



Model Non-linearization, Overfitting & Regularization

Qinliang Su (苏勤亮)

Sun Yat-sen University

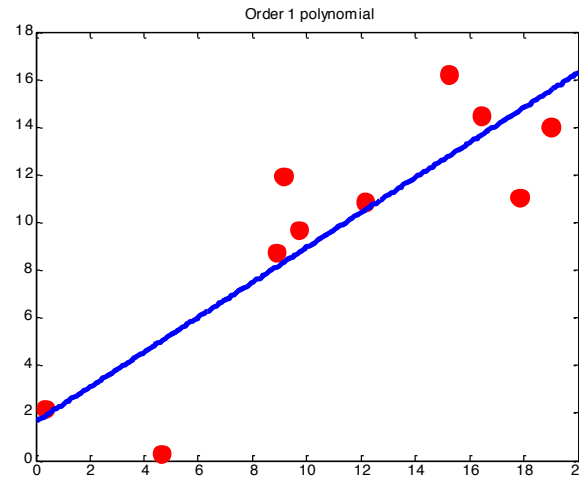
suqliang@mail.sysu.edu.cn

Outline

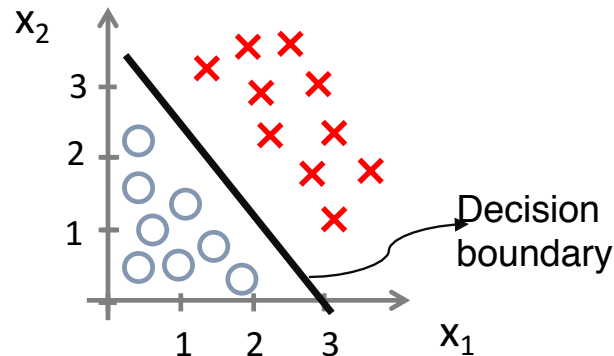
- Model Non-linearization
- Overfitting
- Model Selection
- Regularization

Introduction

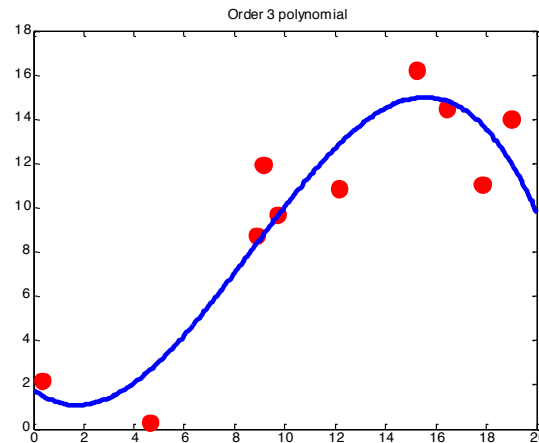
- Only linear relation between input x and output y can be modelled by linear regression



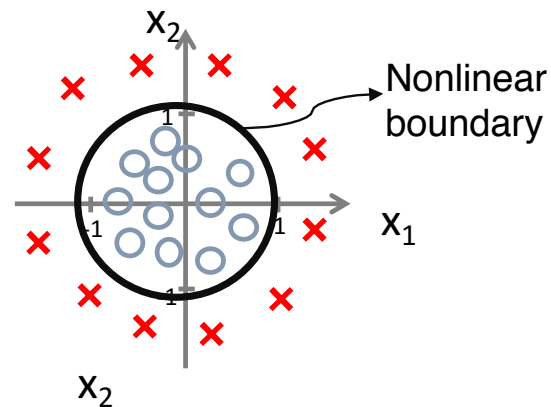
- For linear classifiers, the decision boundaries can only be linear



- For more complex tasks, models should have the ability to capture
 - nonlinear input-output relation



- nonlinear decision boundaries



How to build models possessing nonlinear modeling ability?

