

# **EECS 391**

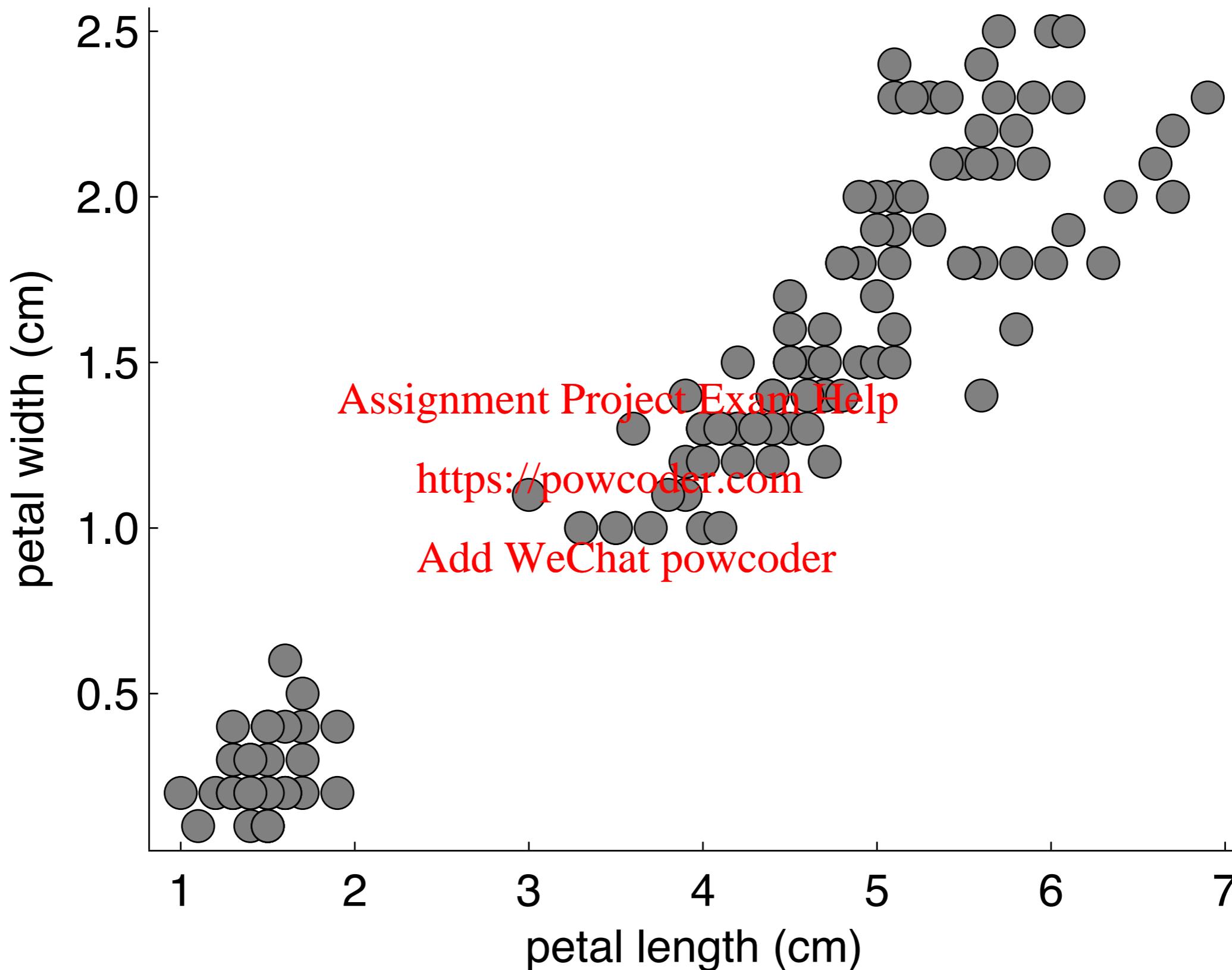
## **Intro to AI**

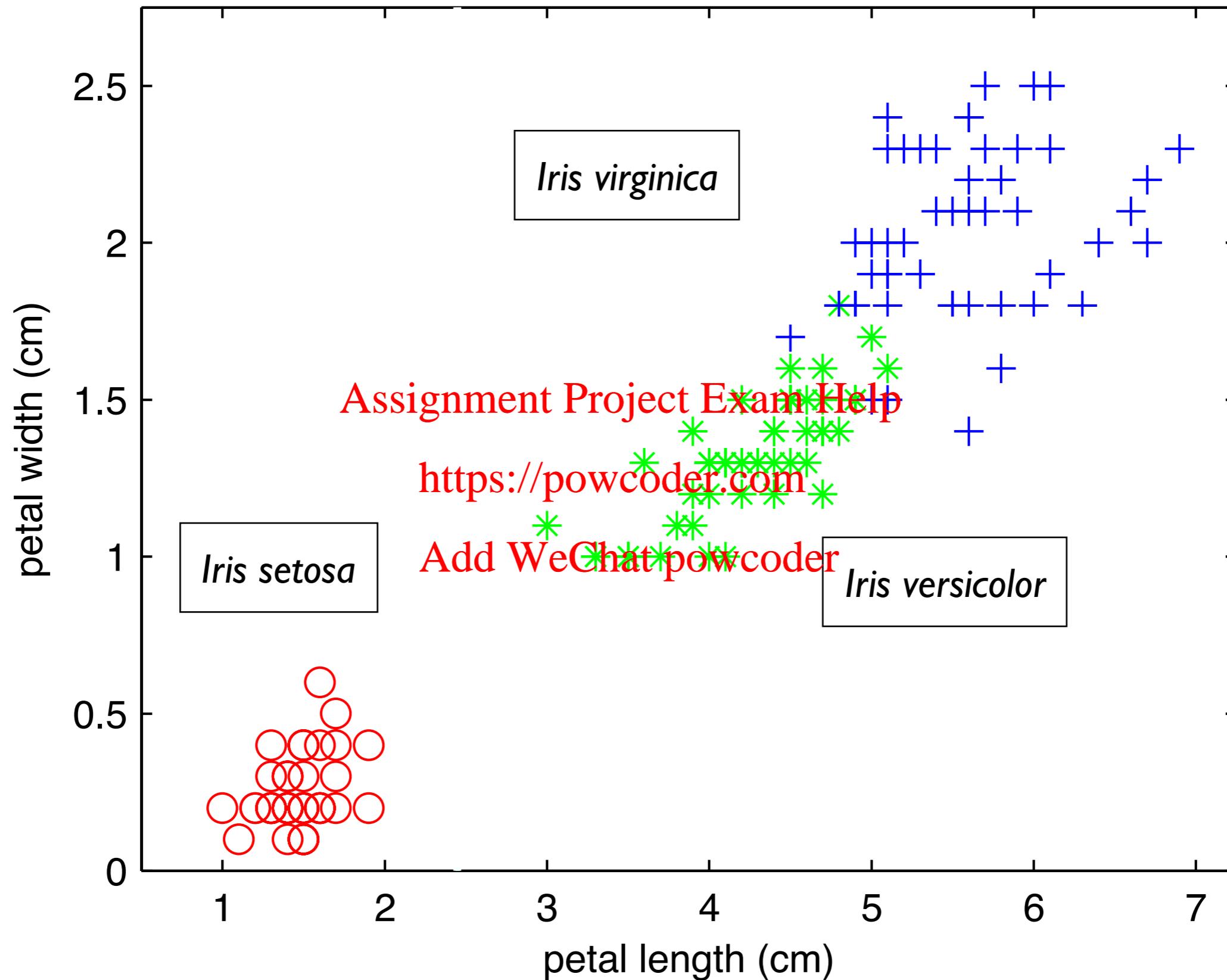
Assignment Project Exam Help  
**Unsupervised Learning and Clustering**  
<https://powcoder.com>

