6.036 midterm review

Topics covered

- **supervised learning**

labels available

- **unsupervised learning**

no labels available

- **semi-supervised learning**

some labels available

– what algorithms have you learned that correspond to each of these categories?

Topics covered

- **supervised learning**
  - classification
    - online learning algorithms (linear and non-linear)
      - perceptron algorithm
      - passive aggressive algorithm
    - maximum-margin separator (SVM) (linear and non-linear)
    - boosting (non-linear)
    - neural networks (non-linear)
  - regression

- **unsupervised learning**
  - clustering
    - k-means clustering
    - k-medoids clustering

- **semi-supervised learning**
  - collaborative filtering

Topics covered

- **supervised learning**
  - classification ← binary
    - online learning algorithms (linear and non-linear)
      - perceptron algorithm
      - passive aggressive algorithm
    - maximum-margin separator (SVM) (linear and non-linear)
    - boosting (non-linear) ← AdaBoost
    - neural networks (non-linear)
  - regression ← ridge regression

kernelized

- **unsupervised learning**
  - clustering
    - k-means clustering
    - k-medoids clustering

- **semi-supervised learning**
  - collaborative filtering

- you have seen binary classification -> how do you think we can extend to multi-class classification? (many binary classifiers)
- ensemble methods combine the opinions of multiple learners; boosting is one way of doing this; AdaBoost is a particular implementation of boosting
- you have seen ridge regression - can also have lasso regression (difference in penalty function - i.e. regularization)
- semi-supervised learning can also occur with clustering, for instance, with some exemplars in the clusters labeled; this information can be used for classification (project 2!)

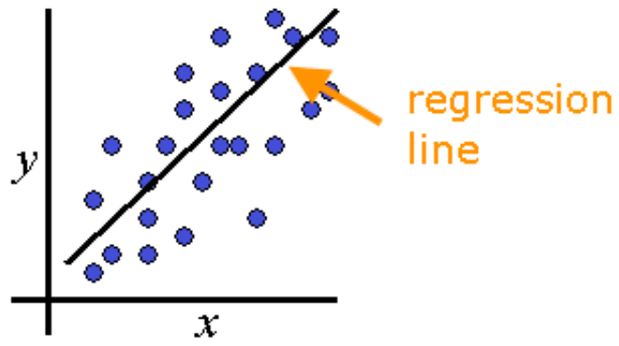
Common themes

- objective function (minimize error)
  - loss functions: euclidean, exponential
- iterative algorithms (e.g. mistake-driven learning)
  - gradient descent
- generalization and regularization
  - training error vs test error
  - overfitting
  - cross-validation
- feature representations
  - nonlinear feature mapping (e.g. kernelized methods)
  - dimensionality reduction (e.g. PCA)
- similarity/distance measures
  - cosine, euclidean, manhattan
  - kernels

– exercise: in which contexts (for which algorithms, problems) have you seen these themes come up?

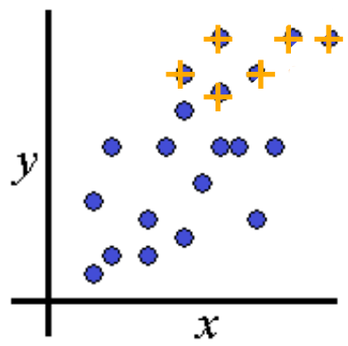
Relationship b/w regression and classification

Regression setting:
given data point (vector) x, make a prediction: $x \cdot \theta + \theta_o$



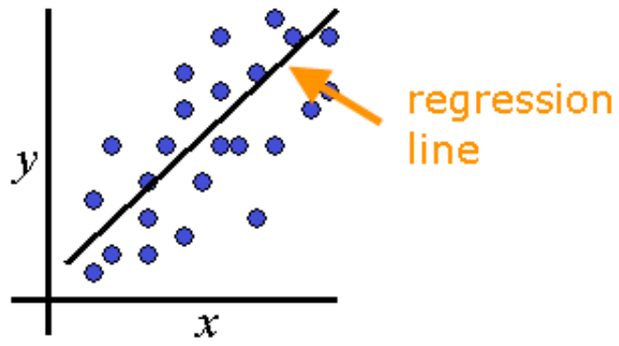
Classification setting:
given data point (vector) x, make a prediction: $sign(x \cdot \theta + \theta_o)$

binary classification



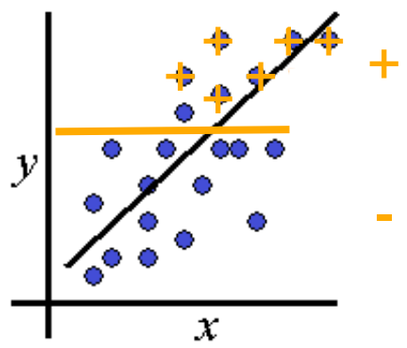
Relationship b/w regression and classification

Regression setting:
given data point (vector) x, make a prediction: $x \cdot \theta + \theta_o$