Basic idea: Non-linearizing the linear models with basis functions

$$[x] \rightarrow [x, x^2, x^3]$$

Non-linearization via Basis Functions

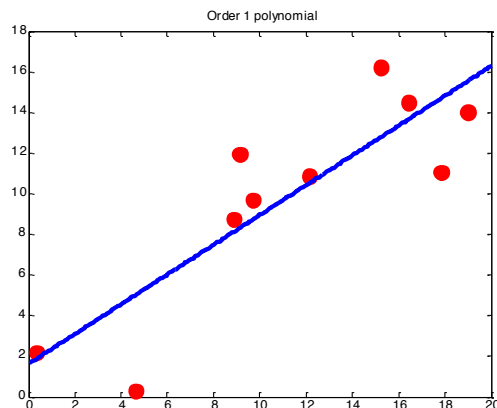
- Transform the features by polynomial

$$[x] \rightarrow [x, x^2, x^3]$$

Single feature is expanded into 3 features

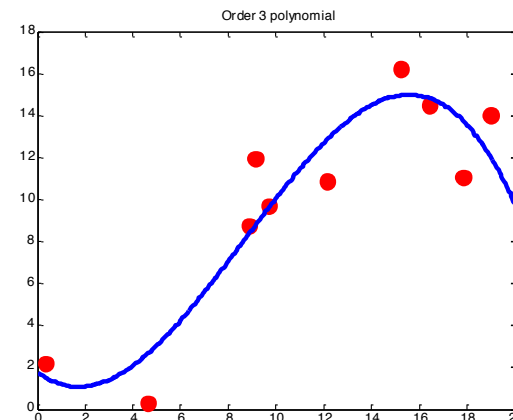
- Model with original feature

$$\begin{aligned} f(x) &= w_0 + w_1 x \\ &= [1, x] \mathbf{w} \end{aligned}$$



- Model with expanded features

$$\begin{aligned} f(x) &= w_0 + w_1 x + w_2 x^2 + w_3 x^3 \\ &= \phi(x) \mathbf{w} \end{aligned}$$



- Generally, the transformation could be expressed as

$$[x_1, x_2, \dots, x_m] \rightarrow [\phi_1(\mathbf{x}), \phi_2(\mathbf{x}), \dots, \phi_n(\mathbf{x})] \triangleq \boldsymbol{\phi}(\mathbf{x})$$

Basis function

$\phi_k(\mathbf{x})$ could be any functions that produce meaningful features,
e.g.,

$$\sqrt{x}, \quad \log x, \quad \frac{1}{x}, \quad x_1 + x_2, \quad x_1 - x_2, \quad x_1 x_2$$

- The non-linearized model now becomes

$$f(\mathbf{x}) = \boldsymbol{\phi}(\mathbf{x})\mathbf{w}$$

which is called **basis function model**

The basis function model is **nonlinear w.r.t. \mathbf{x}** , but is **still linear w.r.t. the model parameters \mathbf{w}**

- With the nonlinearly transformed feature $\phi(x)$, the optimal model parameters \mathbf{w}^* for regression is obtained by optimizing the loss

$$L(\mathbf{w}) = \frac{1}{N} \|\Phi(X)\mathbf{w} - \mathbf{y}\|^2$$

where $\Phi(X) \triangleq \begin{bmatrix} \phi(x^{(1)}) \\ \vdots \\ \phi(x^{(N)}) \end{bmatrix}$

- With the notation $\Phi = \Phi(X)$, the optimal model parameters \mathbf{w}^* is

$$\mathbf{w}^* = (\Phi^T \Phi)^{-1} \Phi^T \mathbf{y}$$

The same as linear regression *except X is replaced by Φ*

- We can also employ the numerical methods, e.g., gradient descent, to obtain the optimal solution

- For the classification using the basis functions, the cross-entropy loss becomes

$$L(\mathbf{W}) = -\frac{1}{N} \sum_{\ell=1}^N \sum_{k=1}^K y_k^{(\ell)} \log[\text{softmax}_k(\boldsymbol{\phi}(\mathbf{x}^{(\ell)})\mathbf{W})]$$