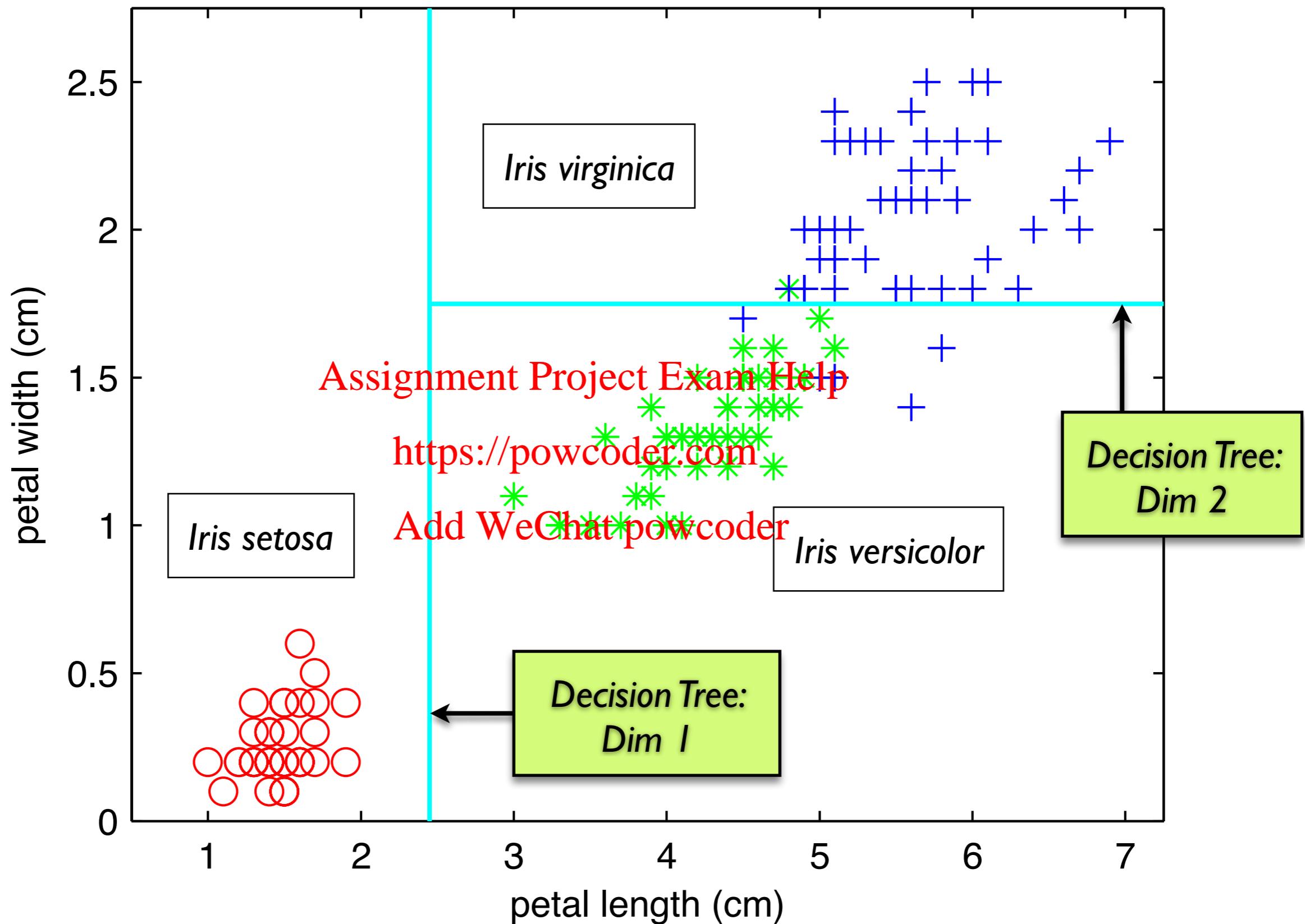
Add WeChat powcoder

L19 Tue Nov 13

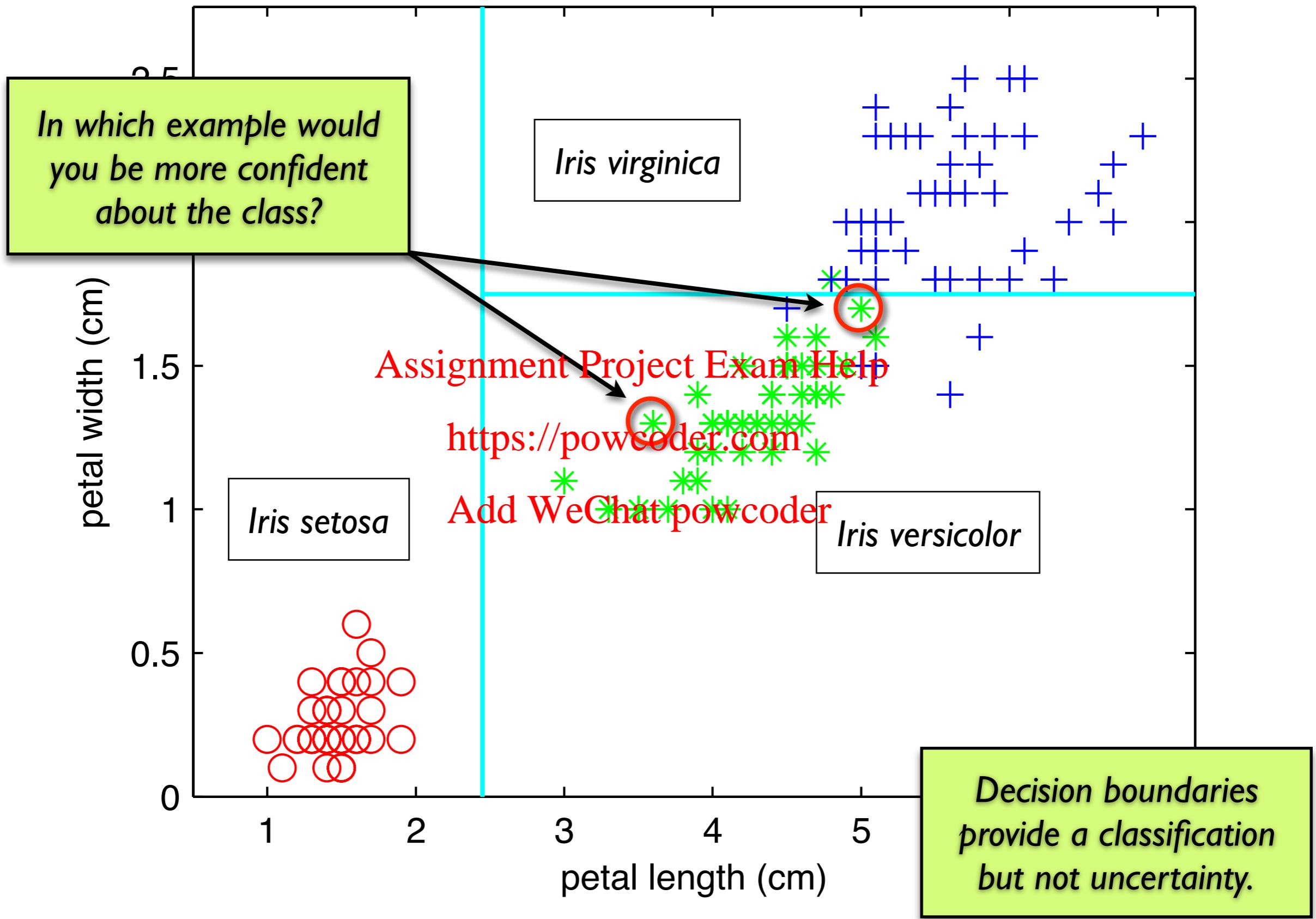
# Fisher's Iris data (unlabeled)