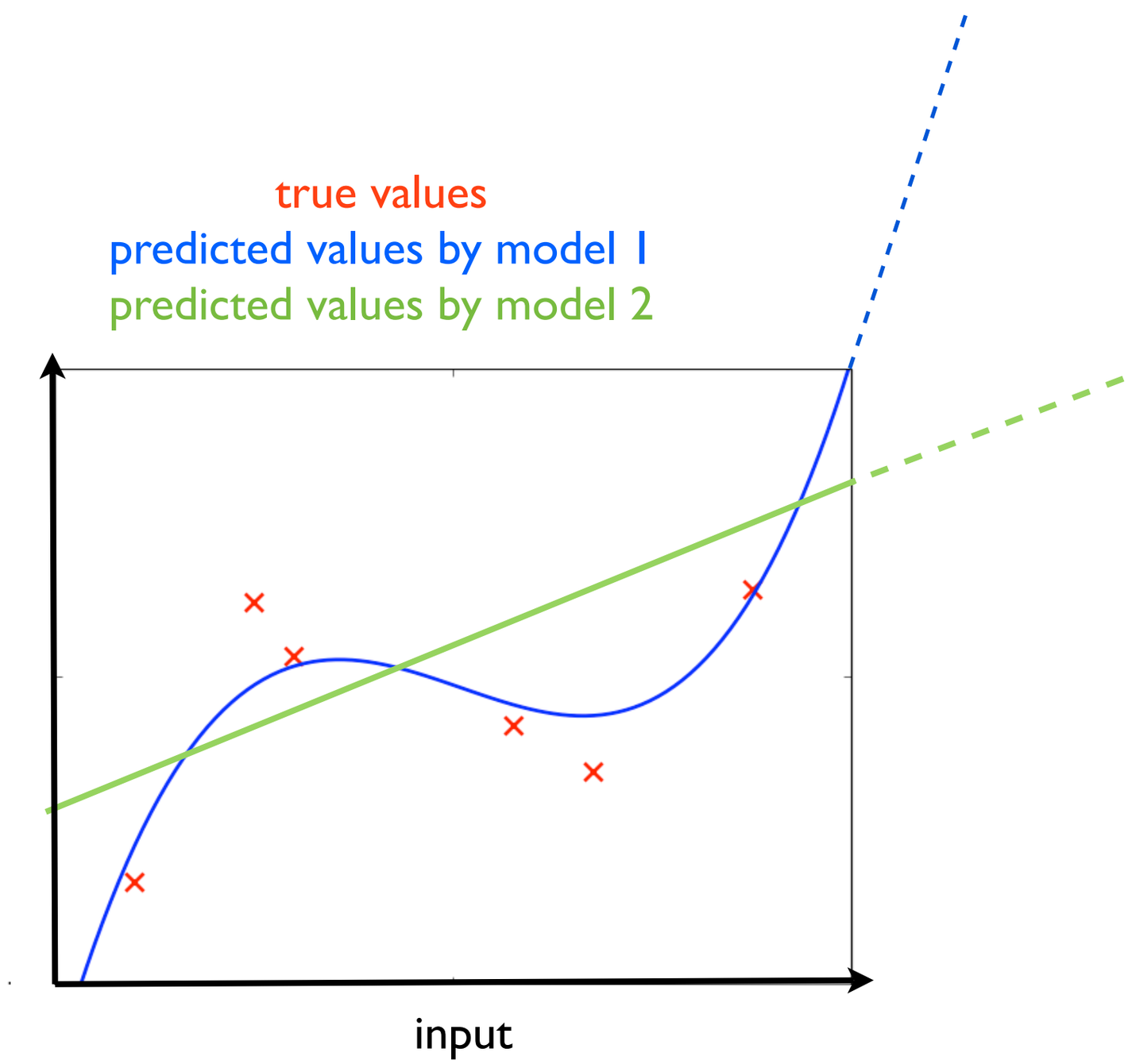
Classification setting:
given data point (vector) x, make a prediction: $sign(x \cdot \theta + \theta_o)$

binary classification



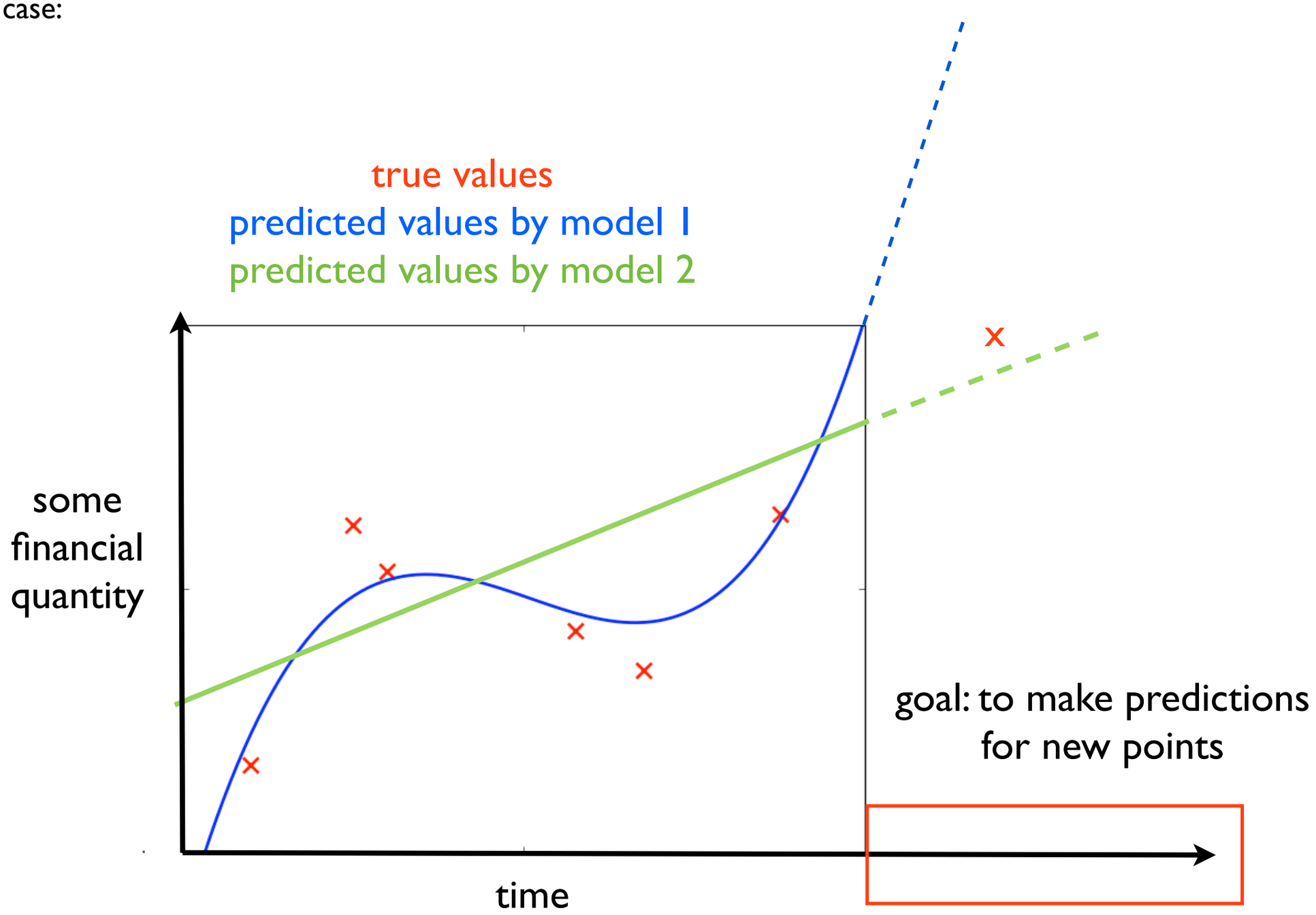
Effect of regularization

consider generalized linear regression on 1D input:

