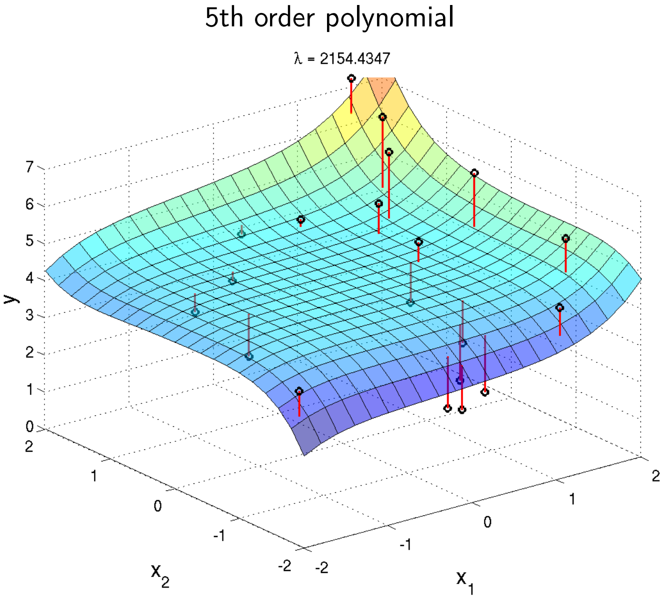


Regularization

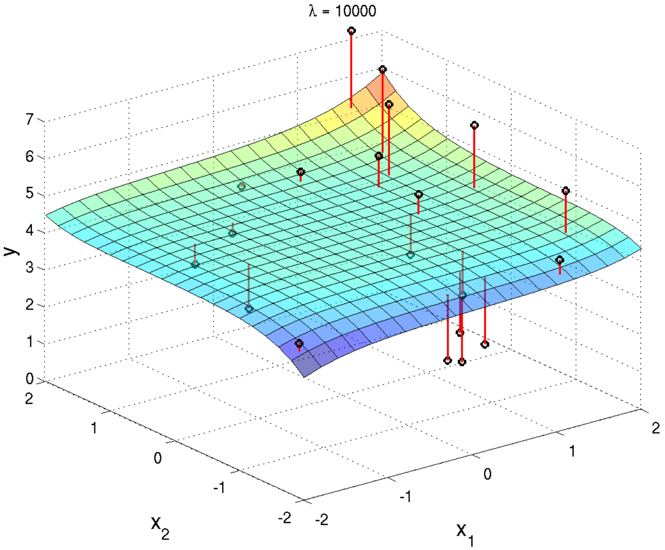
5th order polynomial



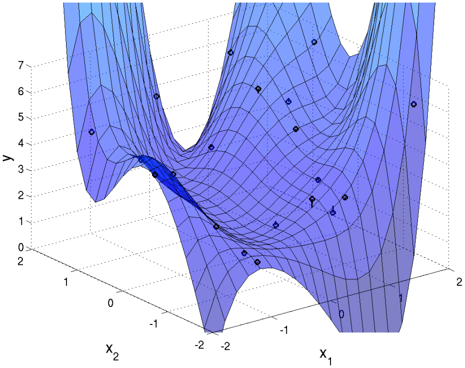
Regularization



5th order polynomial

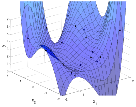


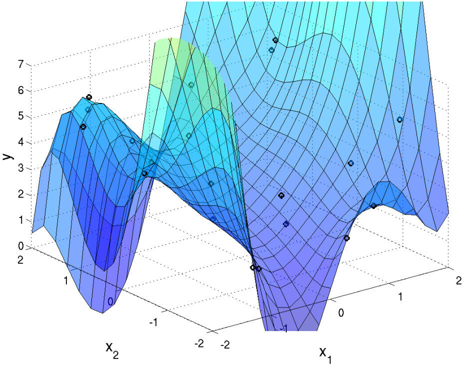
$$\lambda = 0.02$$



Bias-Variance Trade-off

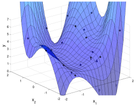
$$\lambda = 0.02$$

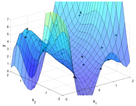


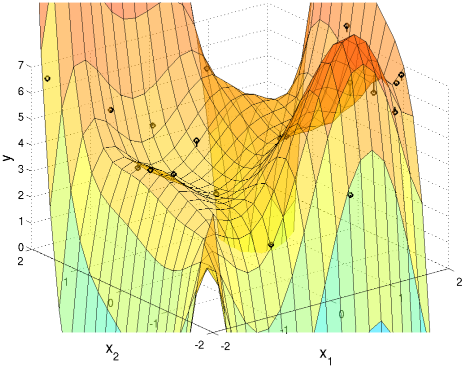


Bias-Variance Trade-off

$$\lambda = 0.02$$