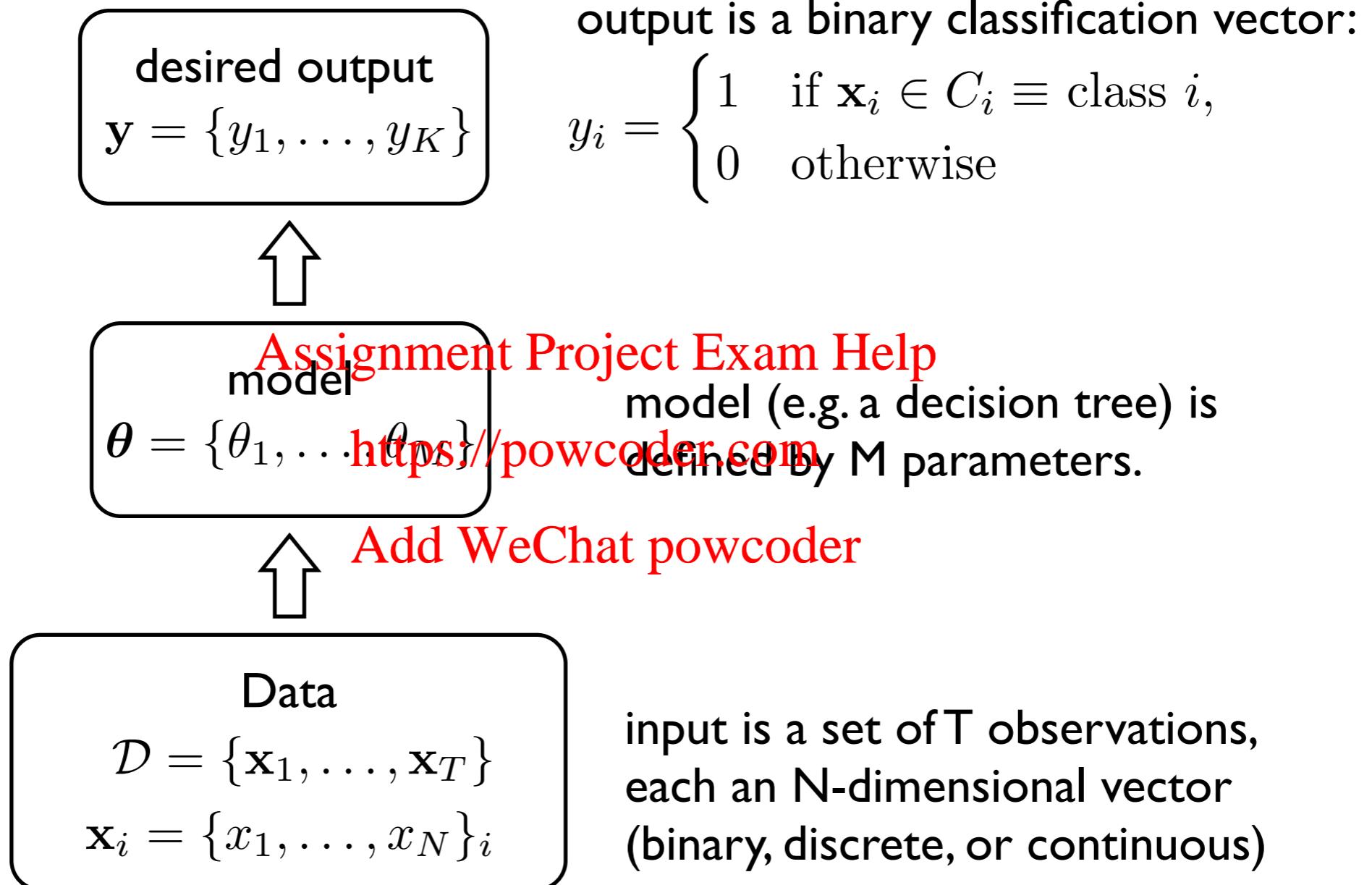




# Fisher's Iris data



# The general classification problem



Given data, we want to learn a model that can correctly classify novel observations.

How do we approach this probabilistically?

# The answer to all questions of uncertainty

- Let's apply Bayes' rule to infer the most probable class given the observation:

$$\begin{aligned} p(C_k | \mathbf{x}) &= \frac{p(\mathbf{x}|C_k)p(C_k)}{p(\mathbf{x})} \\ &= \frac{p(\mathbf{x}|C_k)p(C_k)}{\sum_k p(\mathbf{x}|C_k)p(C_k)} \end{aligned}$$

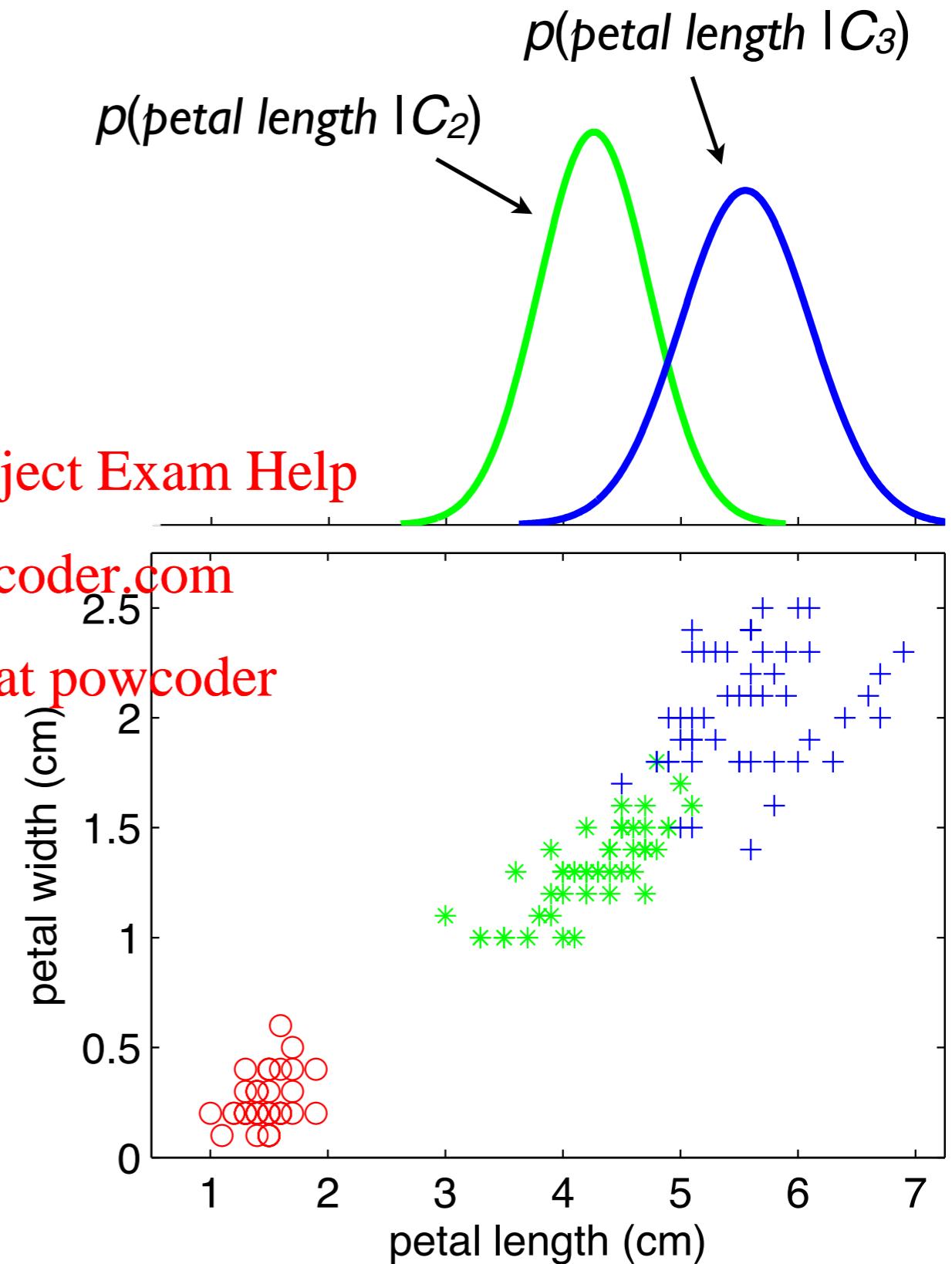
Assignment Project Exam Help

- This is the answer, but what does it mean?
- How do we specify the terms?
  - $p(C_k)$  is the prior probability on the different classes
  - $p(\mathbf{x}|C_k)$  is the data likelihood, ie probability of  $\mathbf{x}$  given class  $C_k$
- How should we define this?