The optimal \mathbf{W}^* can only be obtained by numerical methods

- Denoting $\boldsymbol{\phi}(\mathbf{x}^{(\ell)})$ as $\boldsymbol{\phi}^{(\ell)}$, the gradient can be derived equal to

$$\frac{\partial L(\mathbf{W})}{\partial \mathbf{w}_j} = \frac{1}{N} \sum_{\ell=1}^N \left(\text{softmax}_j(\boldsymbol{\phi}^{(\ell)}\mathbf{W}) - y_j^{(\ell)} \right) \boldsymbol{\phi}^{(\ell)T}$$

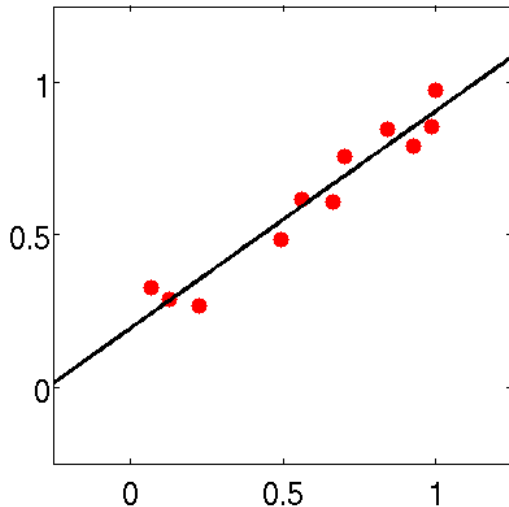
The same as multi-class logistic regression *except that $\mathbf{x}^{(\ell)}$ is replaced by $\boldsymbol{\phi}^{(\ell)}$*

Outline

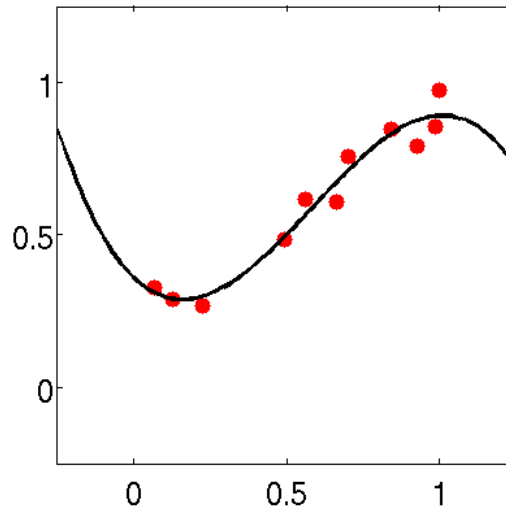
- Model Non-linearization
- **Overfitting**
- Model Selection
- Regularization

Overfitting

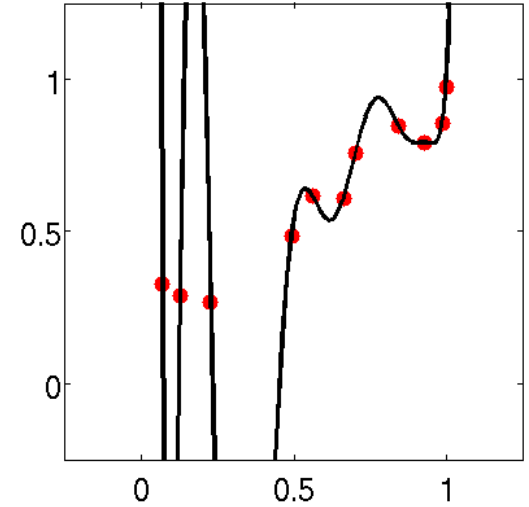
- Higher-dimensional features $\phi(x)$ tend to *fit training data better*