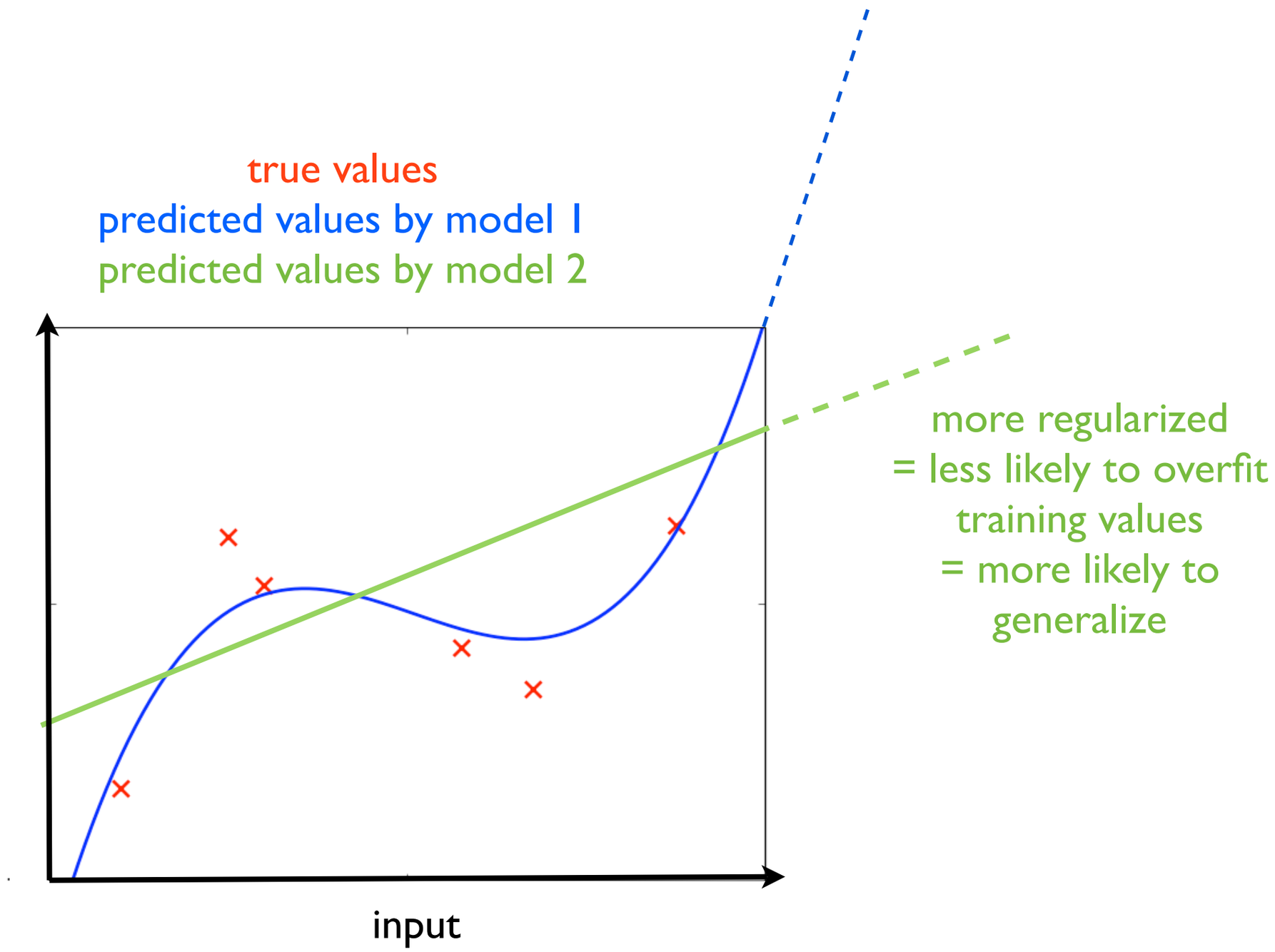


Effect of regularization

real-world use case:



