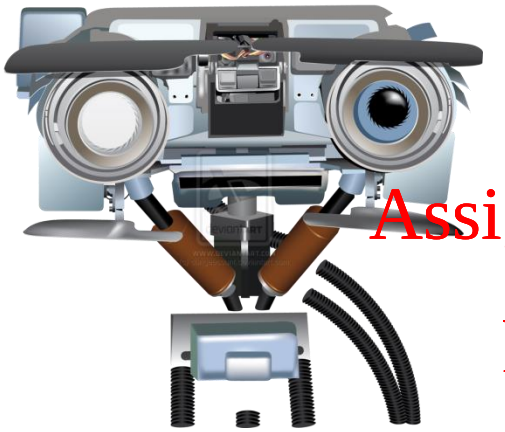
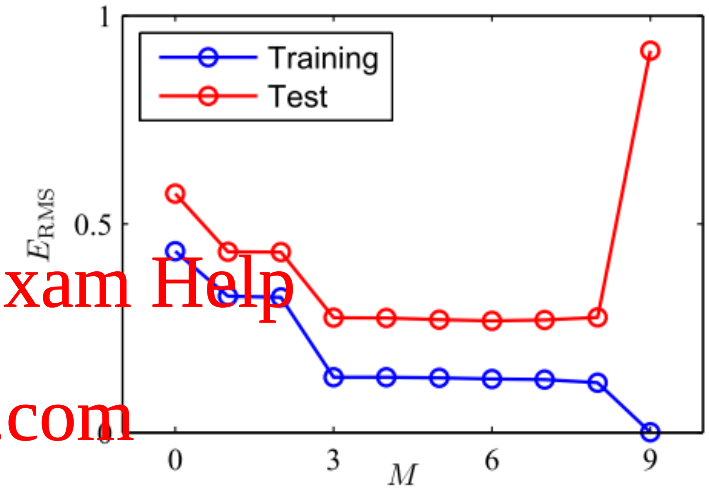


Supervised Learning III

Classification, Regularization

Detecting overfitting

Plot model complexity versus objective function on test/train data



As model becomes more complex, performance on training keeps improving while on test data it increases

Horizontal axis: measure of model complexity
In this example, we use the maximum order of the polynomial basis functions.

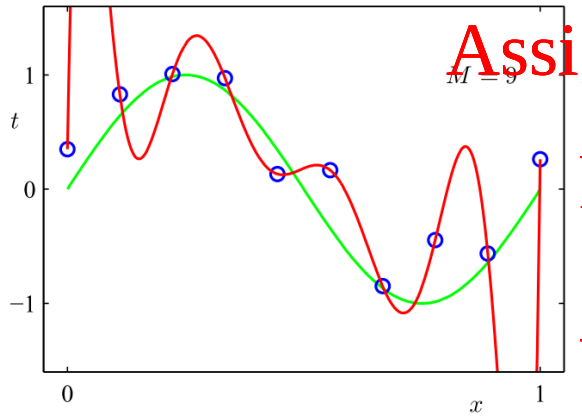
Vertical axis: For regression, it would be SSE or mean SE (MSE)
For classification, the vertical axis would be classification error rate or cross-entropy error function

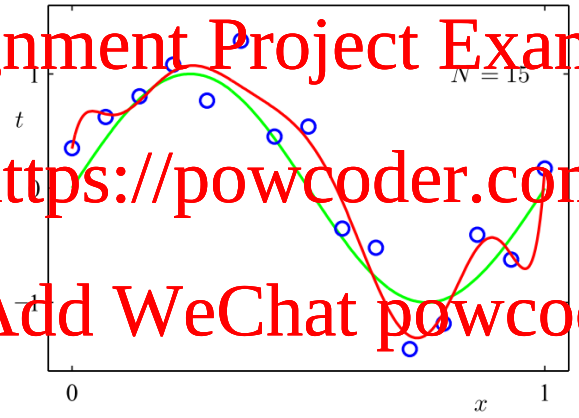
Overcoming overfitting

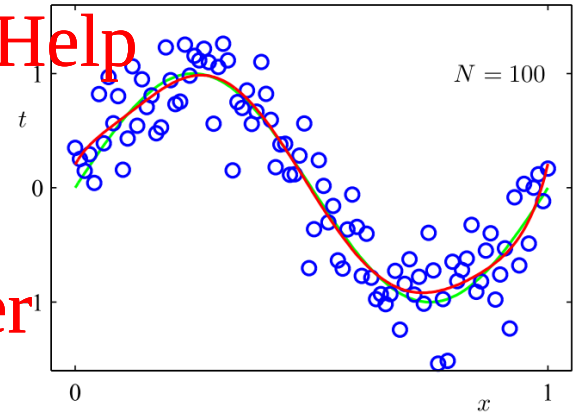
- Basic ideas
  - Use more training data
  - Regularization methods
  - Cross-validation

Solution: use more data

M=9, increase N



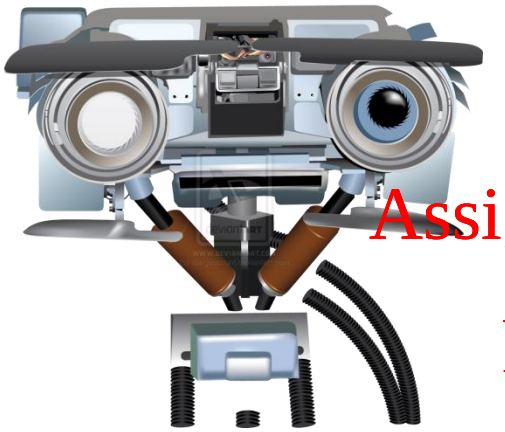




What if we do not have a lot of data?

Overcoming overfitting

- Basic ideas
  - Use more training data
  - Regularization methods
  - Cross-validation

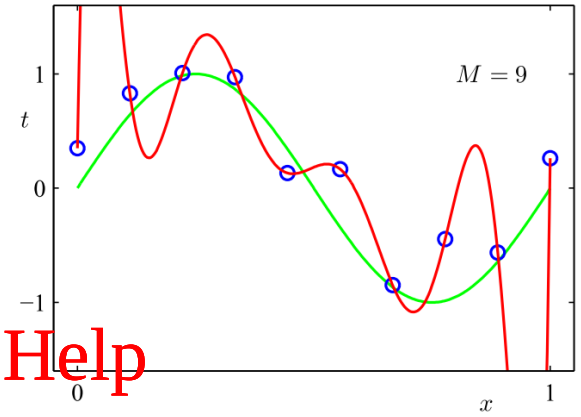


Supervised Learning III

Regularization

Solution: Regularization

- Use regularization:
  - Add $\lambda\|\theta\|_2{}^2$ term to SSE cost function
  - "L-2" norm squared, ie sum of sq. elements $\sum\theta_j^2$
  - Penalizes large θ
  - λ controls amount of regularization