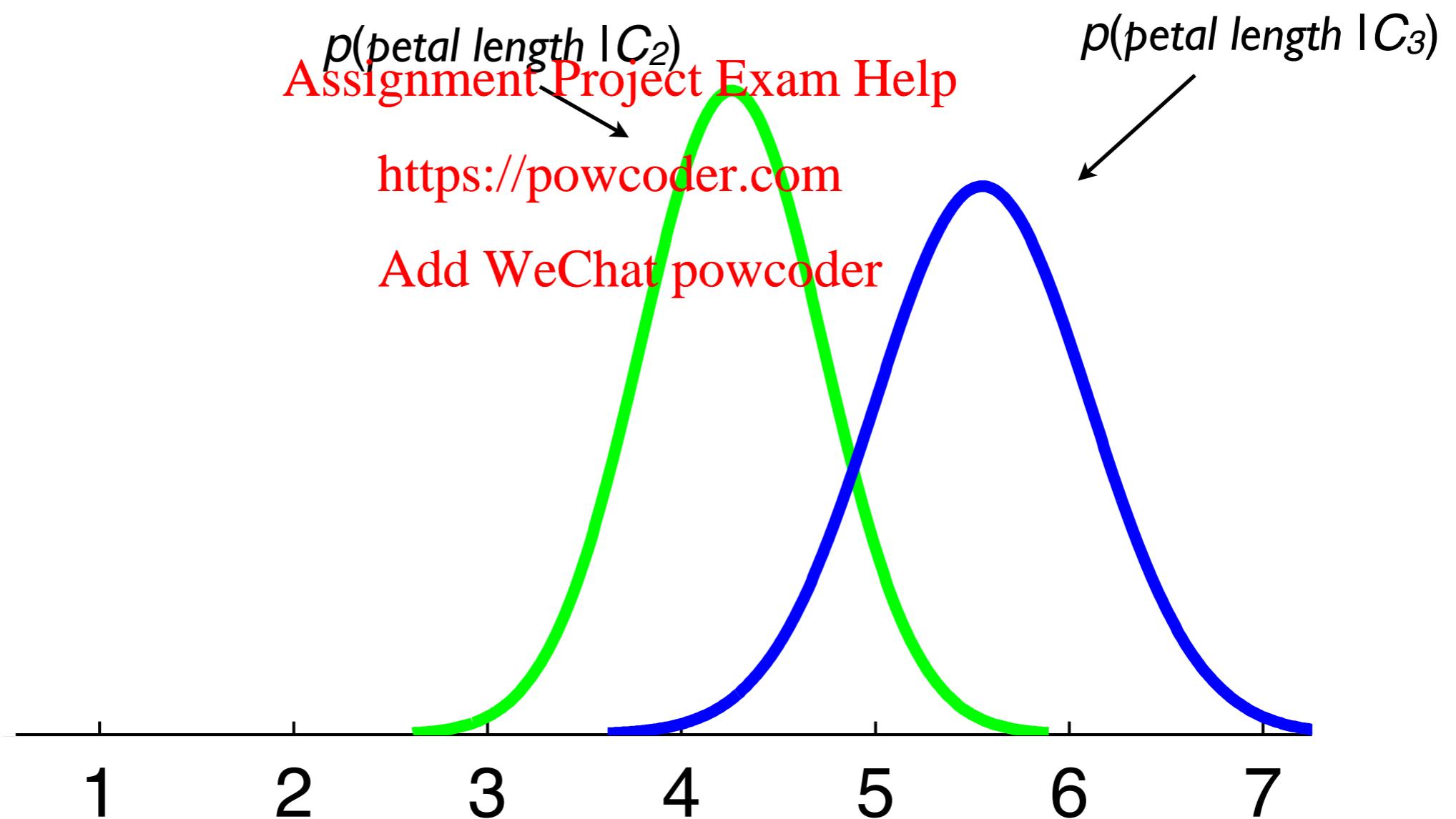
# What classifier would give “optimal” performance?

- Consider the iris data.
- How would we minimize the number of *future* mis-classifications?
- We would need to know the true *distribution* of the classes.
- Assume they follow a Gaussian distribution.
- The number of samples in each class is the same (50), so (assume)  $p(C_k)$  is equal for all classes.
- Because  $p(\mathbf{x})$  is the same for all classes we have:

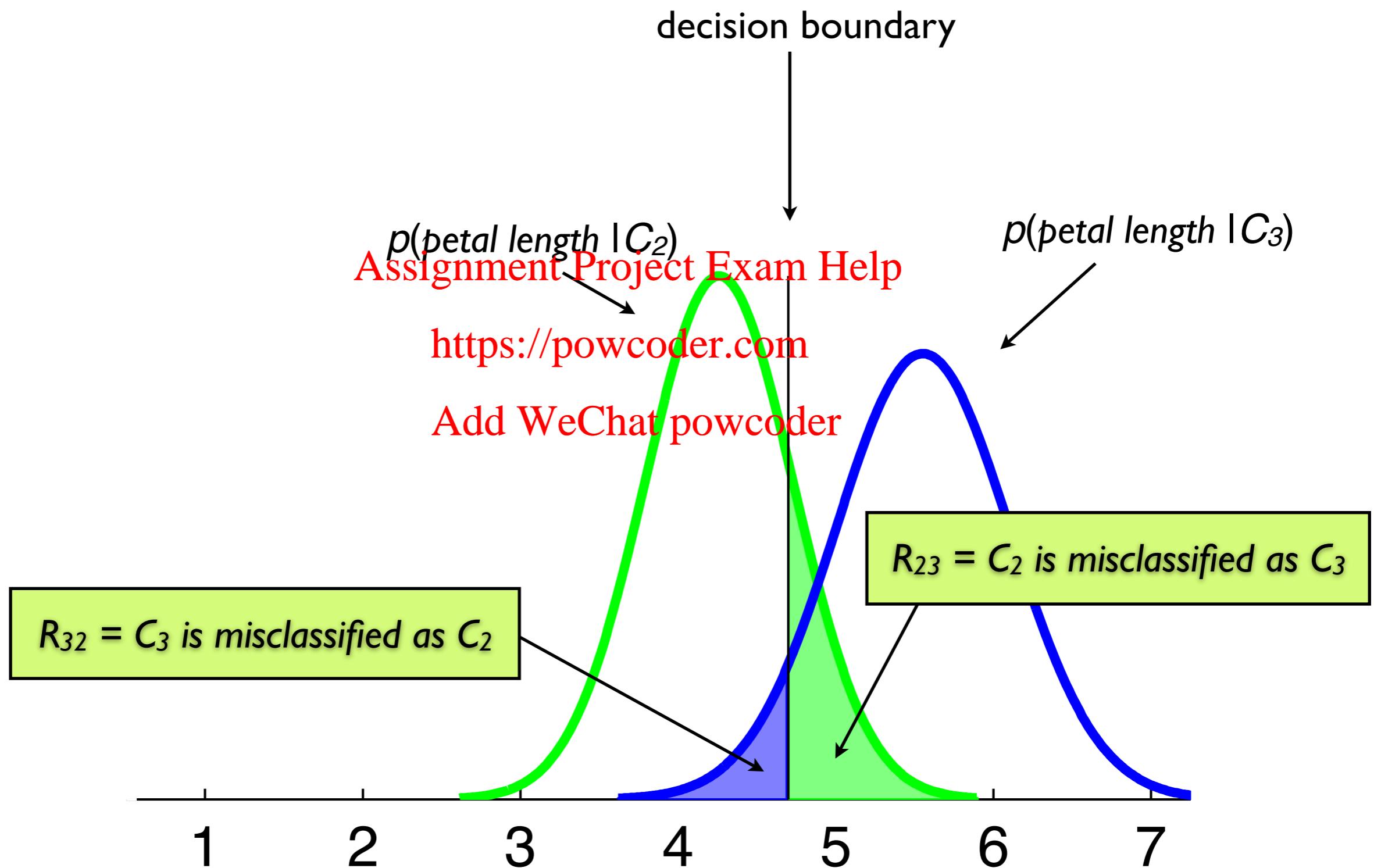
$$\begin{aligned} p(C_k | \mathbf{x}) &= \frac{p(\mathbf{x} | C_k)p(C_k)}{p(\mathbf{x})} \\ &\propto p(\mathbf{x} | C_k)p(C_k) \end{aligned}$$