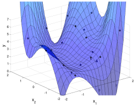


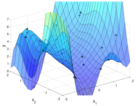


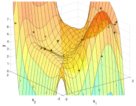


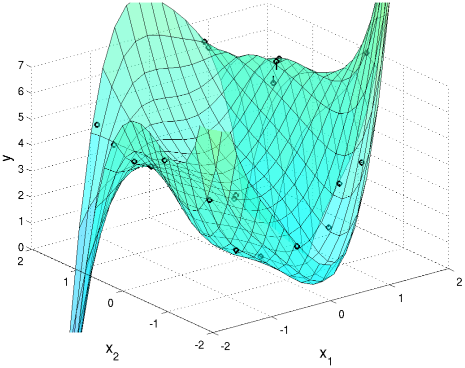
Bias-Variance Trade-off

$\lambda = 0.02$



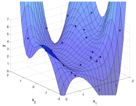


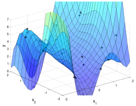


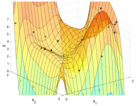


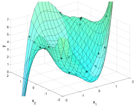
Bias-Variance Trade-off

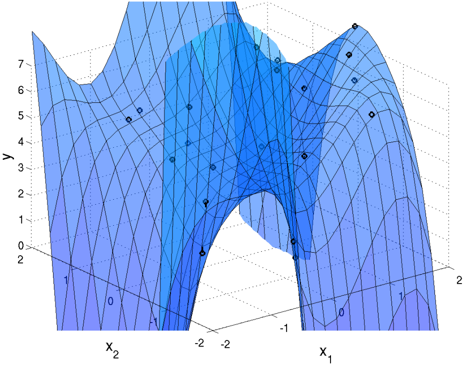
$\lambda = 0.02$





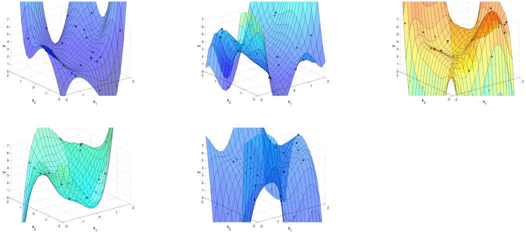


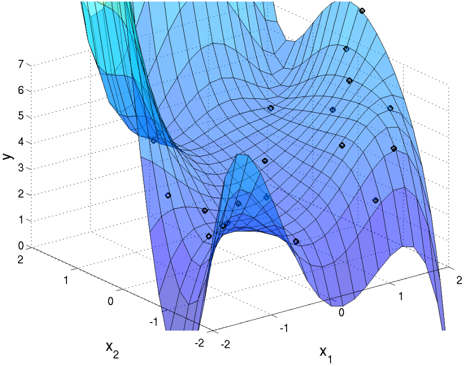