1-order



3-order

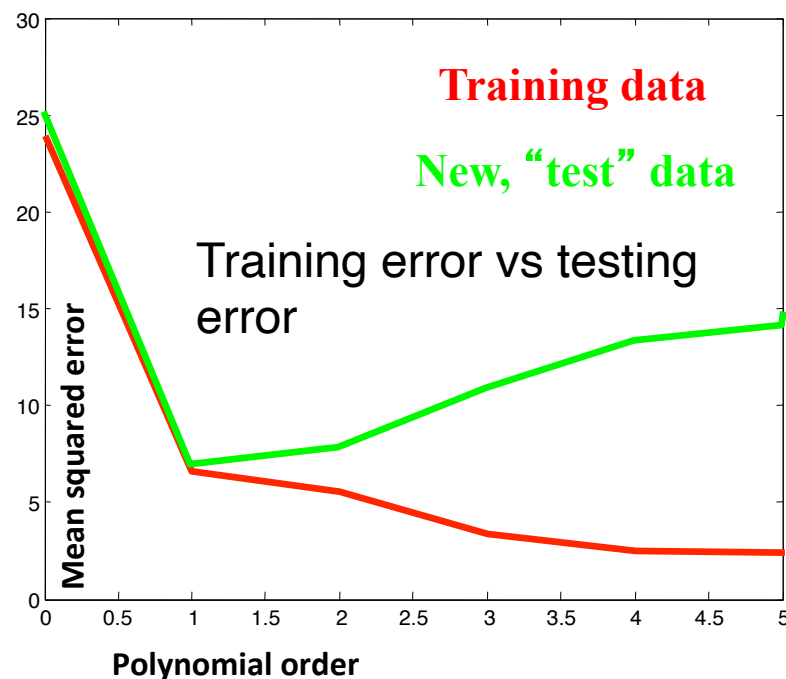
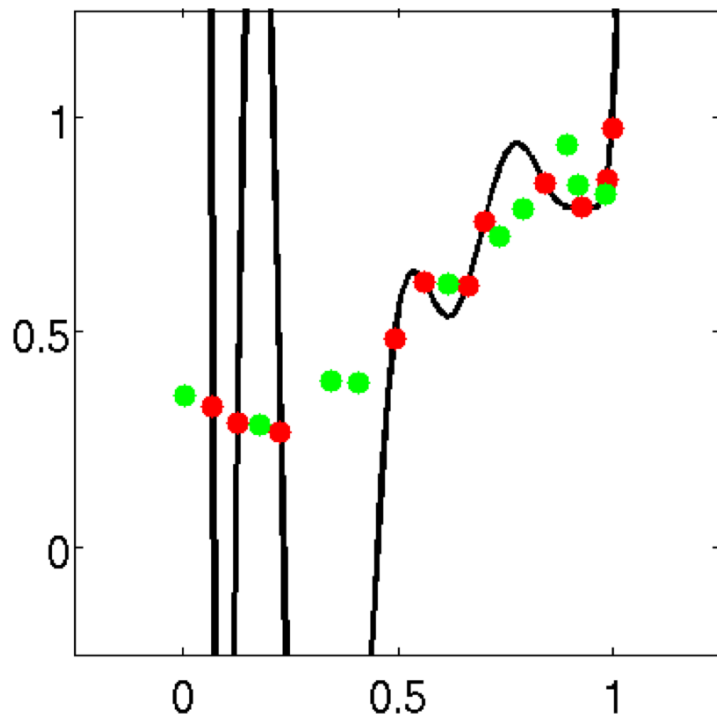


5-order

Which model is better??

- From the viewpoint of fitting training data, of course, the higher the model order is, the better the fitting looks

- But high-order models may *perform poor on the testing data*



The ability that a model can perform well on unseen data is called the *generalization ability of the model*

Model Complexity

- Each model corresponds to a degree of complexity
- But it is difficult to give an exact expression to describe the model complexity
- In general, the model complexity depends on the number of parameters, the more parameters, the more complex a model is
- To have a model perform well, we should *balance between model's complexity and representational ability*