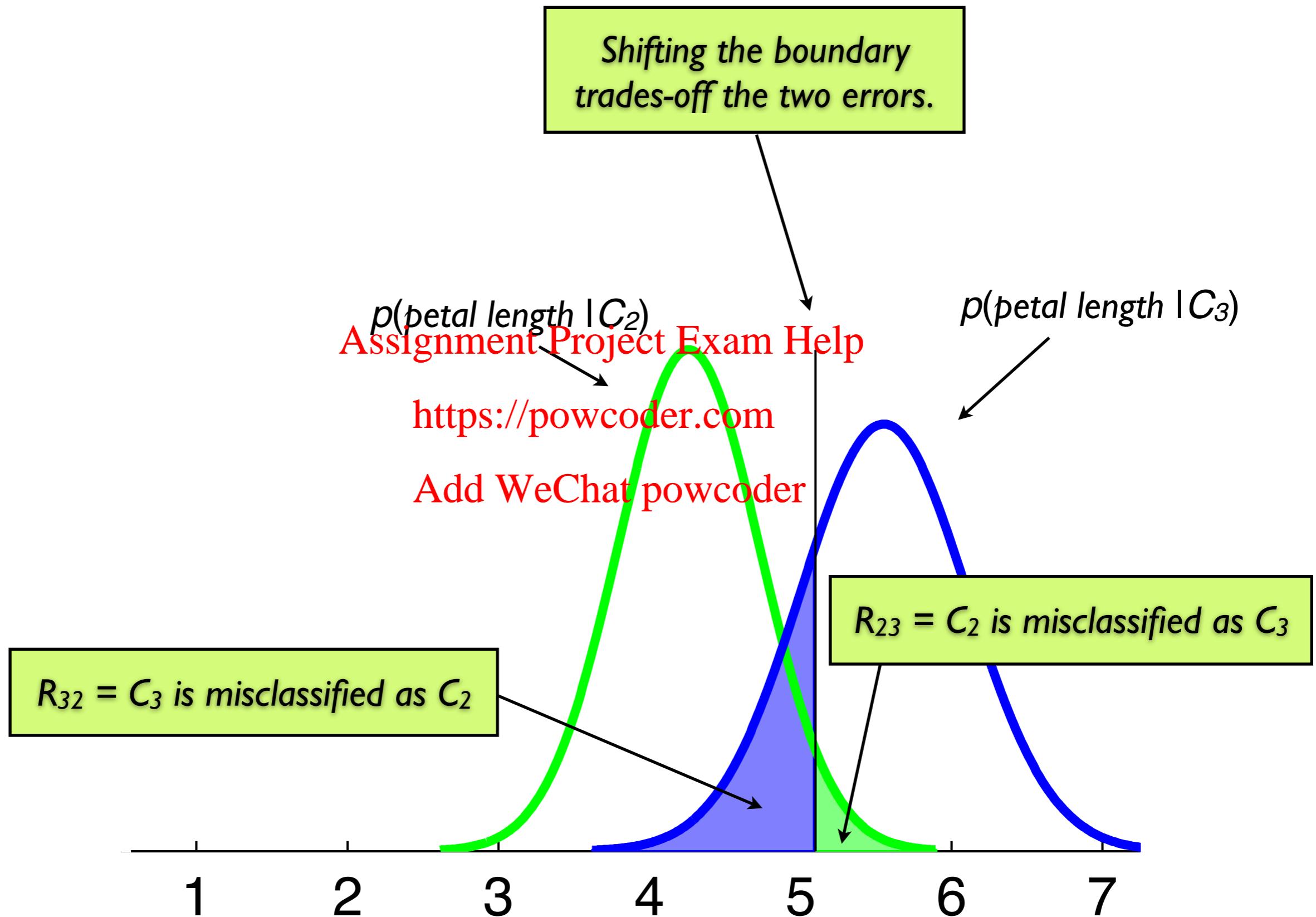
# Where do we put the boundary?



# Where do we put the boundary?



# Where do we put the boundary?

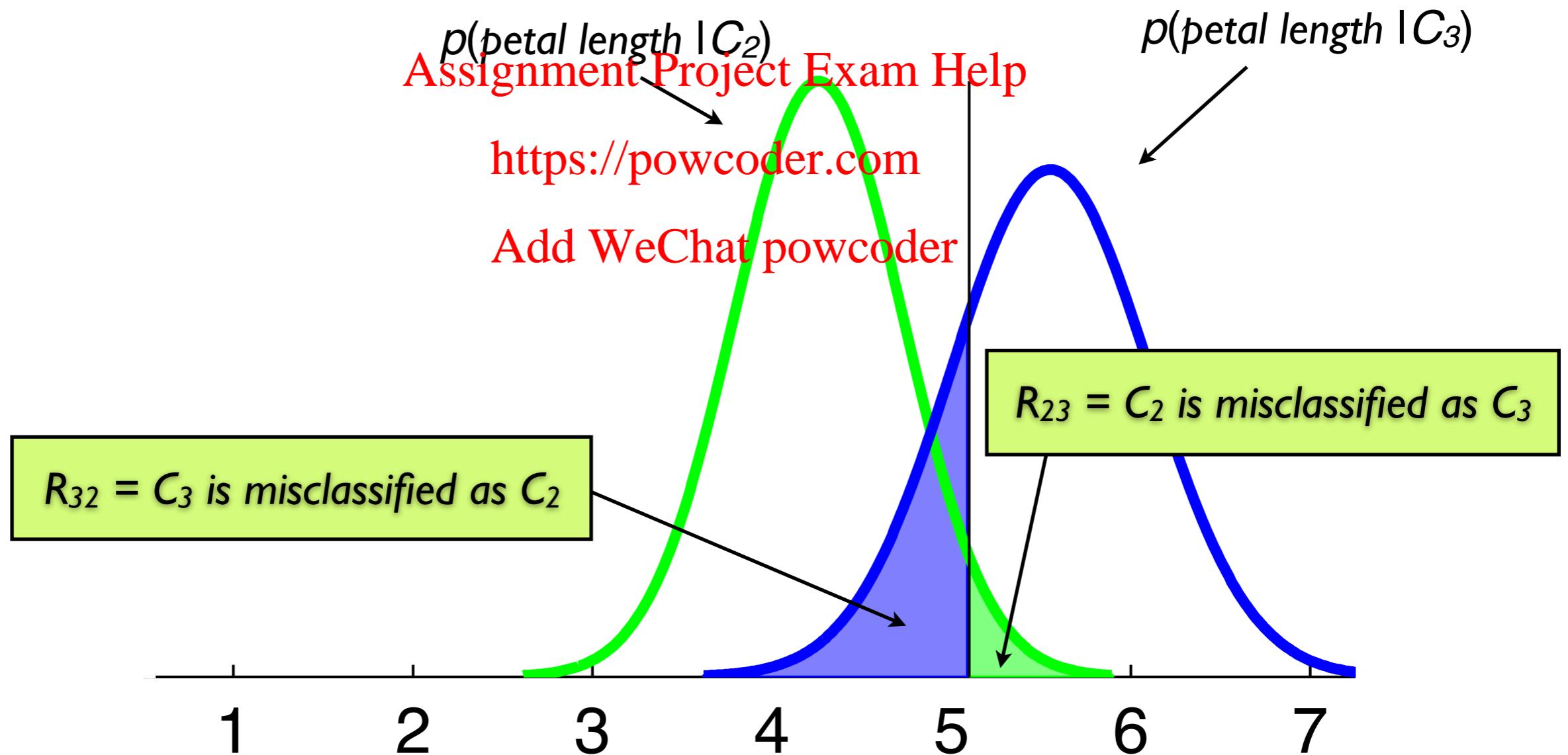


# Where do we put the boundary?

- The misclassification error is defined by

$$p(\text{error}) = \int_{R_{32}} p(\mathbf{x}|C_3)P(C_3)d\mathbf{x} + \int_{R_{23}} p(\mathbf{x}|C_2)P(C_2)d\mathbf{x}$$