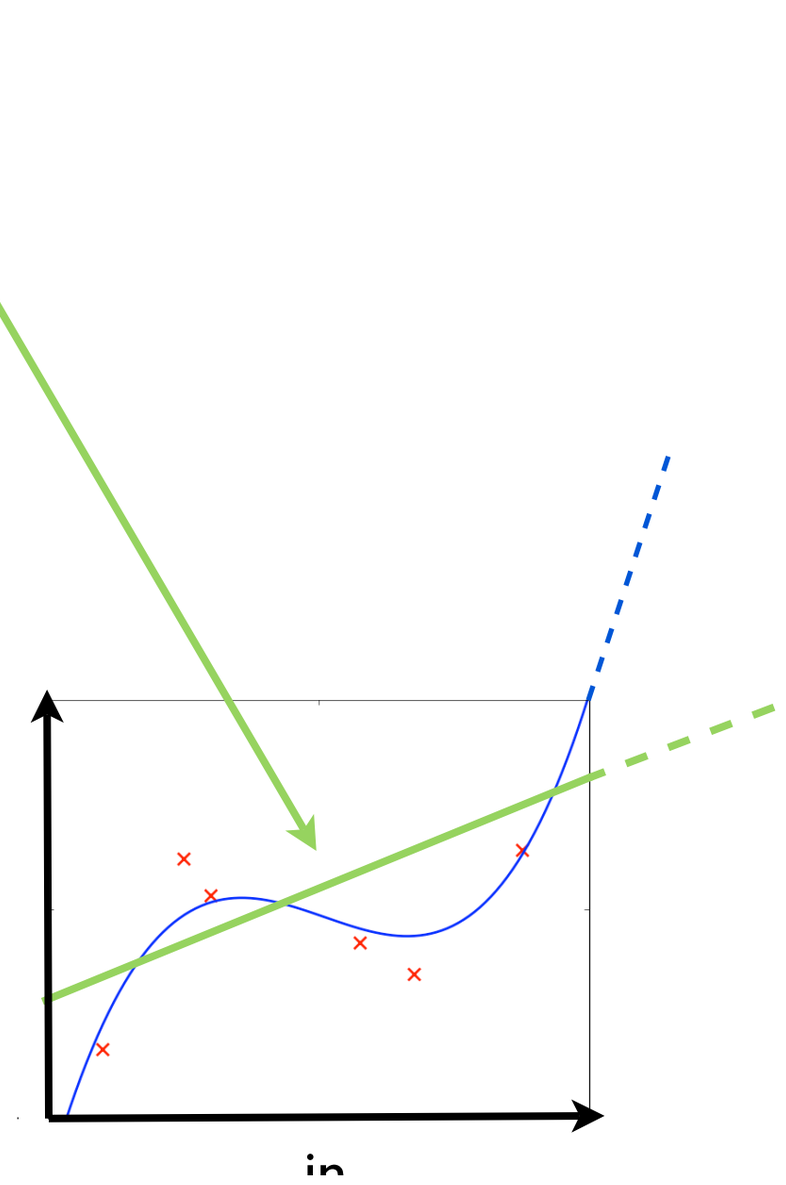
Effect of regularization



Generalized linear regression and kernels

$$y = \phi(x) \cdot \theta$$

$$y = x\theta$$



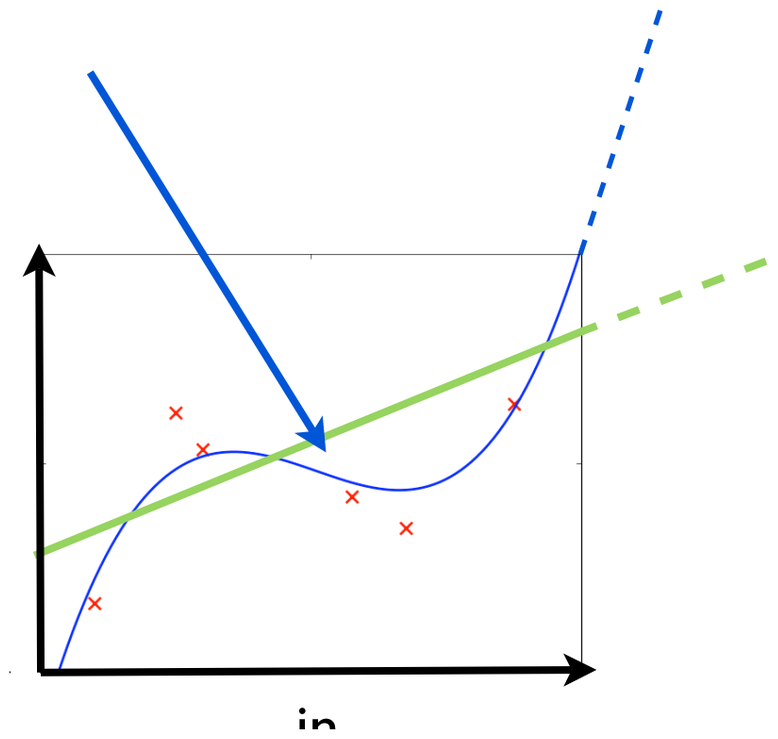
Generalized linear regression and kernels

$$\phi(x) = (x^3, \sqrt{3}x^2, \sqrt{3}x, 1)$$

$$y = \phi(x) \cdot \theta$$

$$y = (x^3, \sqrt{3}x^2, \sqrt{3}x, 1) \cdot (\theta_1, \theta_2, \theta_3, \theta_4)$$

$$y = x^3\theta_1 + \sqrt{3}x^2\theta_2 + \sqrt{3}x\theta_3 + \theta_4$$



in

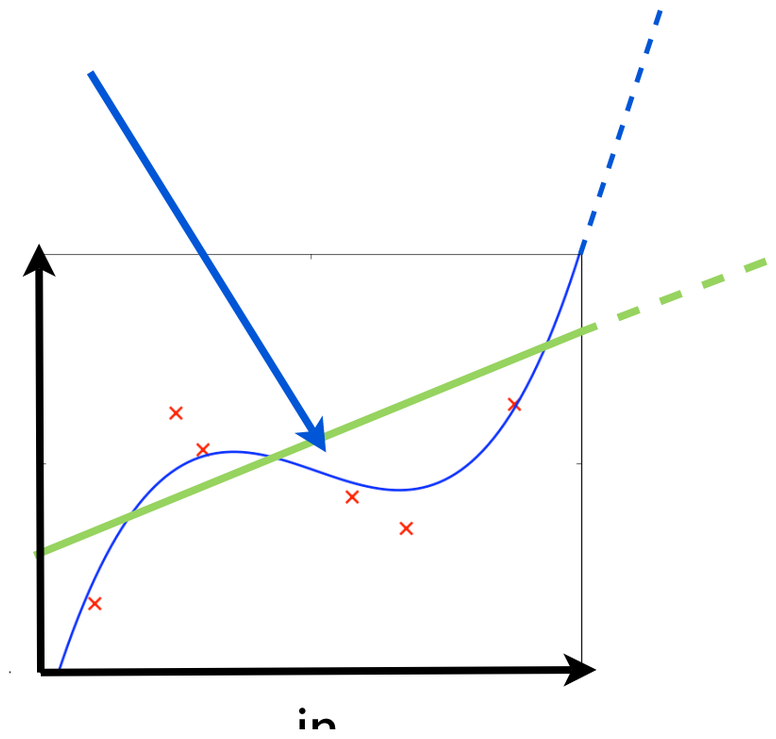
Generalized linear regression and kernels

$$\phi(x) = (x^3, \sqrt{3}x^2, \sqrt{3}x, 1)$$

$$y = \phi(x) \cdot \theta$$

$$y = (x^3, \sqrt{3}x^2, \sqrt{3}x, 1) \cdot (\theta_1, \theta_2, \theta_3, \theta_4)$$

$$y = x^3\theta_1 + \sqrt{3}x^2\theta_2 + \sqrt{3}x\theta_3 + \theta_4$$



$$K(x, z) = (x \cdot z + 1)^3 = x^3z^3 + 3x^2z^2 + 3xz + 1$$

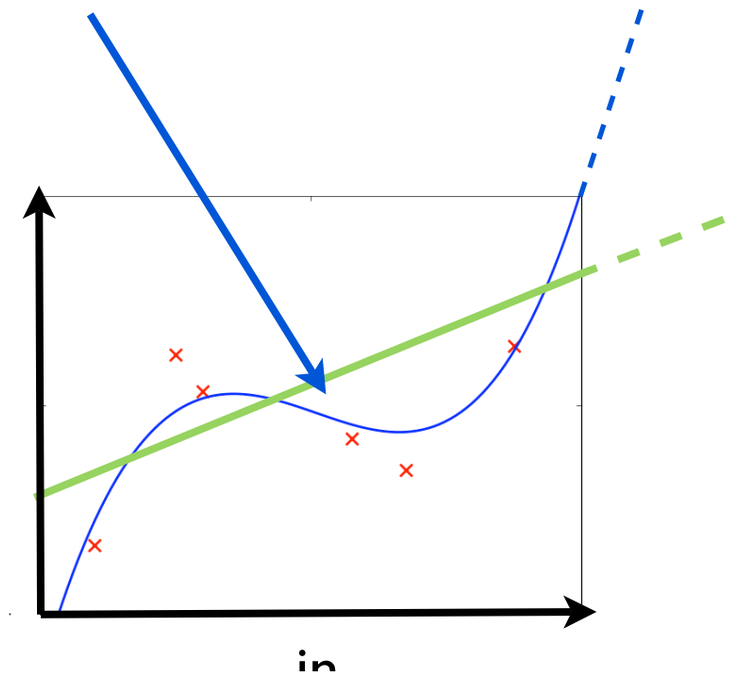
Generalized linear regression and kernels