Outline

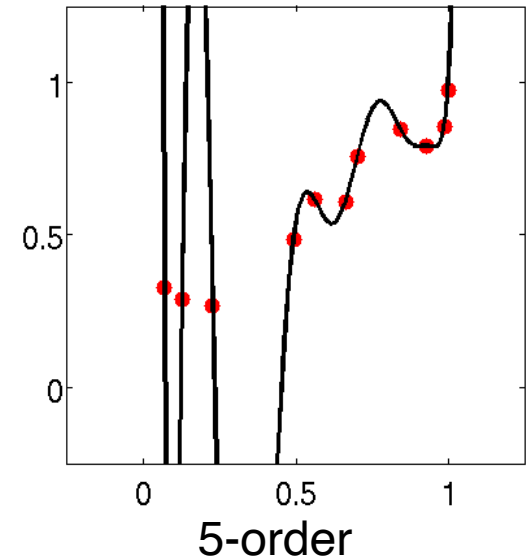
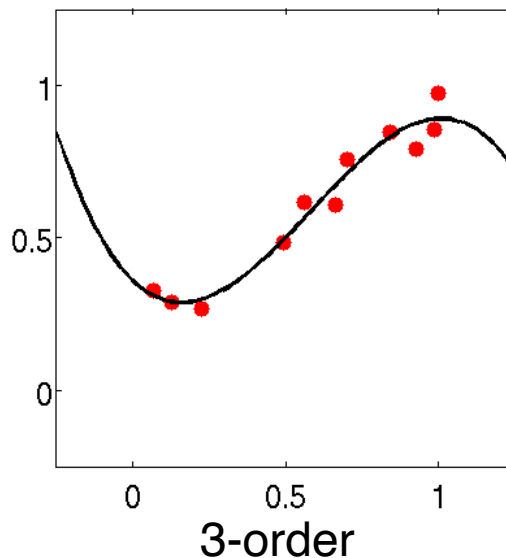
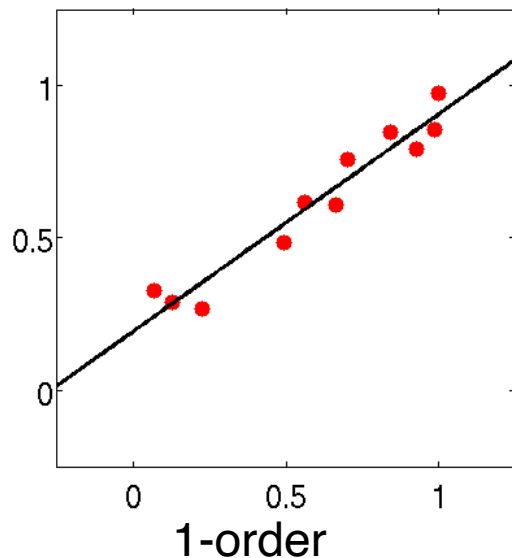
- Model Non-linearization
- Overfitting
- **Model Selection**
- Regularization

Model Selection

- **Model selection:** Given a set of models $\{\mathcal{M}_1, \dots, \mathcal{M}_m\}$, choose the one that can *perform best on the unseen testing data*

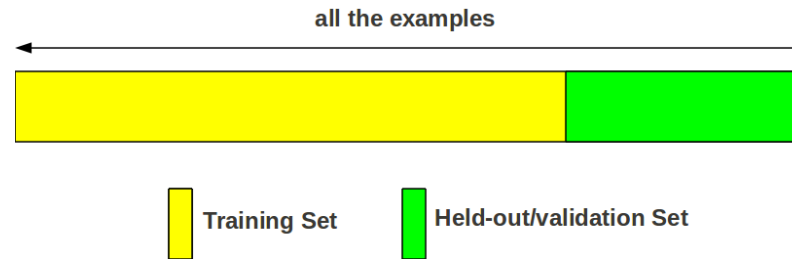
Model candidates could be of the same type, or different types

Cannot select the model based on their performance on training data



Validation Set

- Set aside a portion (20% ~ 30%) of training data as the validation set, and use the remaining as the training data