M =9

0.35

232.37

-5321.83

48568.31

-231639.30

640042.26

-1061800.52

1042400.18

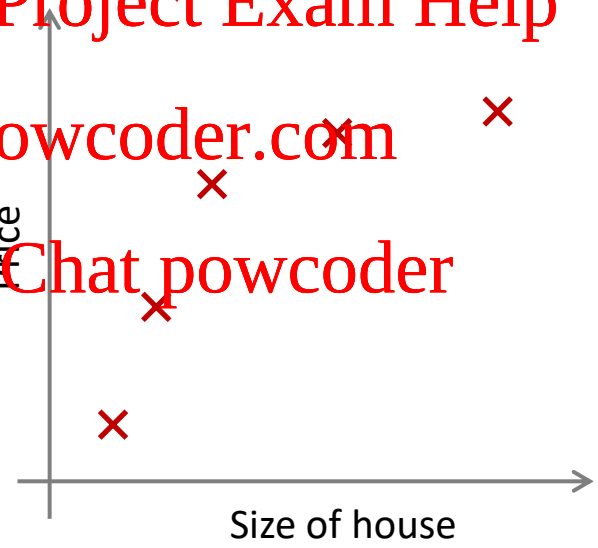
-557682.99

125201.43

Regularized Linear Regression

$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 + \lambda \sum_{j=1}^{n}\theta_j^2\right]$$

$$\min_\theta J(\theta)$$



Gradient descent for Linear Regression

Repeat $\{$

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_0^{(i)}$$

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)} \quad (j = 1, 2, 3, \ldots, n)$$

$\}$

replace with

$$\theta_j := \theta_j (1 - \alpha \frac{\lambda}{m}) - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

Regularized Normal Equation

Suppose $m \leq n$,

(#examples) (#features)

$$\theta = (X^T X)^{-1} X^T y$$ Non-invertible/singular

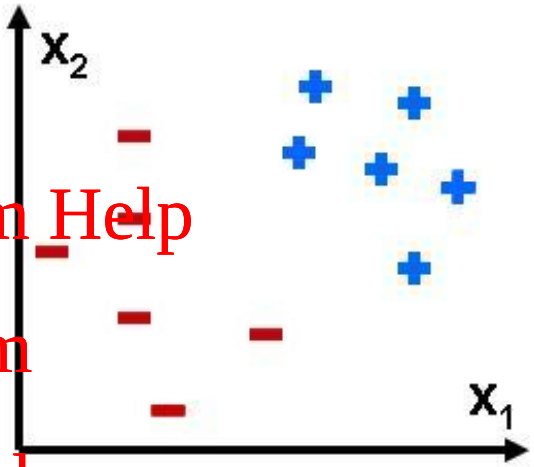
If $\lambda > 0$,

$$\theta = \left(X^T X + \lambda \begin{bmatrix} 0 & & & \\ & 1 & & \\ & & 1 & \\ & & & \ddots \\ & & & & 1 \end{bmatrix} \right)^{-1} X^T y$$

Regularized Logistic Regression

Hypothesis:

$$h_\theta(x) = g(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}}$$



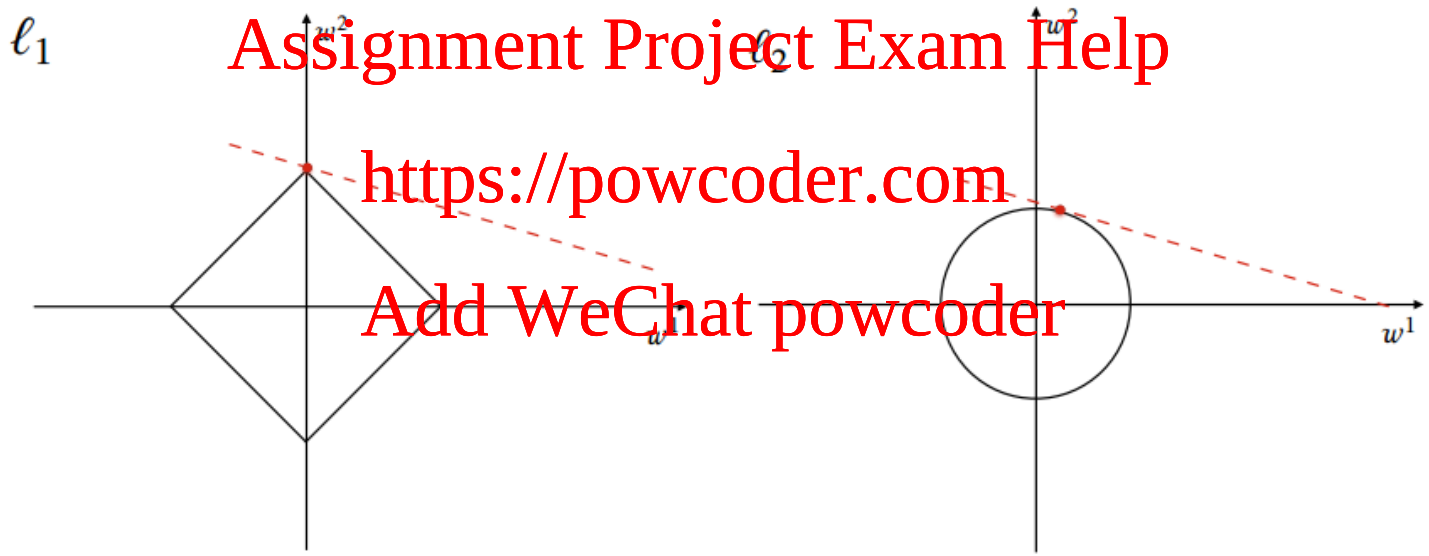
Cost Function:

$$J(\theta) = -\frac{1}{m}[\sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1 - y^{(i)}) \log (1 - h_\theta(x^{(i)}))]$$

Goal: minimize cost $\min_\theta J(\theta)$

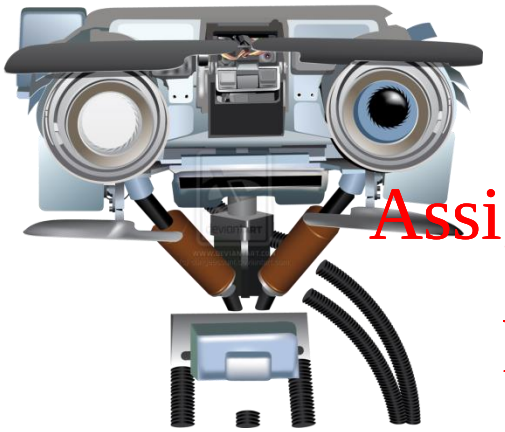
Many types of Regularization

- Most common are ℓ_1 and ℓ_2



ℓ_1 often used to create sparsity

Image credit: https://en.wikipedia.org/wiki/Regularization_(mathematics)