$$\phi(x) = (x^3, \sqrt{3}x^2, \sqrt{3}x, 1)$$

$$y = \phi(x) \cdot \theta$$

$$y = (x^3, \sqrt{3}x^2, \sqrt{3}x, 1) \cdot (\theta_1, \theta_2, \theta_3, \theta_4)$$

$$y = x^3\theta_1 + \sqrt{3}x^2\theta_2 + \sqrt{3}x\theta_3 + \theta_4$$



Note: function is linear in new feature space but projects into a polynomial in original feature space

$$K(x, z) = (x \cdot z + 1)^3 = x^3z^3 + 3x^2z^2 + 3xz + 1$$

Generalized linear regression and kernels

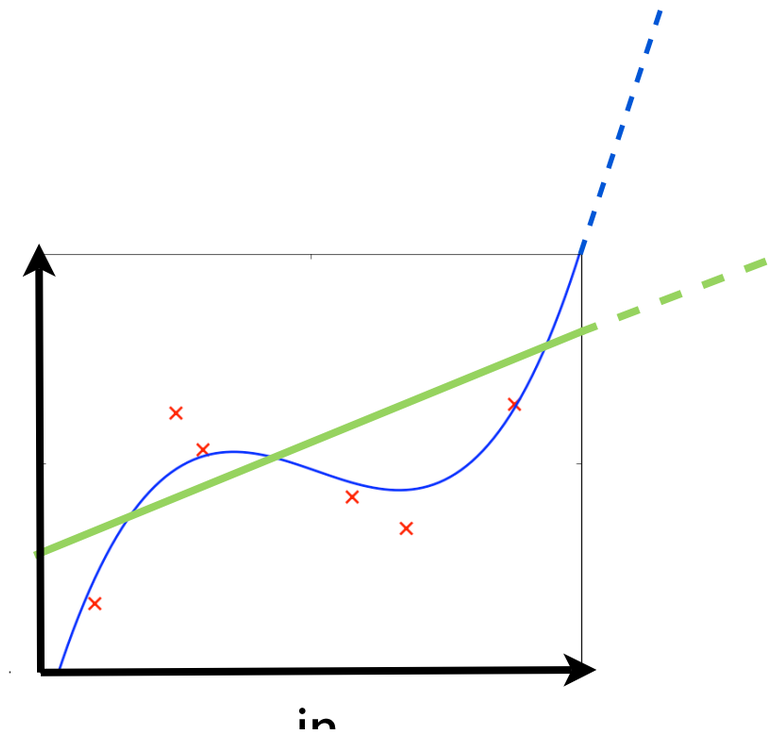
$$\phi(x) = (x^3, \sqrt{3}x^2, \sqrt{3}x, 1)$$

$$y = \phi(x) \cdot \theta$$

what happens when we regularize Θ?

$$y = (x^3, \sqrt{3}x^2, \sqrt{3}x, 1) \cdot (\theta_1, \theta_2, \theta_3, \theta_4)$$

$$y = x^3\theta_1 + \sqrt{3}x^2\theta_2 + \sqrt{3}x\theta_3 + \theta_4$$



Generalized linear regression and kernels

$$\phi(x) = (x^3, \sqrt{3}x^2, \sqrt{3}x, 1)$$

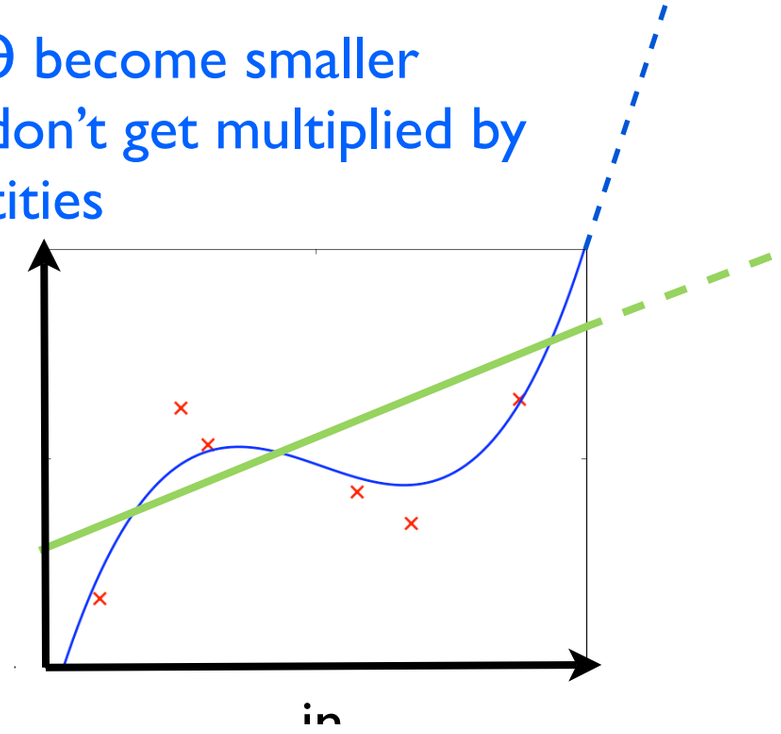
$$y = \phi(x) \cdot \theta$$

what happens when we regularize Θ?

$$y = (x^3, \sqrt{3}x^2, \sqrt{3}x, 1) \cdot (\theta_1, \theta_2, \theta_3, \theta_4)$$

$$y = x^3\theta_1 + \sqrt{3}x^2\theta_2 + \sqrt{3}x\theta_3 + \theta_4$$

all the components of Θ become smaller
so higher-order terms of x don't get multiplied by
large quantities



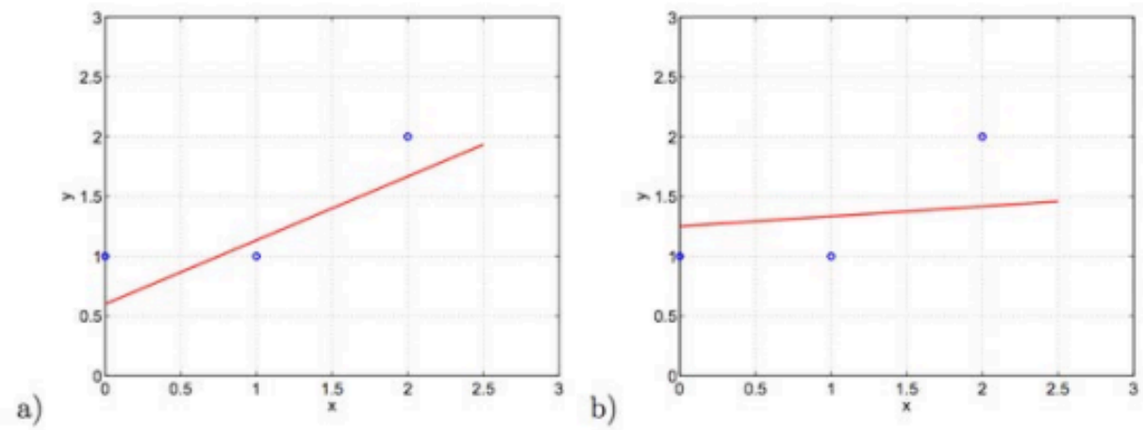
Regularization example (for linear regression)