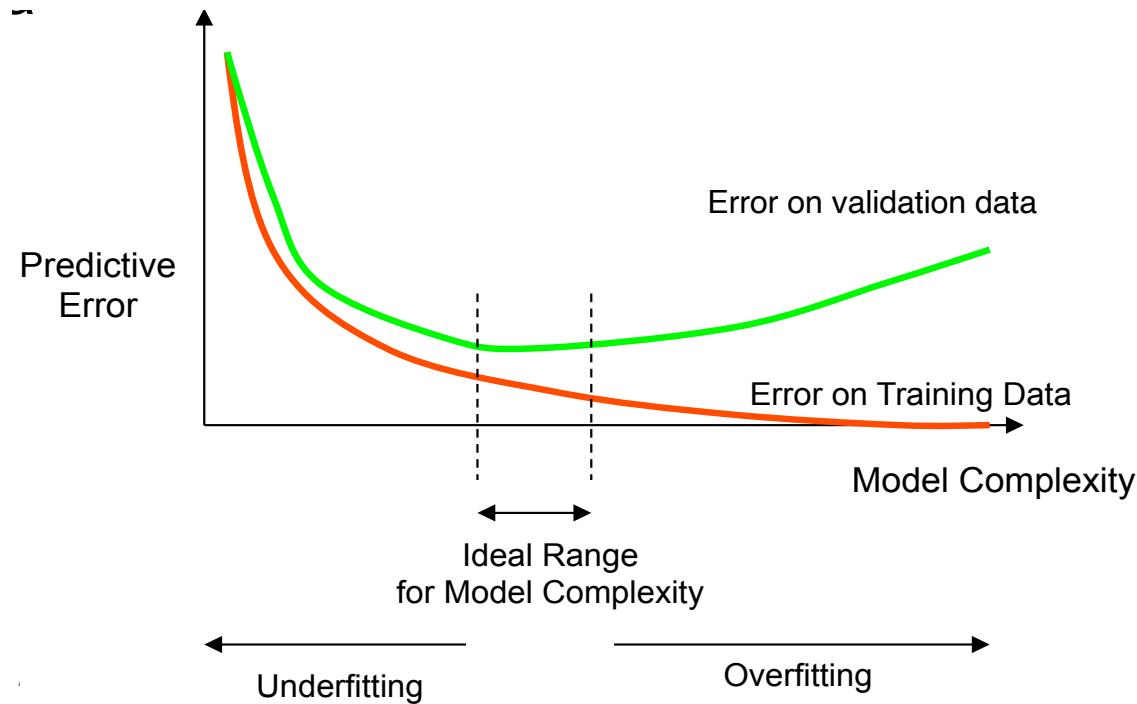


Both the training and validation set *cannot include testing examples*

The validation set cannot be too small. Why??

- Train the model on the training set, while evaluating the model on the held-out validation set
- Choose the model with the best performance on the validation set

- The prediction error on the training and validation datasets



- If the validation error decreases as the model complexity grows, it suggests the model is *under-fitting*
- Otherwise, it implies the model is *overfitting*

Cross-Validation

- Issue with the ordinary validation method

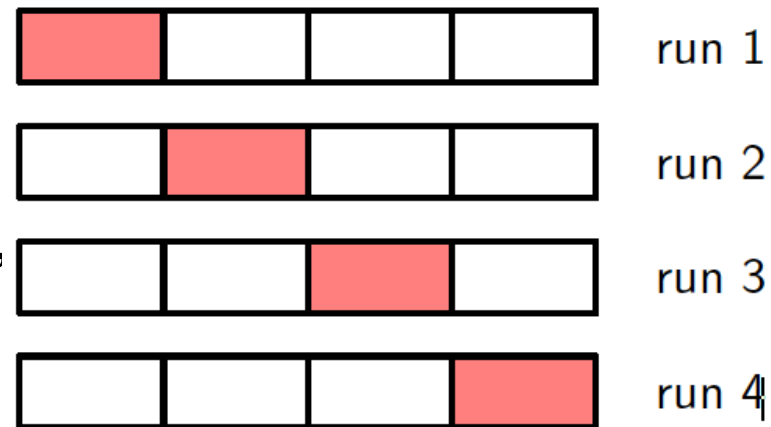
The training data is often scarce. If a large portion is set aside for validation, there are no sufficient data samples used for training