- which in our case is proportional to the data likelihood

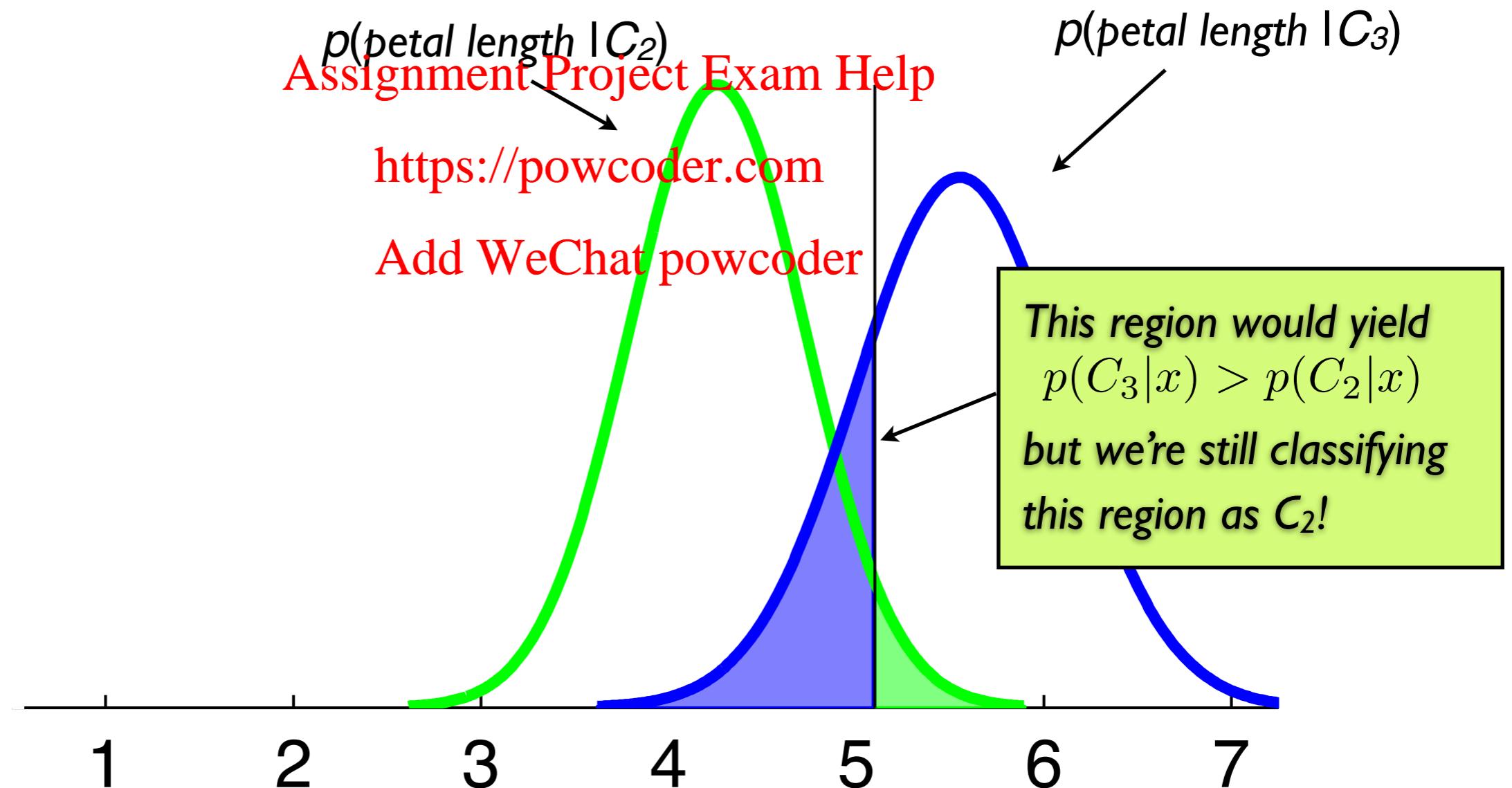


# Where do we put the boundary?

- The misclassification error is defined by

$$p(\text{error}) = \int_{R_{32}} p(\mathbf{x}|C_3)P(C_3)d\mathbf{x} + \int_{R_{23}} p(\mathbf{x}|C_2)P(C_2)d\mathbf{x}$$

- which in our case is proportional to the data likelihood

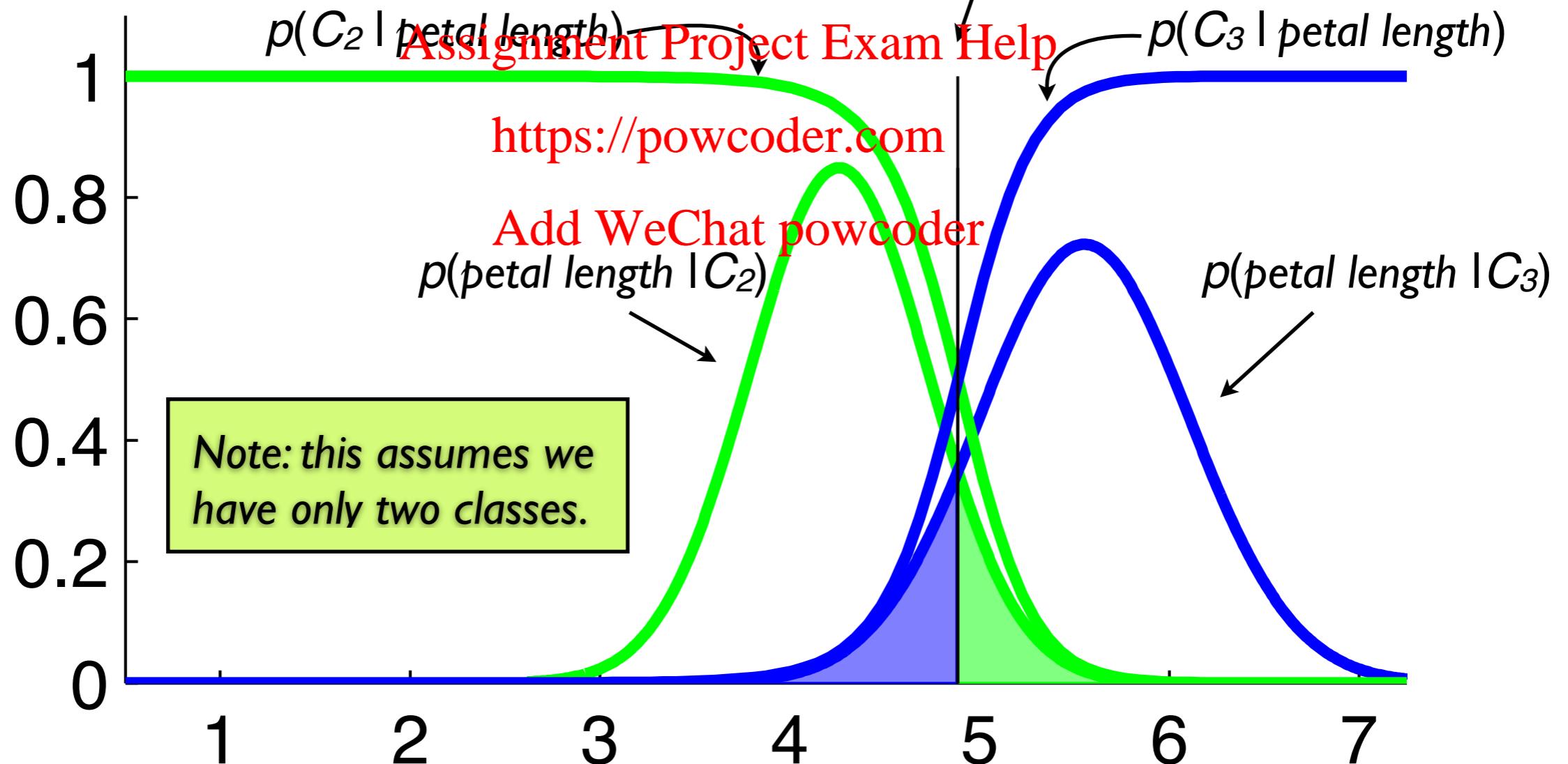


# The optimal decision boundary

- The minimal misclassification error at the point where

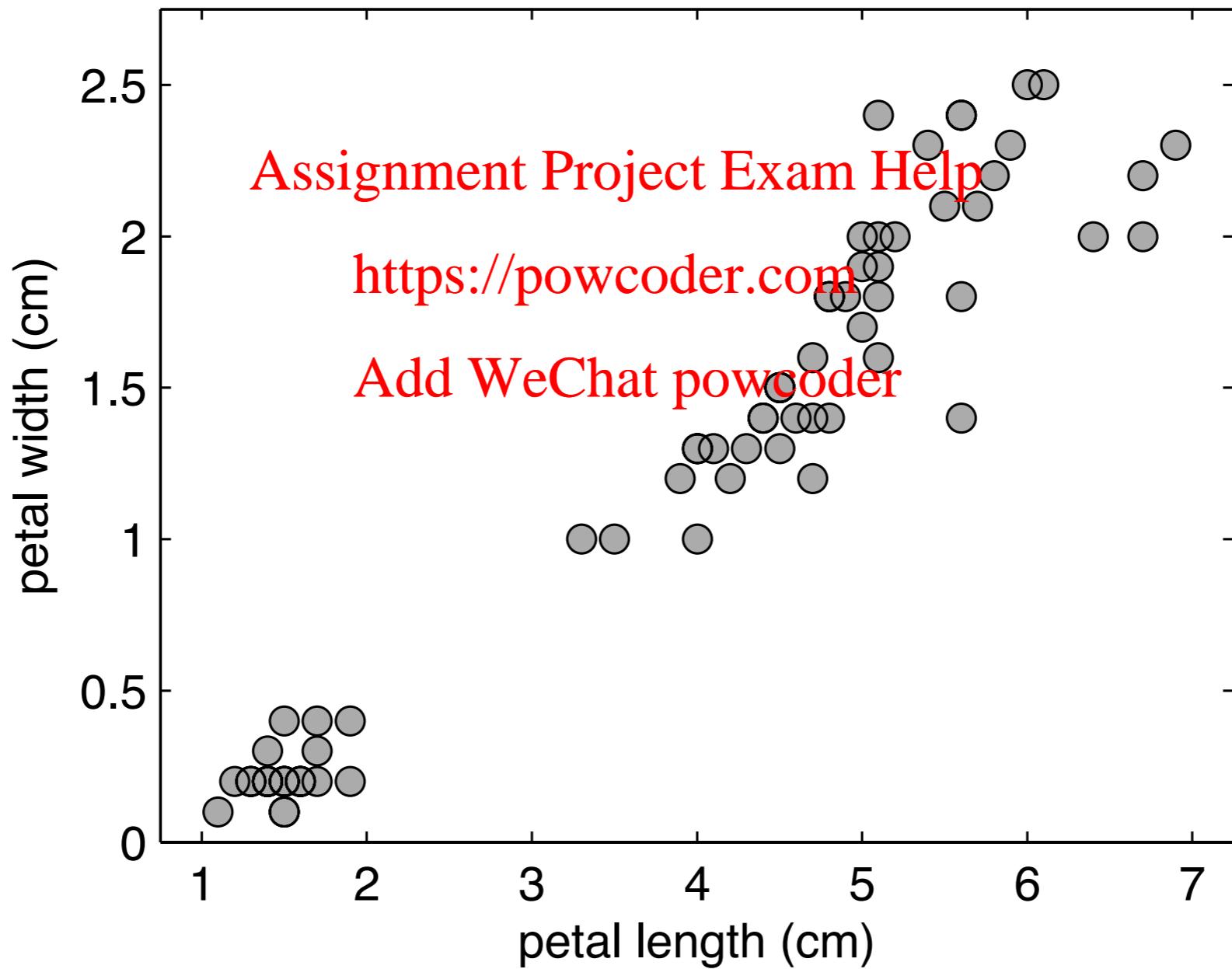
$$\begin{aligned} p(C_3|x) &= p(C_2|x) \\ \Rightarrow p(x|C_3)p(C_3)/p(x) &= p(x|C_2)p(C_2)/p(x) \\ \Rightarrow p(x|C_3) &= p(x|C_2) \end{aligned}$$