Supervised Learning III

Bias-Variance

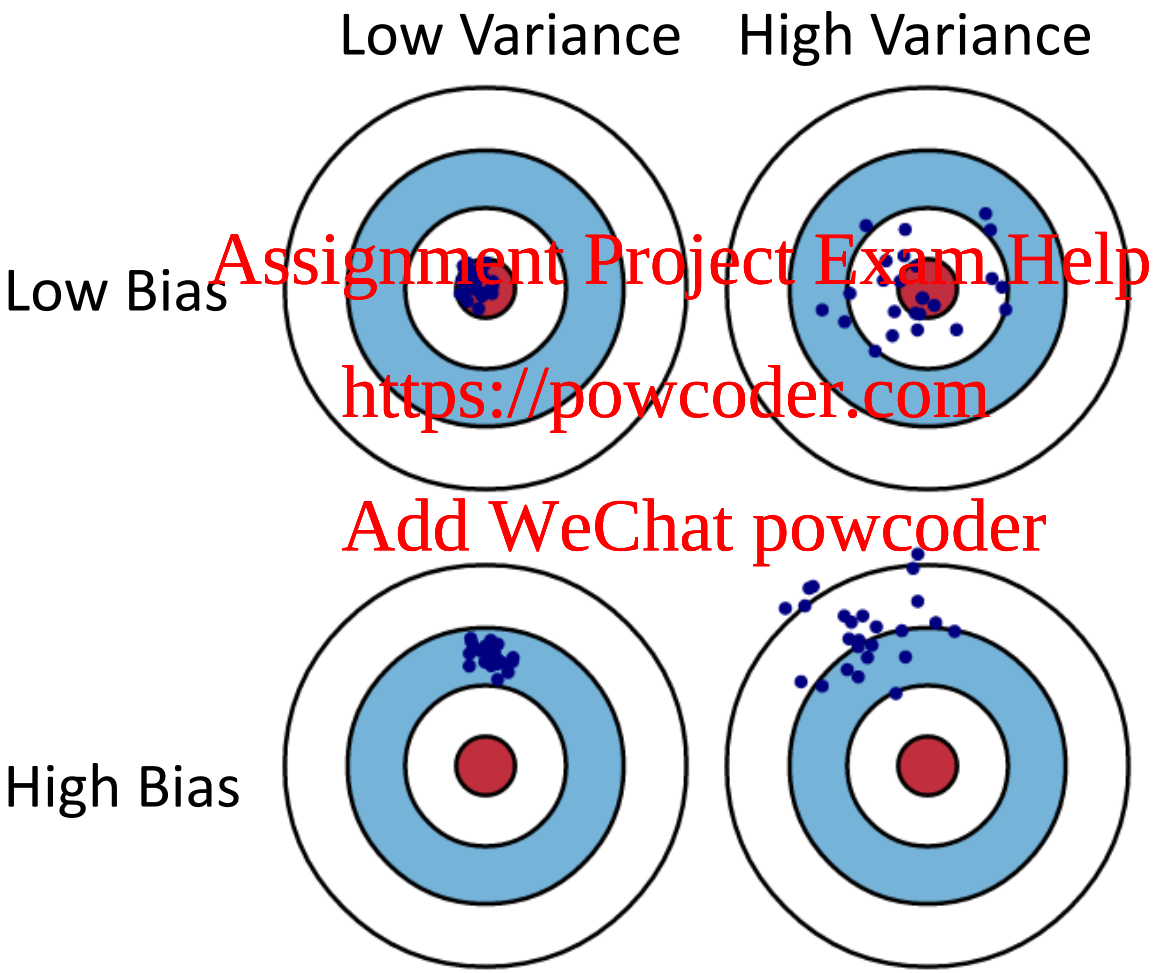
Bias vs Variance

- Understanding how different sources of error lead to bias and variance helps us improve model fitting

- **Error due to Bias**: The error due to bias is taken as the difference between the expected (or average) prediction of our model and the correct value which we are trying to predict (imagine you could repeat the whole model fitting process on many datasets)

- **Error due to Variance**: The variance is how much the predictions for a given point vary between different realizations of the model.

Graphical Illustration



http://scott.fortmann-roe.com/docs/BiasVariance.html

The Bias-Variance Trade-off

There is a trade-off between bias and variance:

- Less complex models (fewer parameters) have high bias and hence low variance
- More complex models (more parameters) have low bias and hence high variance
- Optimal model will have a balance

Which is worse?

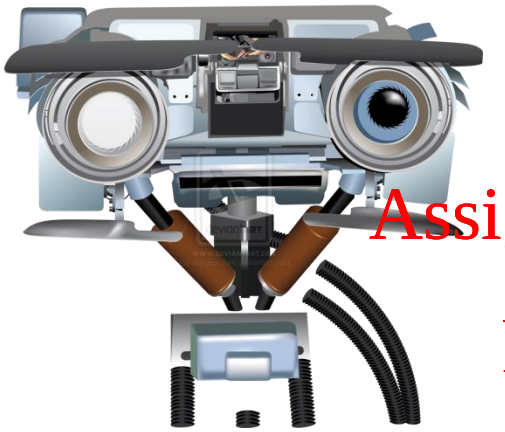
- A gut feeling many people have is that they should minimize bias even at the expense of variance

- This is mistaken logic. It is true that a high variance and low bias model can preform well in some sort of long-run average sense. However, in practice modelers are always dealing with a single realization of the data set

- In these cases, long run averages are irrelevant, bias and variance are equally important, and one should not be improved at an excessive expense to the other.

How to deal with bias/variance

- Can deal with variance by
  - Bagging, e.g. Random Forest
  - Bagging trains multiple models on random subsamples of the data, averages their prediction
- Can deal with high bias by
  - Decreasing regularization/increasing complexity of model
  - Also known as *model selection*



Supervised Learning III

Model selection and training/validation/test sets

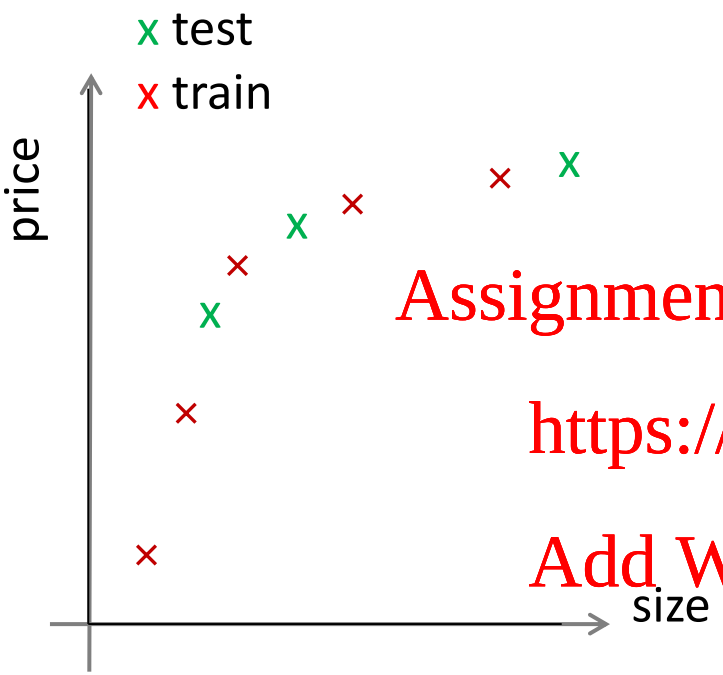
Not performing well on training data

Not generalizing well from training to unseen data

(underfit)

(overfit)

Model selection



$$h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

Hyperparameters (e.g., degree of polynomial, regularization weight, learning rate) must be selected prior to training.

How to choose them?

Try several values, choose one with the lowest test error?

Problem: test error is likely an overly optimistic estimate of generalization error because we “cheat” by fitting the hyperparameter to the actual test examples.

Train/Validation/Test Sets

	Size	Price
train	2104	400
	1600	330
	2400	369
	1416	232
	3000	540
validation	1985	300
	1534	315
	1427	199
test	1380	212
	1494	243

Solution: split data into three sets.