- A compromise solution: **K -fold cross-validation**

➤ Partition the whole training dataset into K subsets equally

➤ Train on the $(K - 1)$ subsets, evaluate on the remaining subset

➤ Repeat the above steps for K times, with each using a different subset for validation



Information Criteria

- Akaike Information Criteria (AIC)

$$AIC = 2M - 2 \log(\mathcal{L})$$

- *M is the number of parameters*
- *\mathcal{L} is the log-likelihood*

- Bayesian Information Criteria (BIC)

$$AIC = M \log N - 2 \log(\mathcal{L})$$

- *N is the number of training data examples*

These criteria *can only be used in the probabilistic models due to the requirement of log-likelihood \mathcal{L}*

Outline

- Model Non-linearization
- Overfitting
- Model Selection
- Regularization

- Imposing some prior preferences on the parameters, in addition to fitting the training data, *e.g.*,

$$\tilde{L}(\mathbf{w}) = L(\mathbf{w}) + \lambda \|\mathbf{w}\|_2^2$$

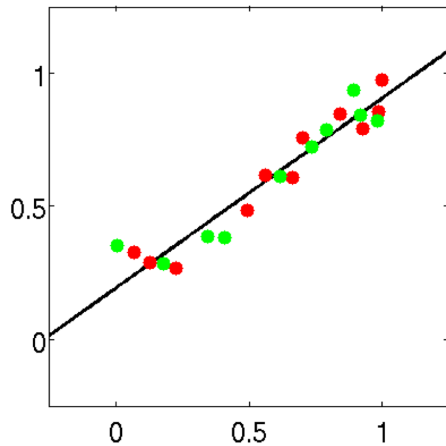
- $L(\mathbf{w})$ is the original regression or classification loss
- $\|\mathbf{w}\|_2 = \left(\sum_{k=1}^K w_k^2\right)^{\frac{1}{2}}$ is the L_2 norm
- λ is the hyper-parameter used to control the influence of $\|\mathbf{w}\|^2$

L_2 regularization

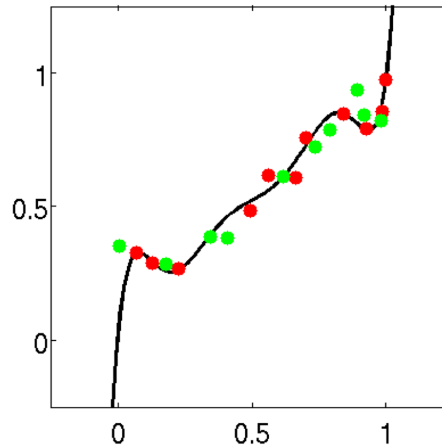
- The properties of L_2 regularization
 - Prone to shrink the model parameters towards zero
 - The larger the λ is, the preference to small values of \mathbf{w} is more strong