$$\sum_{t=1}^{3}(y_t - \theta x_t - \theta_0)^2 + \lambda\theta^2, \text{where } \lambda = 1 \tag{1}$$

$$\sum_{t=1}^{3}(y_t - \theta x_t - \theta_0)^2 + \lambda\theta^2, \text{where } \lambda = 10 \tag{2}$$

$$\sum_{t=1}^{3}(y_t - \theta x_t - \theta_0)^2 + \lambda(\theta^2 + \theta_0^2), \text{where } \lambda = 1 \tag{3}$$

$$\sum_{t=1}^{3}(y_t - \theta x_t - \theta_0)^2 + \lambda(\theta^2 + \theta_0^2), \text{where } \lambda = 10 \tag{4}$$



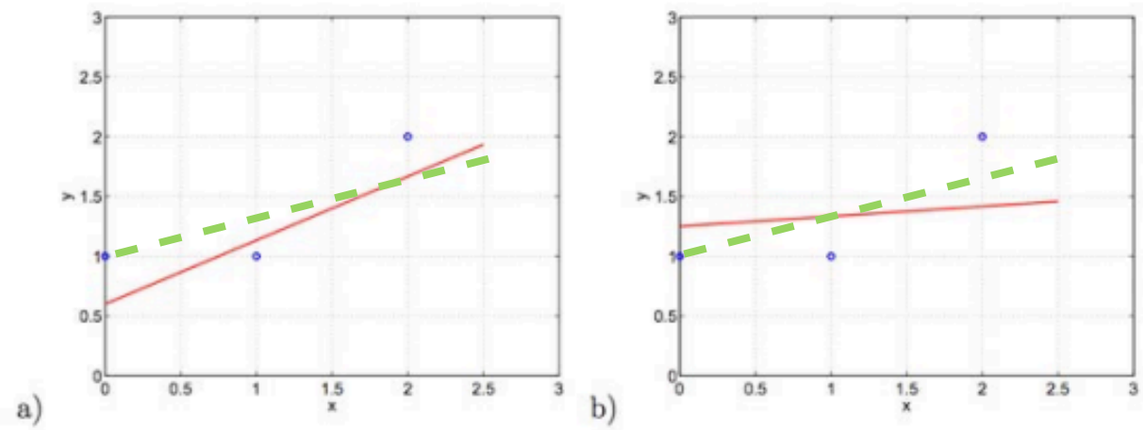
Regularization example (for linear regression)

$$\sum_{t=1}^{3}(y_t - \theta x_t - \theta_0)^2 + \lambda\theta^2, \text{where } \lambda = 1 \tag{1}$$

$$\sum_{t=1}^{3}(y_t - \theta x_t - \theta_0)^2 + \lambda\theta^2, \text{where } \lambda = 10 \tag{2}$$

$$\sum_{t=1}^{3}(y_t - \theta x_t - \theta_0)^2 + \lambda(\theta^2 + \theta_0^2), \text{where } \lambda = 1 \tag{3}$$

$$\sum_{t=1}^{3}(y_t - \theta x_t - \theta_0)^2 + \lambda(\theta^2 + \theta_0^2), \text{where } \lambda = 10 \tag{4}$$



Regularization example (for linear regression)

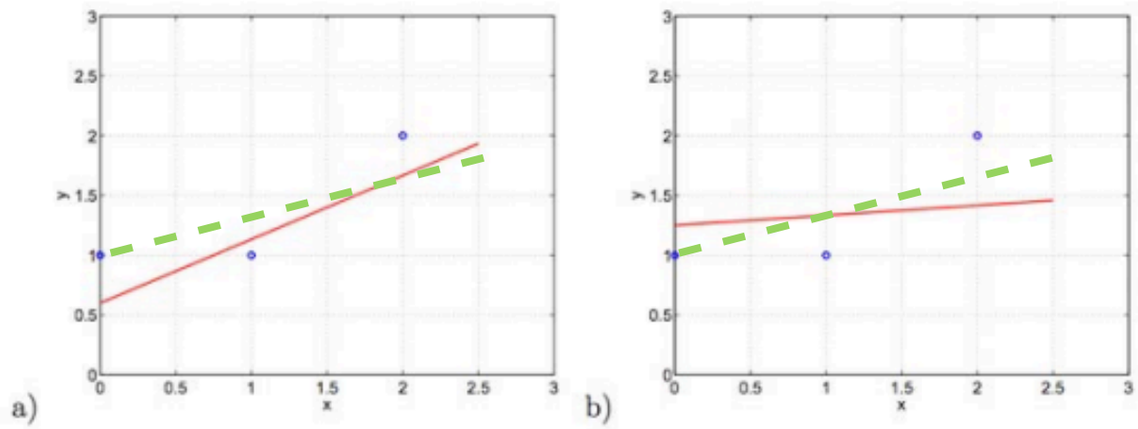
how to think about these problems:
- what is the best fit?
- what has been modified to move away from best fit?
- would changing the slope yield a better fit? if yes, perhaps slope was overly regularized
- would changing the offset yield a better fit? if yes, perhaps offset was overly regularized

$$\sum_{t=1}^{3}(y_t - \theta x_t - \theta_0)^2 + \lambda\theta^2, \text{where } \lambda = 1 \quad (1)$$