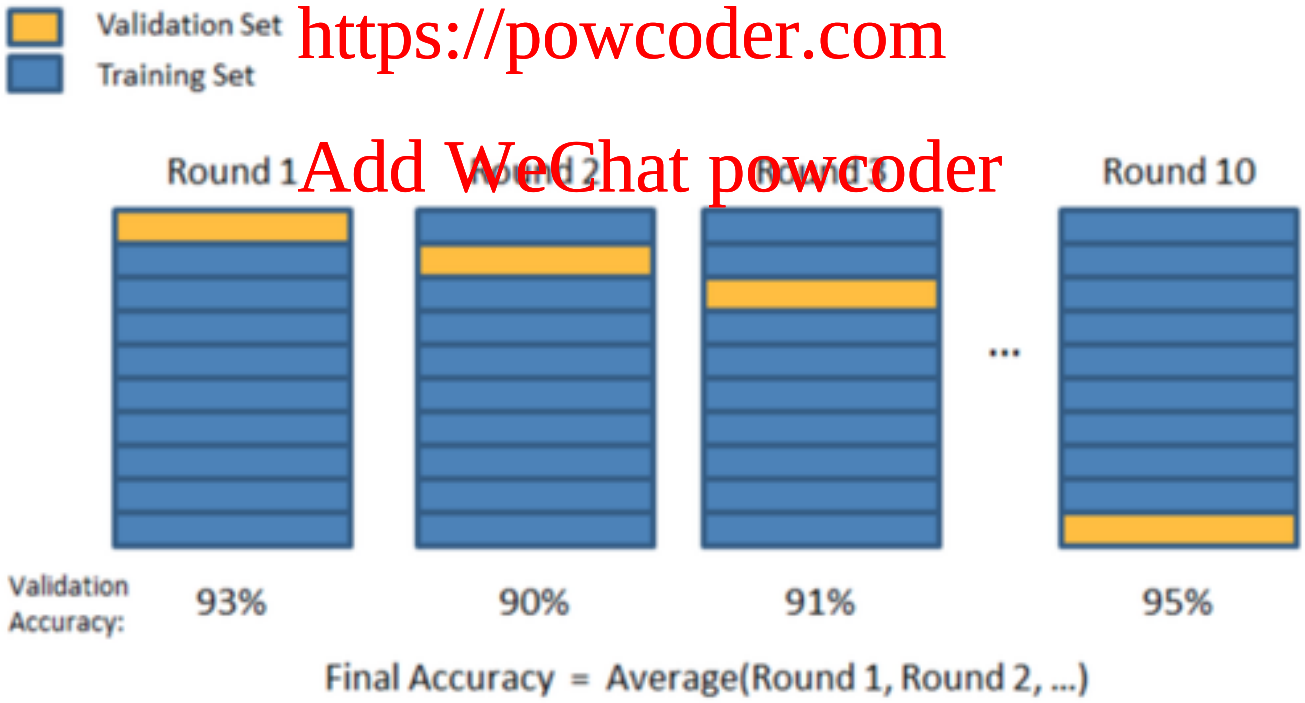
For each value of a hyperparameter, train on the train set, evaluate learned parameters on the validation set.

Pick the model with the hyper parameter that achieved the lowest validation error.

Report this model's test set error.

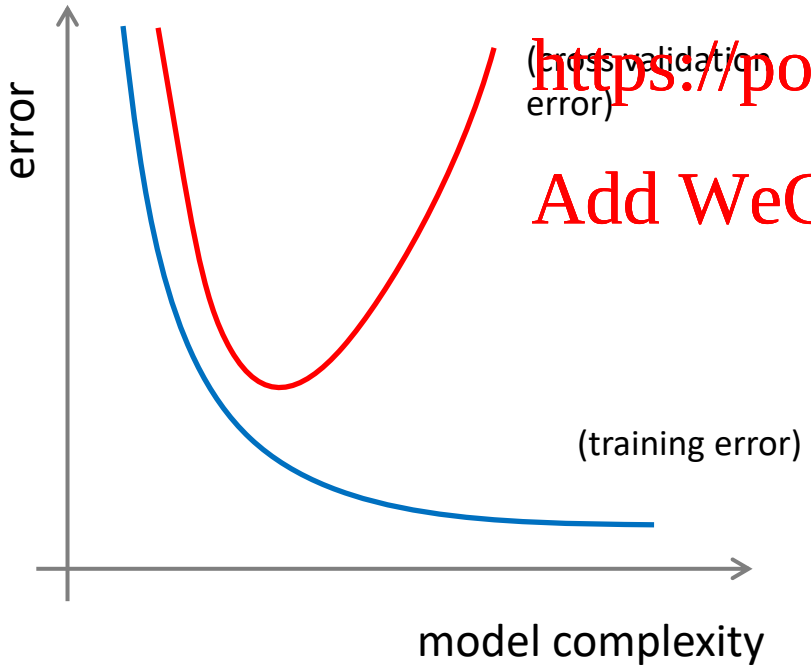
N-Fold Cross Validation

- What is we don't have enough data for train/test/validation sets?
- Solution: use N-fold cross validation.
- Split training set into train/validation sets N times
- Report average predictions over N val sets, e.g. N=10:



Diagnosing bias vs. variance

Suppose your learning algorithm is performing less well than you were hoping. ($J_{cv}(\theta)$ or $J_{test}(\theta)$ is high.) Is it a bias problem or a variance problem?



Bias (underfit):

$J_{train}(\theta)$ will be high,

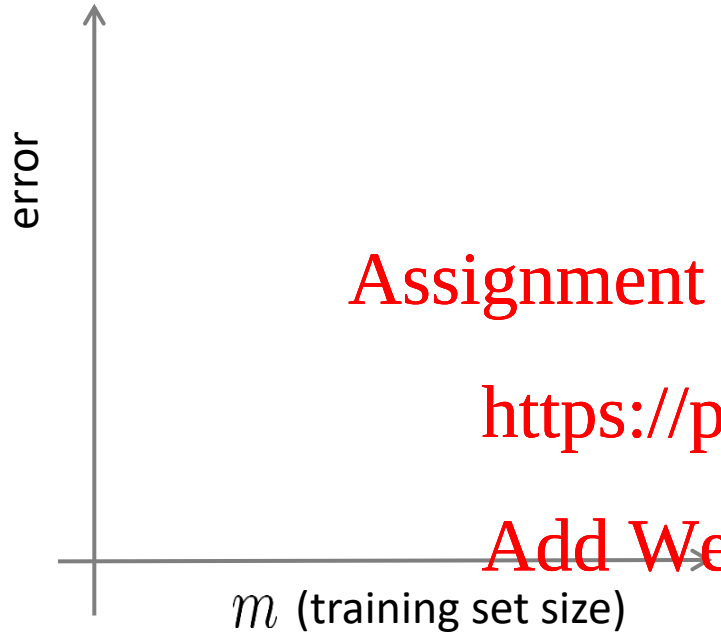
$J_{cv}(\theta) \approx J_{train}(\theta)$

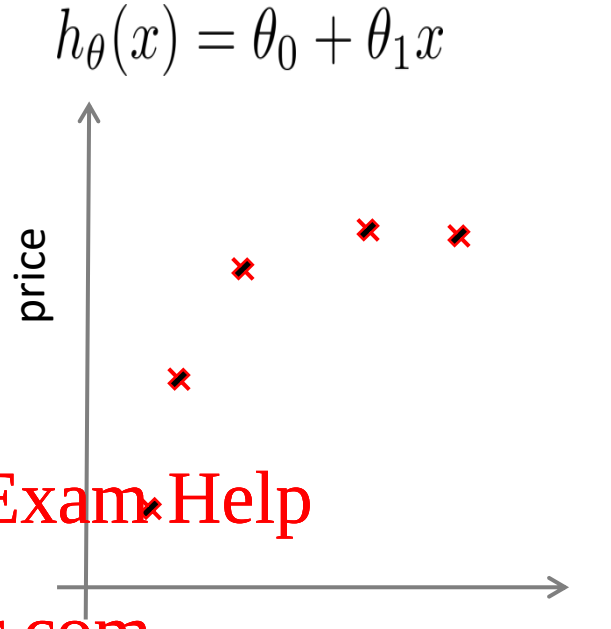
Variance (overfit):