Bias-Variance Trade-off

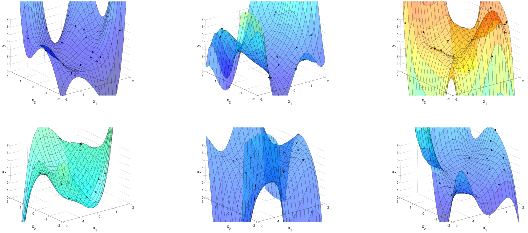
$\lambda = 0.02$

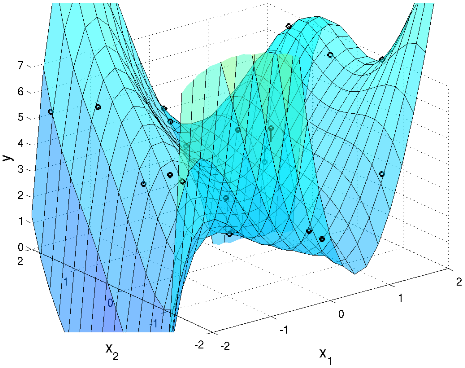




Bias-Variance Trade-off

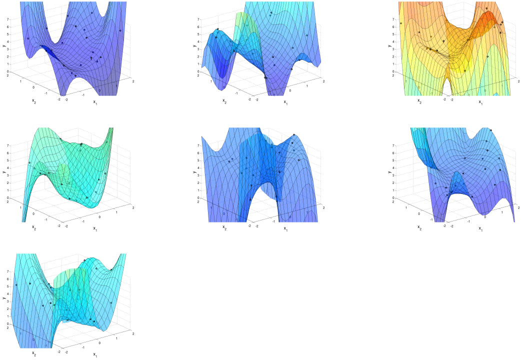
$$\lambda = 0.02$$

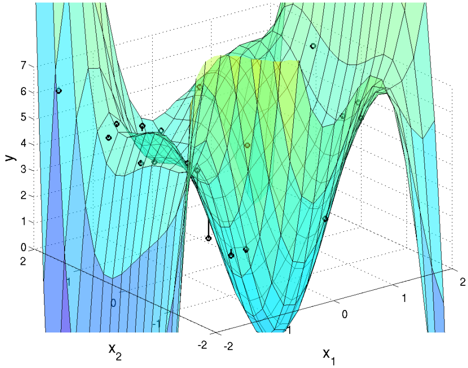




Bias-Variance Trade-off

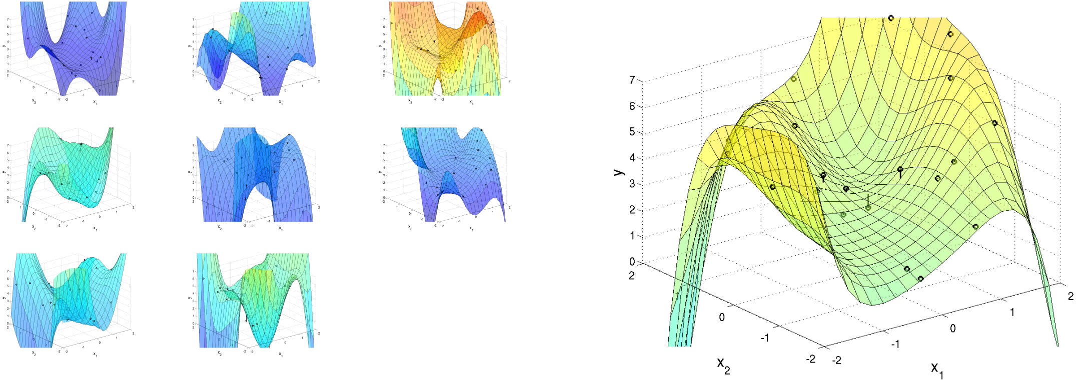
$\lambda = 0.02$





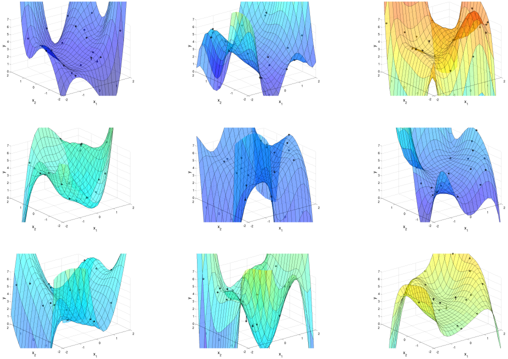
Bias-Variance Trade-off

$\lambda = 0.02$



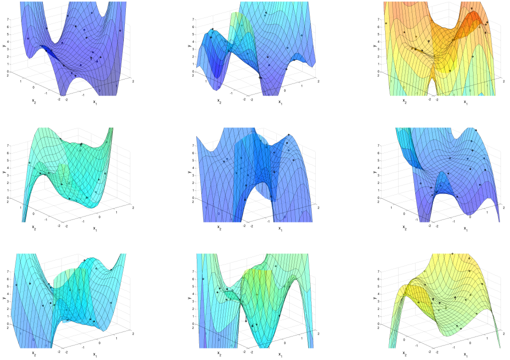
Bias-Variance Trade-off

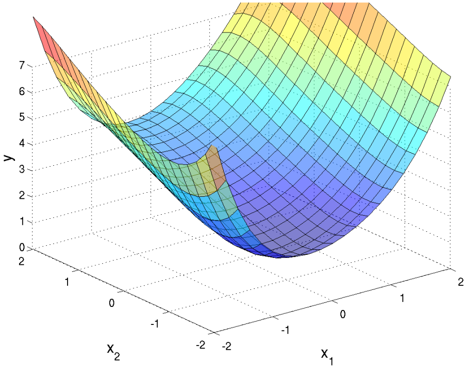