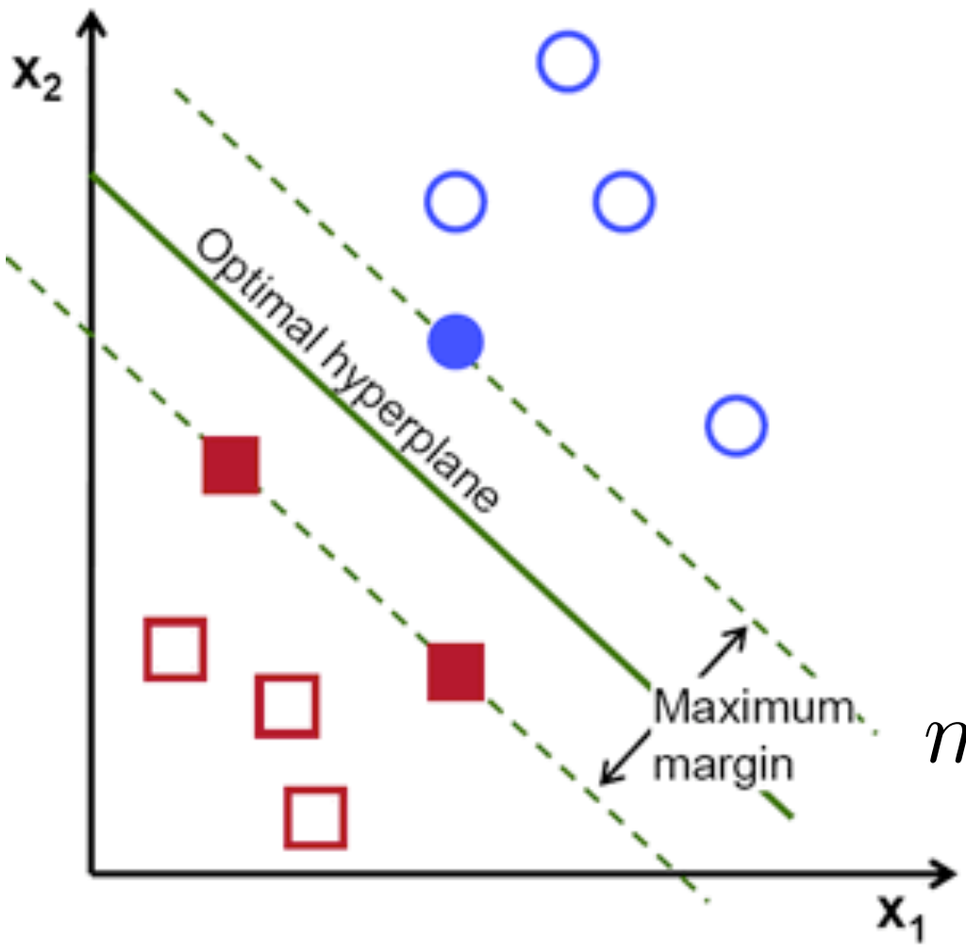
$$\sum_{t=1}^{3}(y_t - \theta x_t - \theta_0)^2 + \lambda\theta^2, \text{where } \lambda = 10 \quad (2)$$

$$\sum_{t=1}^{3}(y_t - \theta x_t - \theta_0)^2 + \lambda(\theta^2 + \theta_0^2), \text{where } \lambda = 1 \quad (3)$$

$$\sum_{t=1}^{3}(y_t - \theta x_t - \theta_0)^2 + \lambda(\theta^2 + \theta_0^2), \text{where } \lambda = 10 \quad (4)$$



Max-margin separator:
Support Vector Machine (SVM)

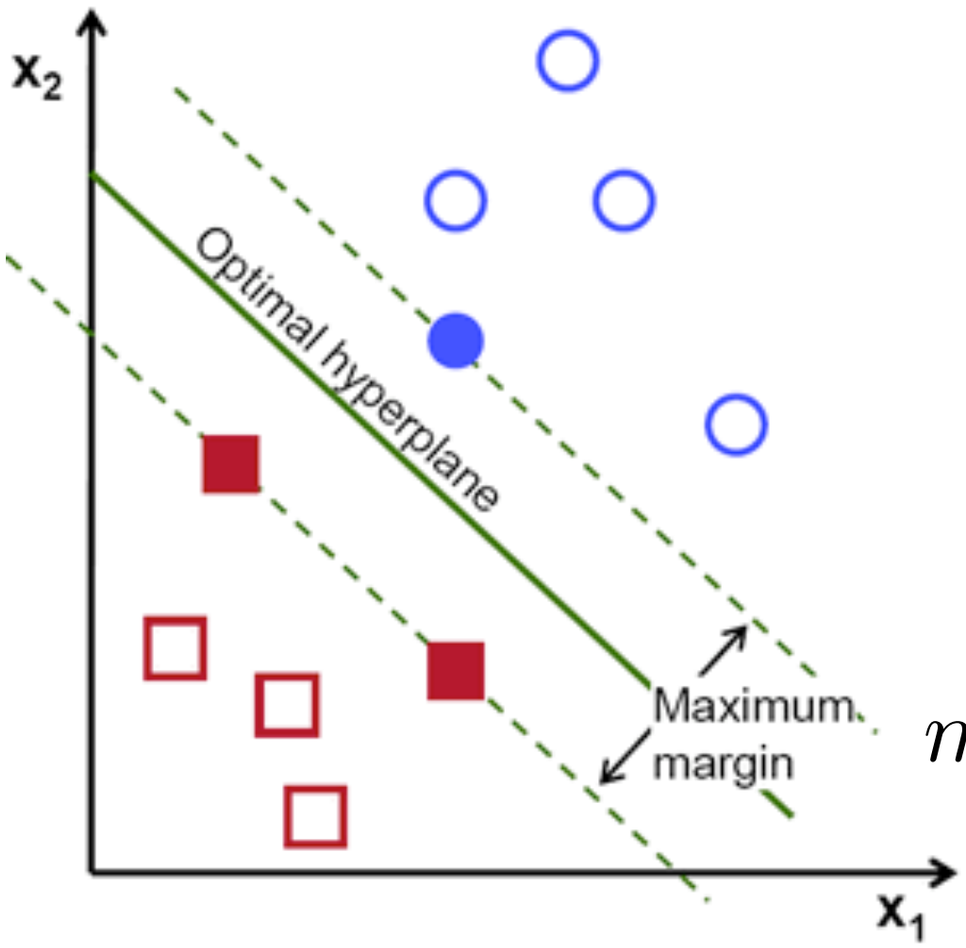


$$max \frac{1}{\|\theta\|}$$

$$min \frac{1}{2}\|\theta\|^2 \text{ subject to } y^{(i)}(\theta \cdot x^{(i)}) \geq 1$$

Max-margin separator: Support Vector Machine (SVM)