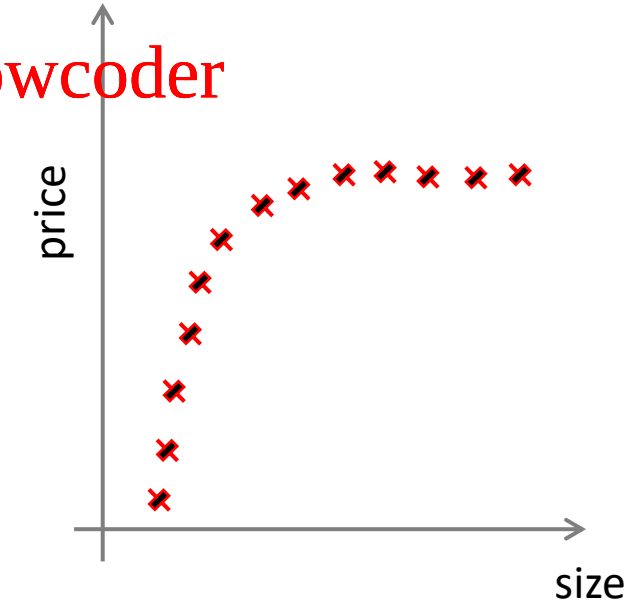
$J_{train}(\theta)$ will be low,

$J_{cv}(\theta) >> J_{train}(\theta)$

Learning Curves: High bias





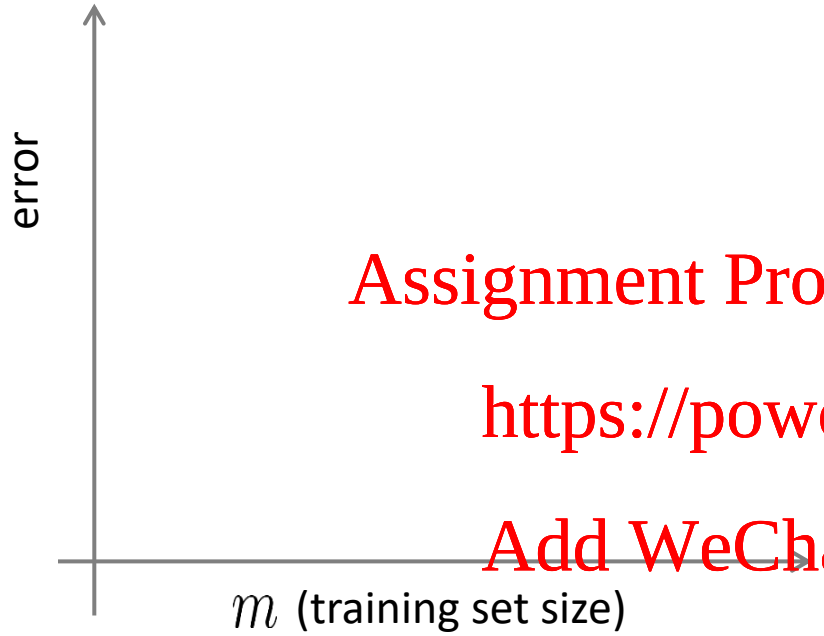


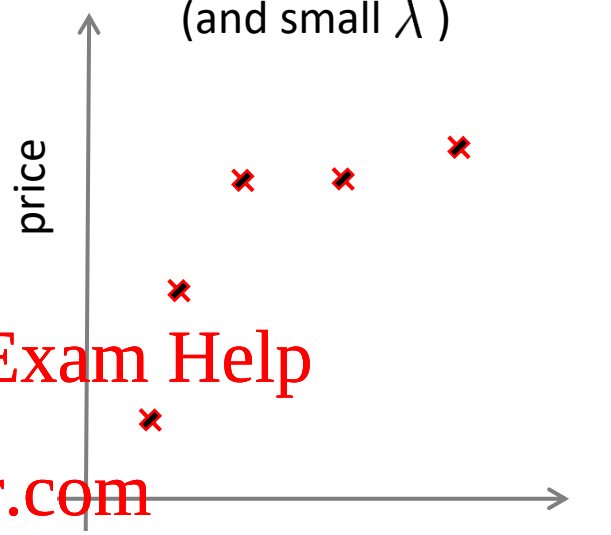
If a learning algorithm is suffering from high bias, getting more training data will not (by itself) help much.

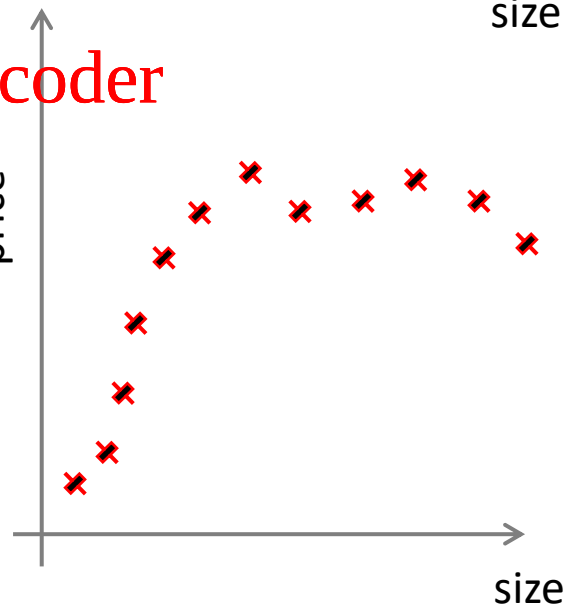
Learning Curves: High variance

$h_\theta(x) = \theta_0 + \theta_1 x + \cdots + \theta_{100} x^{100}$

(and small λ)







If a learning algorithm is suffering from high variance, getting more training data is likely to help.

Debugging a learning algorithm

Suppose you have implemented regularized linear regression to predict housing prices. However, when you test your hypothesis in a new set of houses, you find that it makes unacceptably large errors in its prediction. What should you try next?