- Visualization of the impacts of regularization

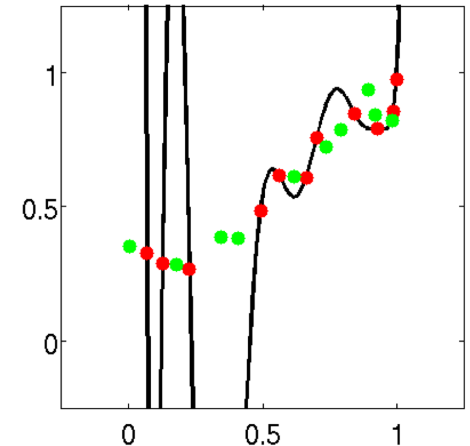
- No regularization



1-order

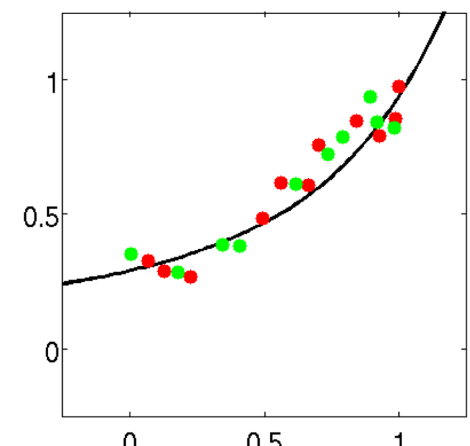
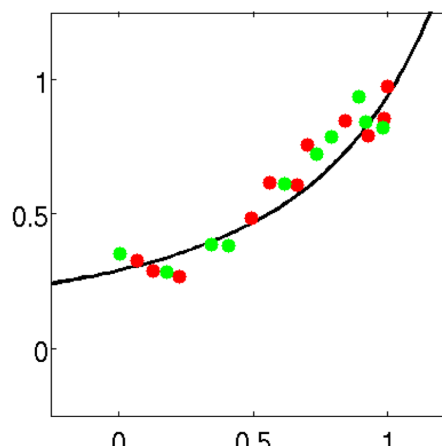
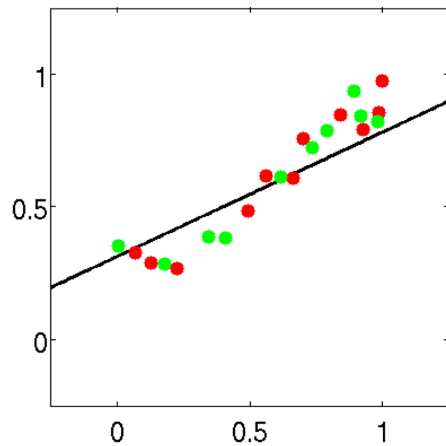


3-order



5-order

- L_2 regularization with $\lambda = 1$

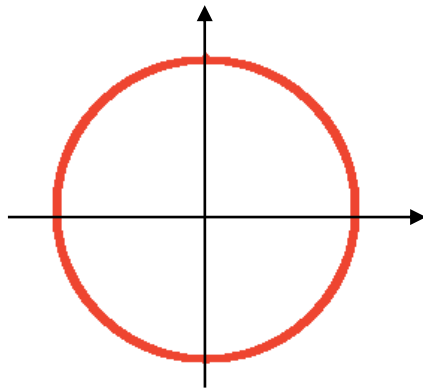


- Another widely used regularization is L_1 regularization

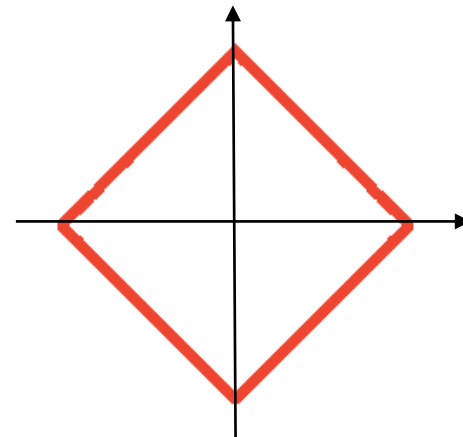
$$\tilde{L}(\mathbf{w}) = L(\mathbf{w}) + \lambda \|\mathbf{w}\|_1$$

where $\|\mathbf{w}\|_1 \triangleq \sum_{k=1}^K |w_k|$ is the L_1 norm

- The contour line of L_2 and L_1 norm



L_2 norm



L_1 norm

- Similar to the L_2 regularization, the L_1 regularization also prefers to have small values for the model parameters
- But the L_1 regularization often **leads to sparse solutions for \mathbf{w}** , that is, many elements in \mathbf{w} are zeros