$$\lambda = 0.02$$



Bias-Variance Trade-off

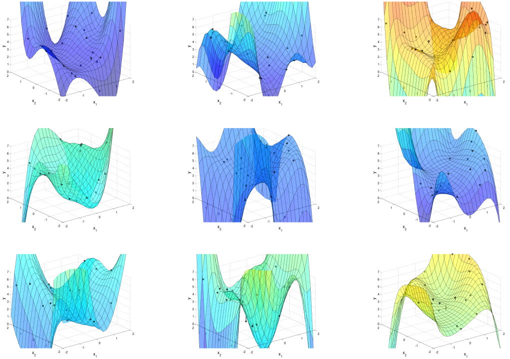
$$\lambda = 0.02$$

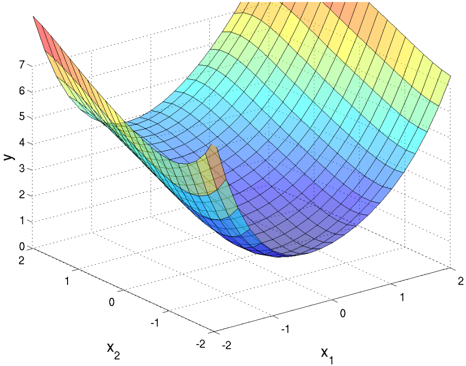




Bias-Variance Trade-off

$$\lambda = 0.02$$



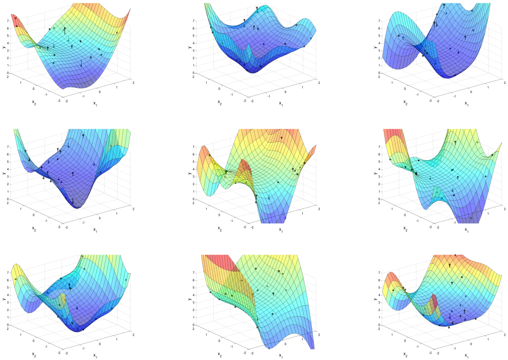


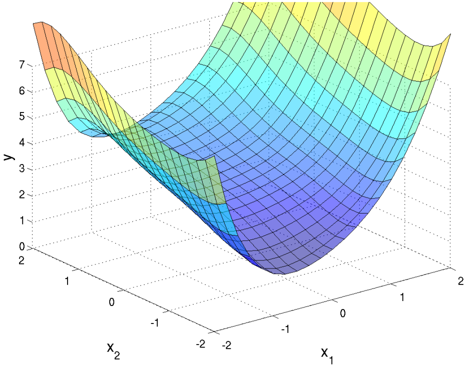
High Variance

Low Bias

Bias-Variance Trade-off

$\lambda = 1$



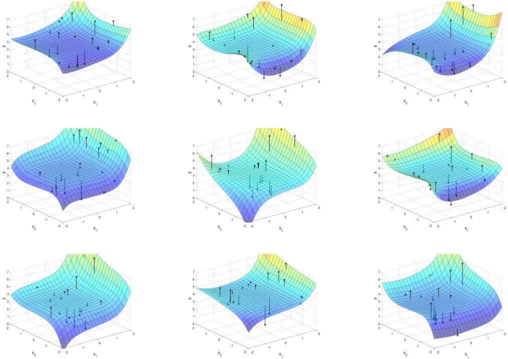


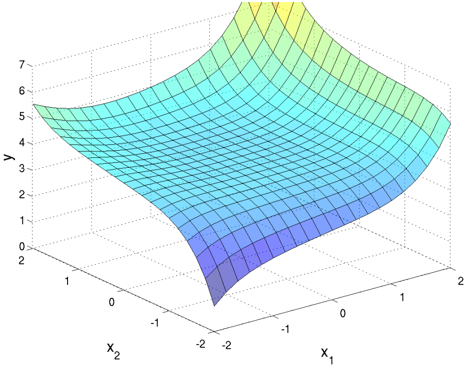
Moderate Variance

Lowish Bias

Bias-Variance Trade-off

$$\lambda = 500$$





Low Variance

High Bias