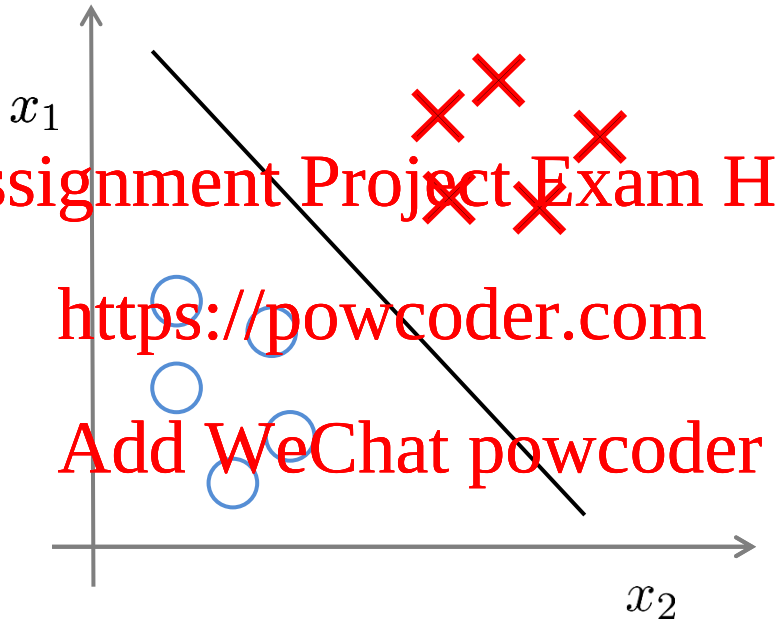
To fix high variance

- Get more training examples
- Try smaller sets of features
- Try increasing λ

To fix high bias

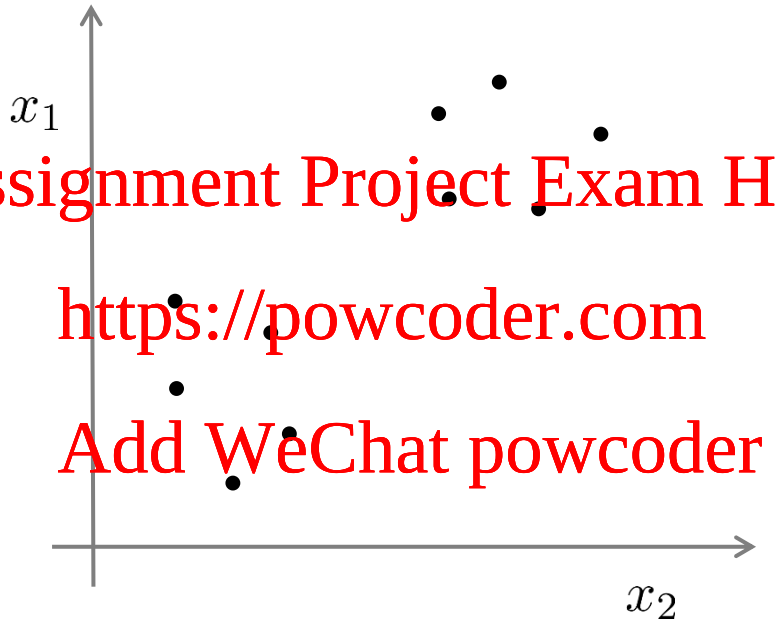
- Try getting additional features
- Try adding polynomial features
- Try decreasing λ

Supervised learning



Training set: $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), (x^{(3)}, y^{(3)}), \ldots, (x^{(m)}, y^{(m)})\}$