Note that the decision boundary is fully specified by the training examples

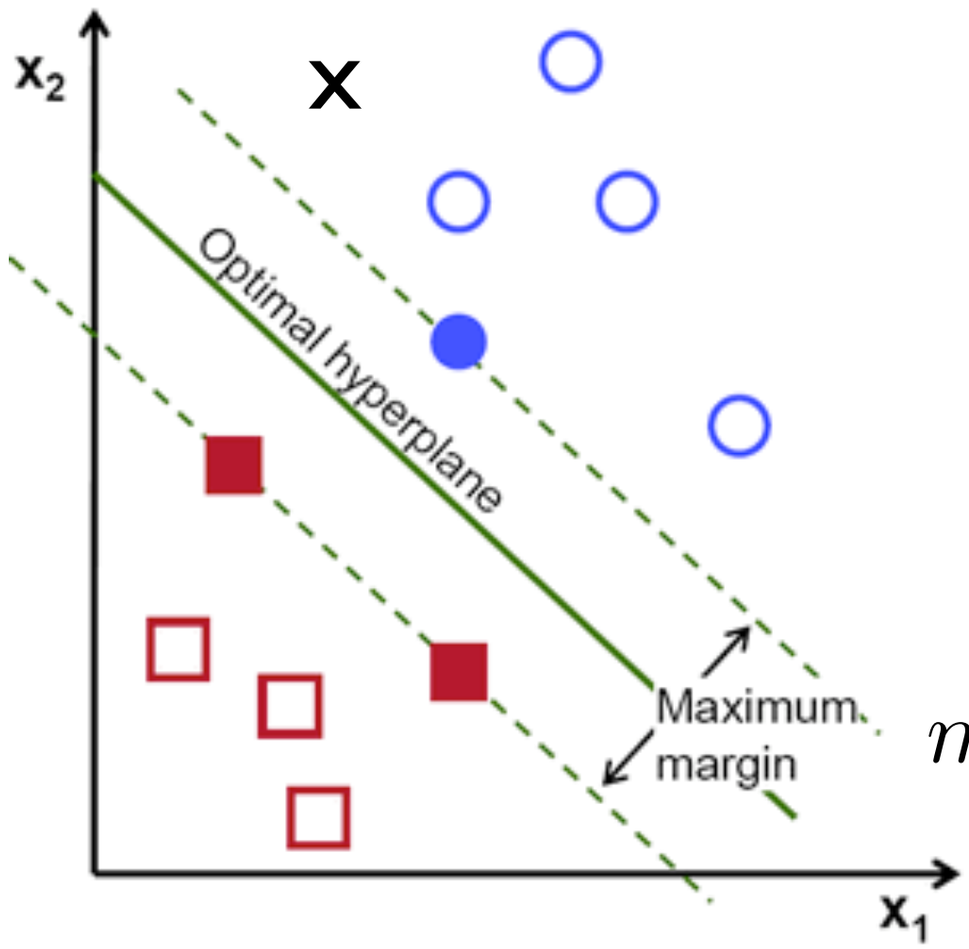


$$max \frac{1}{\|\theta\|}$$

$$min \frac{1}{2}\|\theta\|^2 \text{ subject to } y^{(i)}(\theta \cdot x^{(i)}) \geq 1$$

Max-margin separator: Support Vector Machine (SVM)

prediction: $sign(\theta \cdot x^{(new)})$



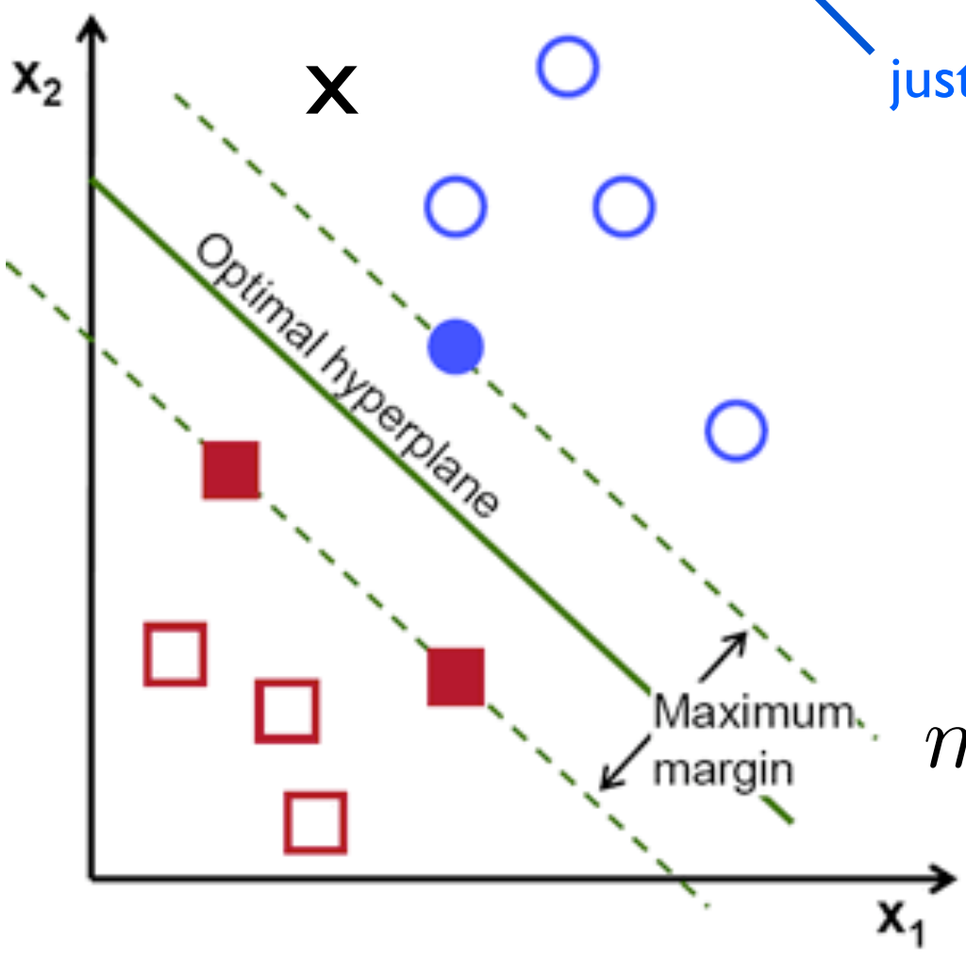
$max \frac{1}{\|\theta\|}$

We want to make a prediction for a new data point. To do this, we use our computed decision boundary (specified by theta). But our decision boundary is fully specified by our training examples, so why not directly use them?

Max-margin separator:
Support Vector Machine (SVM)

prediction: $sign(\sum_i \alpha^{(i)} y^{(i)} K(x^{(i)}, x^{(new)}))$

just a similarity function



$max \frac{1}{\|\theta\|}$

we can! this is an alternative (dual) formulation of the SVM problem that does not rely on the explicit computation of the decision boundary

Q: does our decision boundary depend on ALL the points, or only a subset of them?
A: only a subset of points - our support vectors are required!

Studying tips

- go through all the derivations and see if you can translate them to English sentences (ask yourself: what do we do at this step and why?)
- consider how different parts of a function depend on and influence each other
- modify the original formulations and see what happens when you vary different components (regularization, offset, loss function)
- consider how the algorithms you have learned behave under different situations (many vs few training examples, linearly (in)separable cases, etc.)
- in other words: GENERALIZE from the examples you saw in class