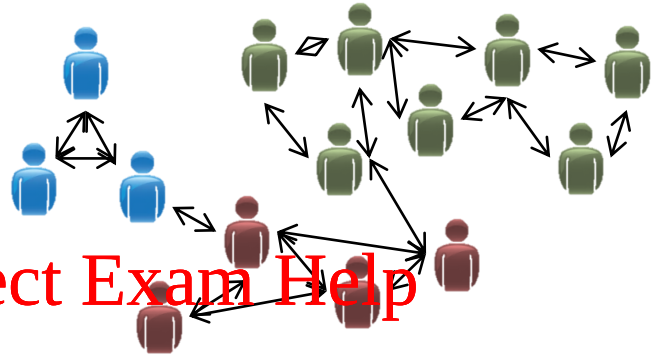
Unsupervised learning



Training set: $\{x^{(1)}, x^{(2)}, x^{(3)}, \ldots, x^{(m)}\}$

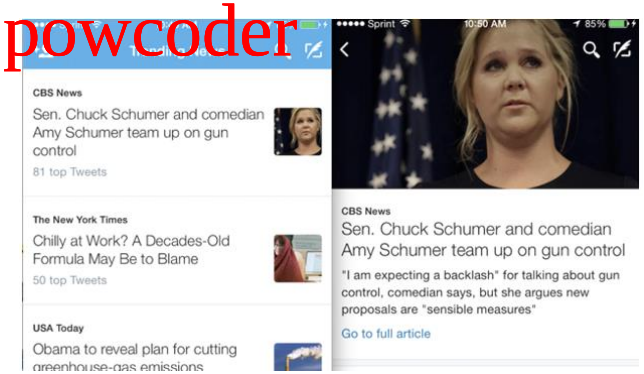
Clustering

Gene analysis

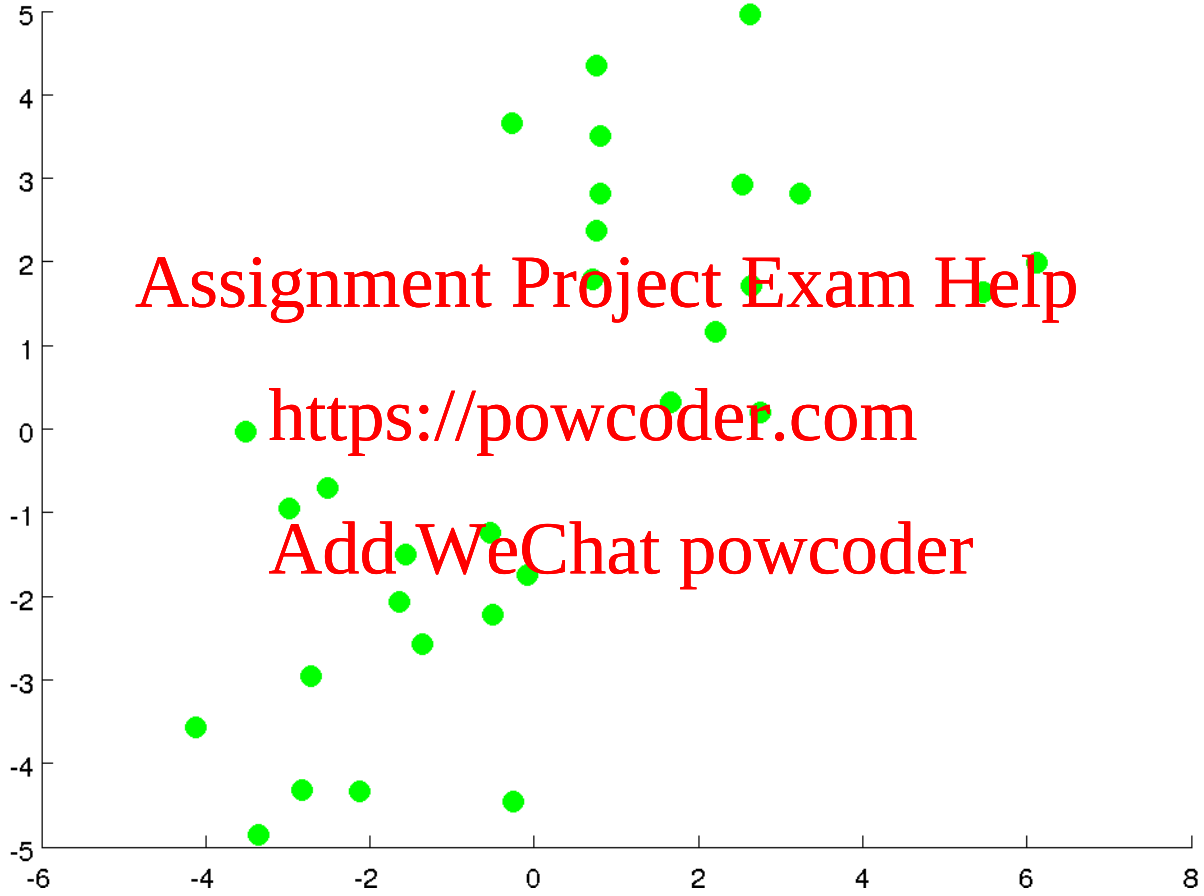


Social network analysis

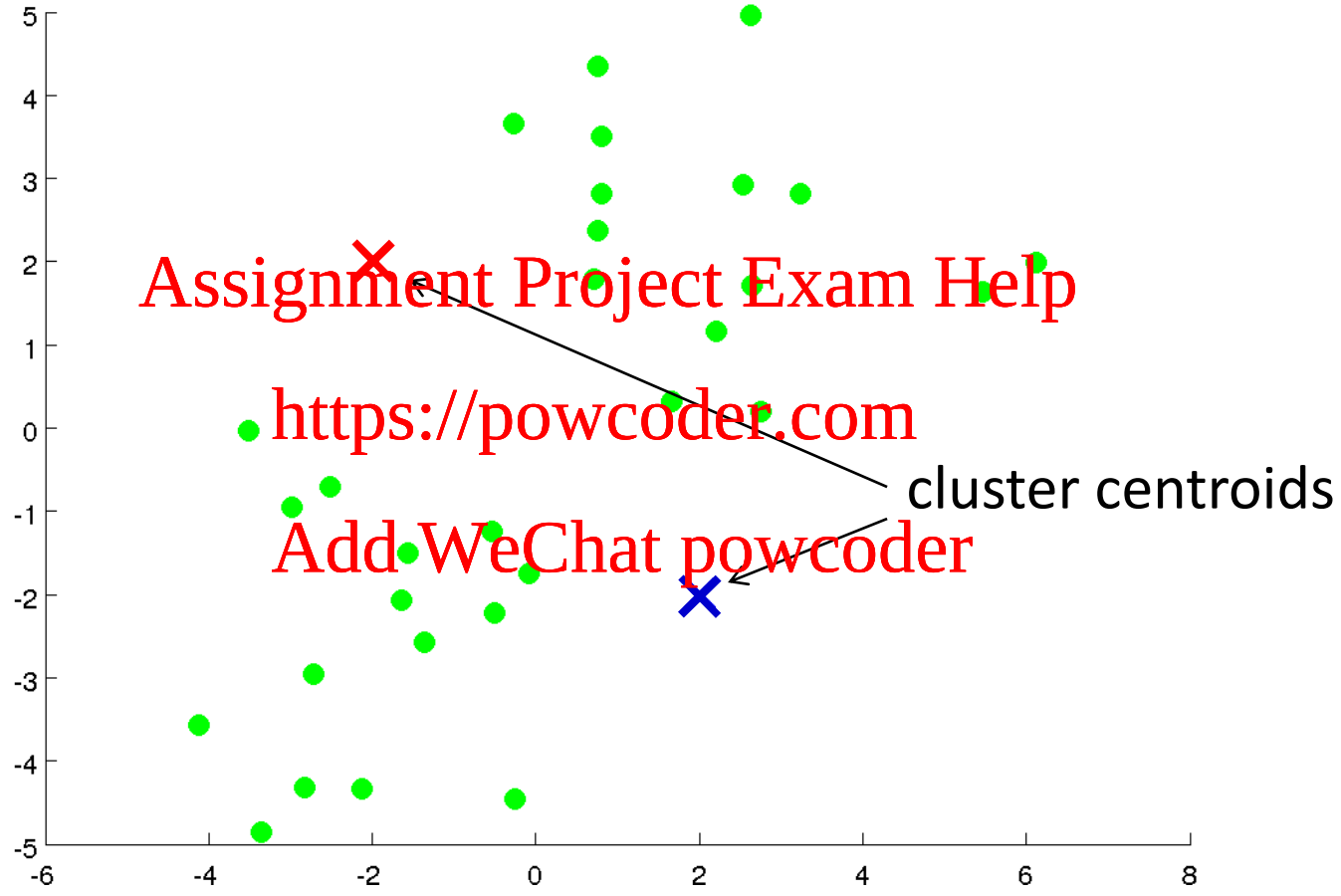
Types of voters



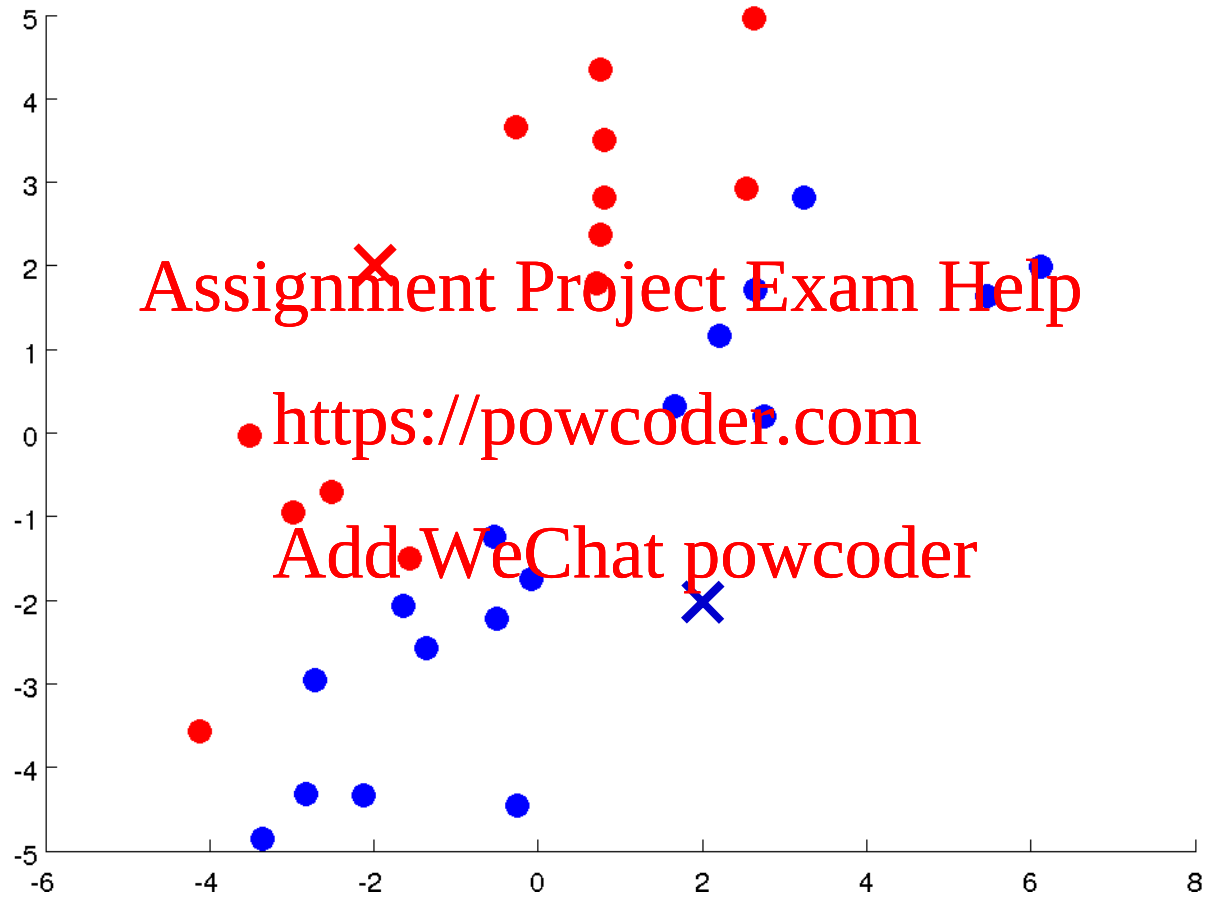
Trending news



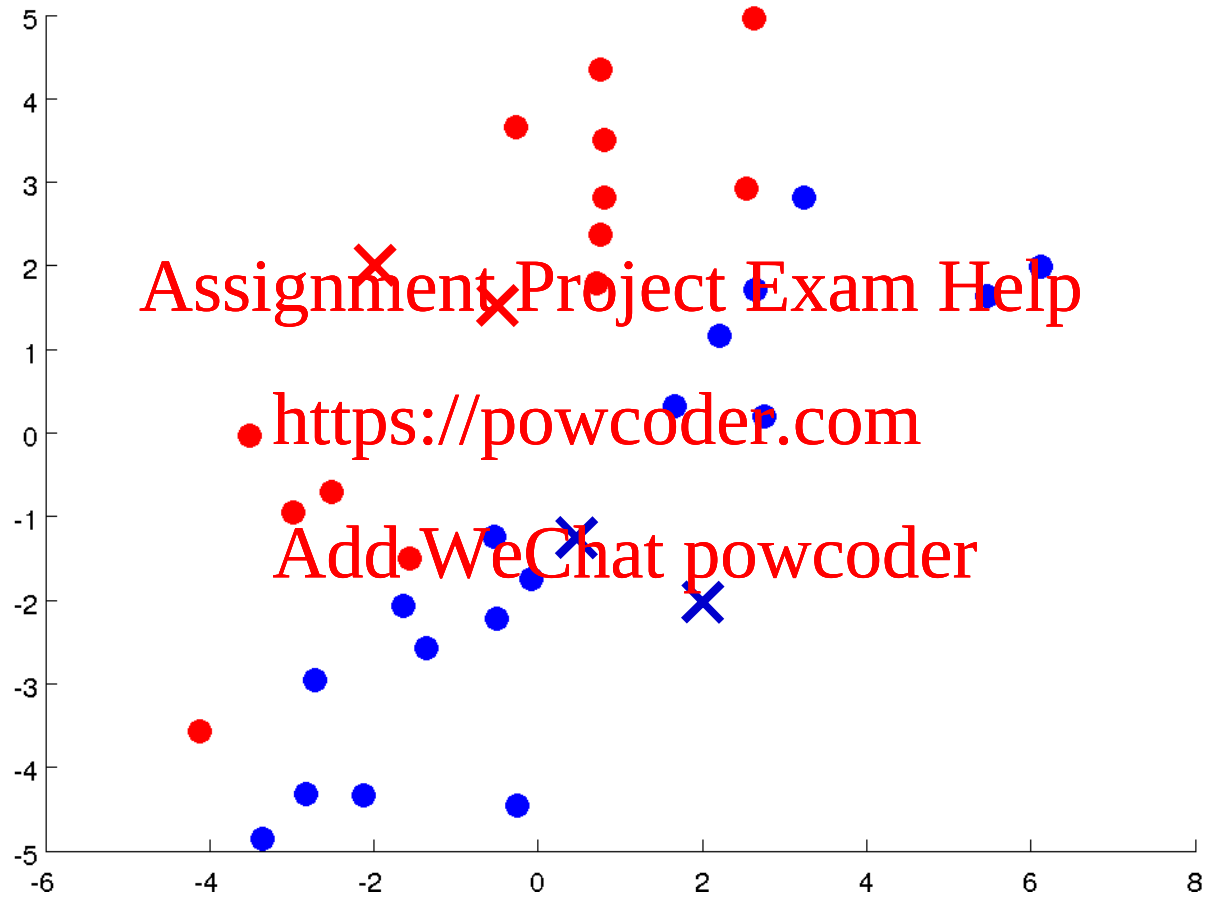
slide credit: Andrew Ng



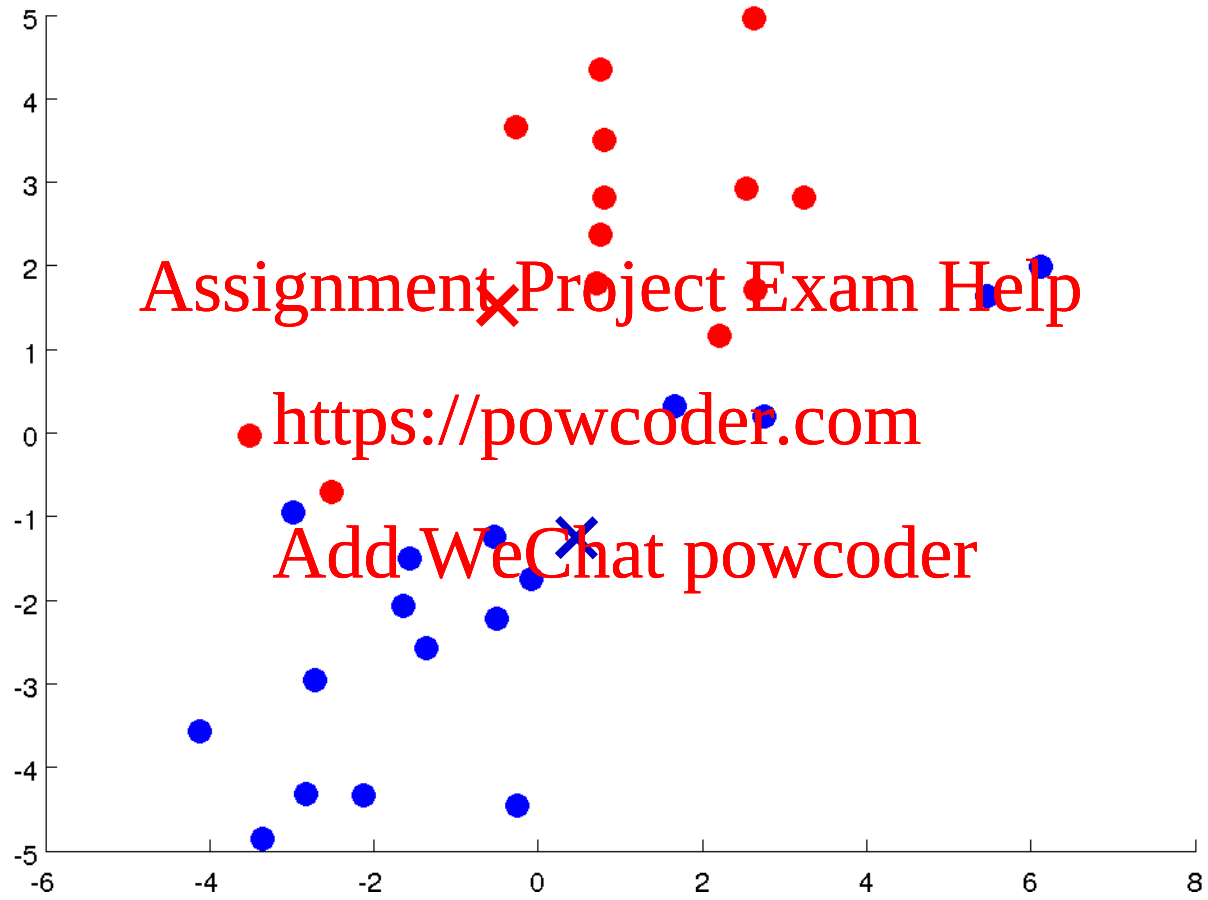
slide credit: Andrew Ng



slide credit: Andrew Ng



slide credit: Andrew Ng



slide credit: Andrew Ng

Next Class

Unsupervised Learning I: Clustering:

clustering, k-means, Gaussian mixtures.

Reading: Bishop 9.1-9.2

PSet 2 Out

- Due in 1 week: 9/24 11:59pm GMT -5 (Boston Time)
- Regression, gradient descent