Optimal decision boundary



# Clustering: Classification without labels

- In many situations we don't have labeled training data, only unlabeled data.
- Eg, in the iris data set, what if we were just starting and didn't know any classes?

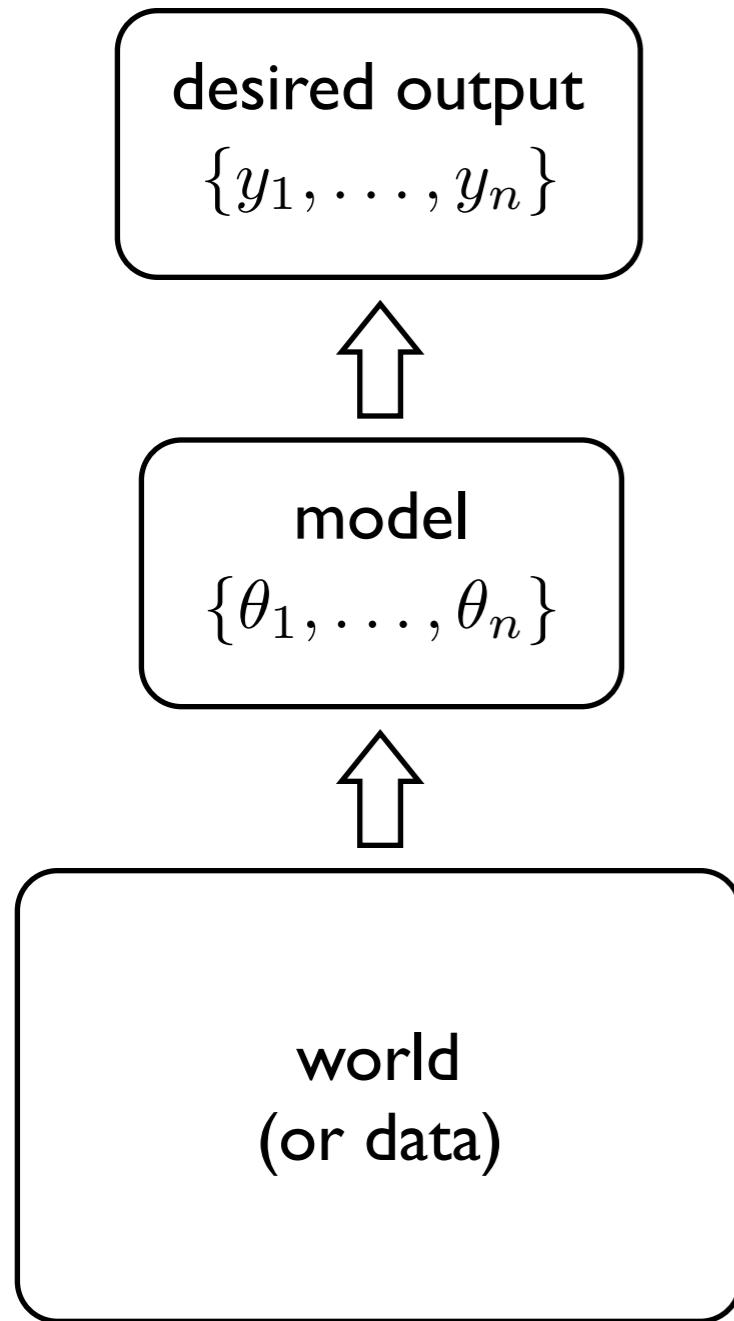


# Types of learning

**supervised**

**unsupervised**

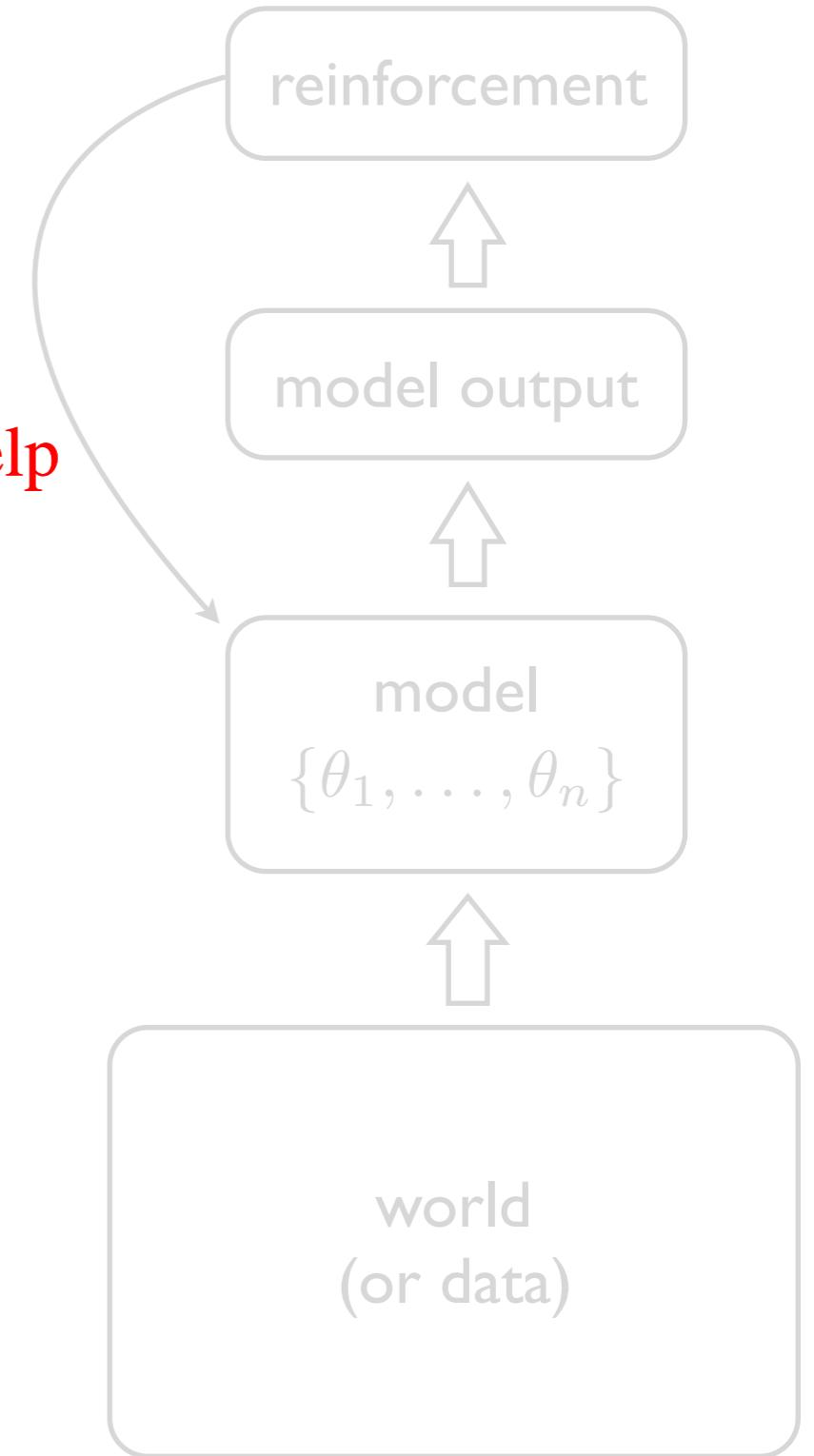
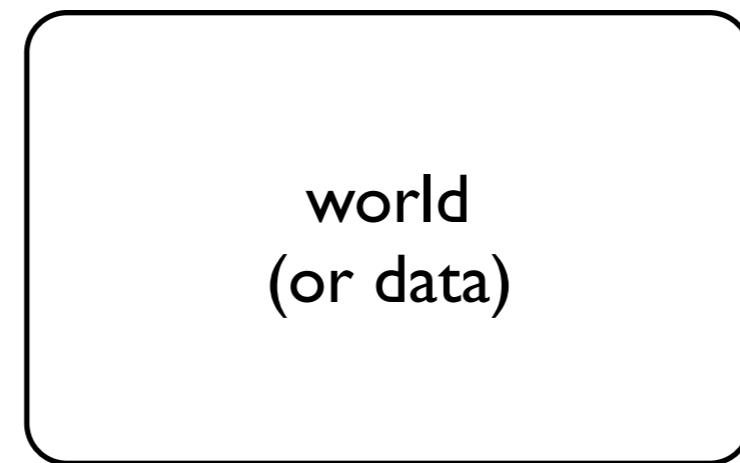
**reinforcement**



Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder  
model  $\{\theta_1, \dots, \theta_n\}$



# A different approach to classification

- Nearby points are likely to be members of the same class.
- What if we used the points themselves to classify?

*classify  $\mathbf{x}$  in  $C_k$  if  $\mathbf{x}$  is “similar” to a point we already know is in  $C_k$ .*

Assignment Project Exam Help

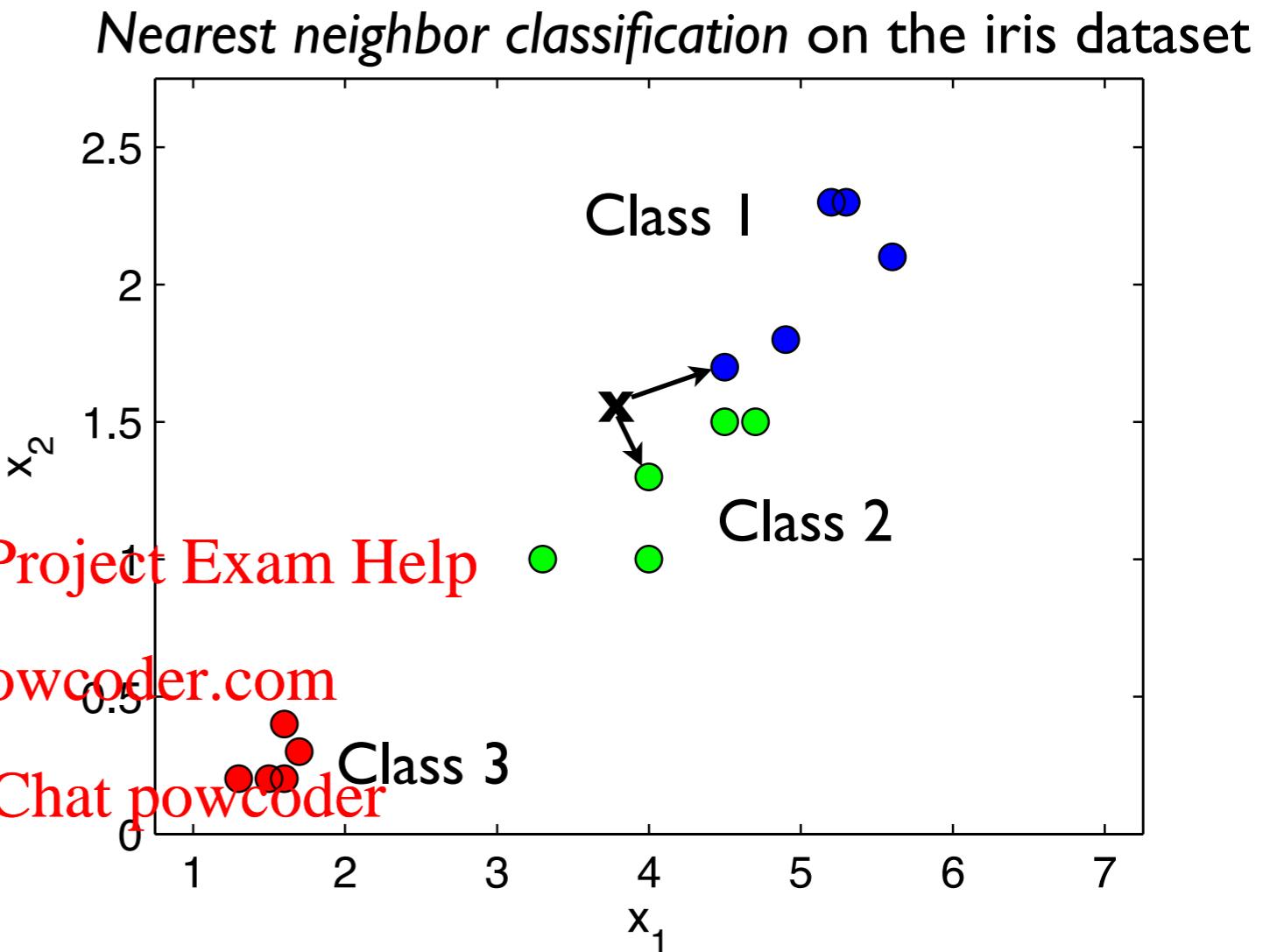
<https://powcoder.com>

Add WeChat powcoder

- Eg: unclassified point  $\mathbf{x}$  is more similar Class 2 than Class 1.
- Issue: How to define “similar” ?  
Simplest is Euclidean distance:

$$d(\mathbf{x}, \mathbf{y}) = \sum_i (x_i - y_i)^2$$

- Could define other metrics depending on application, e.g. text documents, images, etc.



Potential advantages:

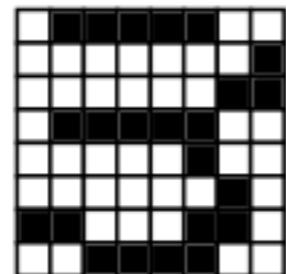
- don't need an explicit model
- the more examples the better
- might handle more complex classes
- easy to implement
- “no brain on part of the designer”

# Example: Handwritten digits

3 6 8 1 7 9 6 6 9 1  
6 7 5 7 8 6 3 4 8 5  
2 1 7 9 7 1 2 8 4 5  
4 8 1 9 0 1 8 8 9 4  
7 6 1 8 6 4 1 5 6 0  
7 5 9 2 6 5 8 1 9 7  
1 2 2 2 2 3 4 4 8 0  
0 2 3 8 0 7 3 8 5 7  
0 1 4 6 4 6 0 2 4 3  
7 1 2 8 1 6 9 8 6 1

from LeCun et al, 1998  
digit data available at:  
<http://yann.lecun.com/exdb/mnist/>

Digits are just represented as a vector.



- Use Euclidean distance to see which known digit is closest to each class.
- But not all neighbors are the same:

example nearest neighbors

0 0 0 0 0 0 6  
2 2 2 8 8 8 7  
4 4 4 4 4 4 4  
9 9 9 9 9 9 9  
7 7 7 7 7 7 7

example from Sam Roweis

- “k-nearest neighbors”: look at k-nearest neighbors and choose most frequent.
- Cautions: can get expensive to find neighbors

Assignment Project Exam Help

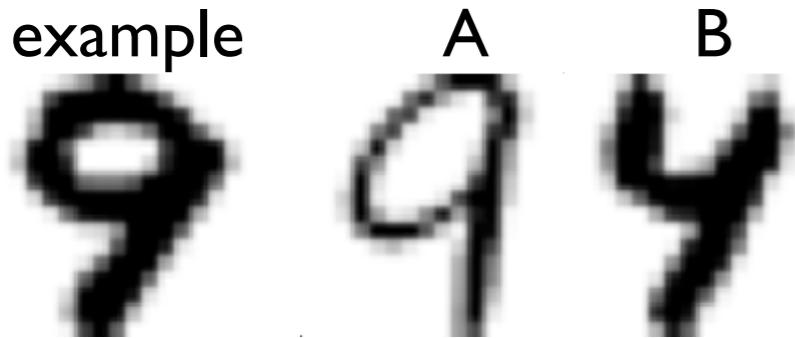
<https://powcoder.com>

Add WeChat powcoder

# The problem of using templates (ie Euclidean distance)

- Which of these is more like the example? A or B?

example



from Simard et al, 1998

A      B

performance results of various classifiers  
(from <http://yann.lecun.com/exdb/mnist/>)

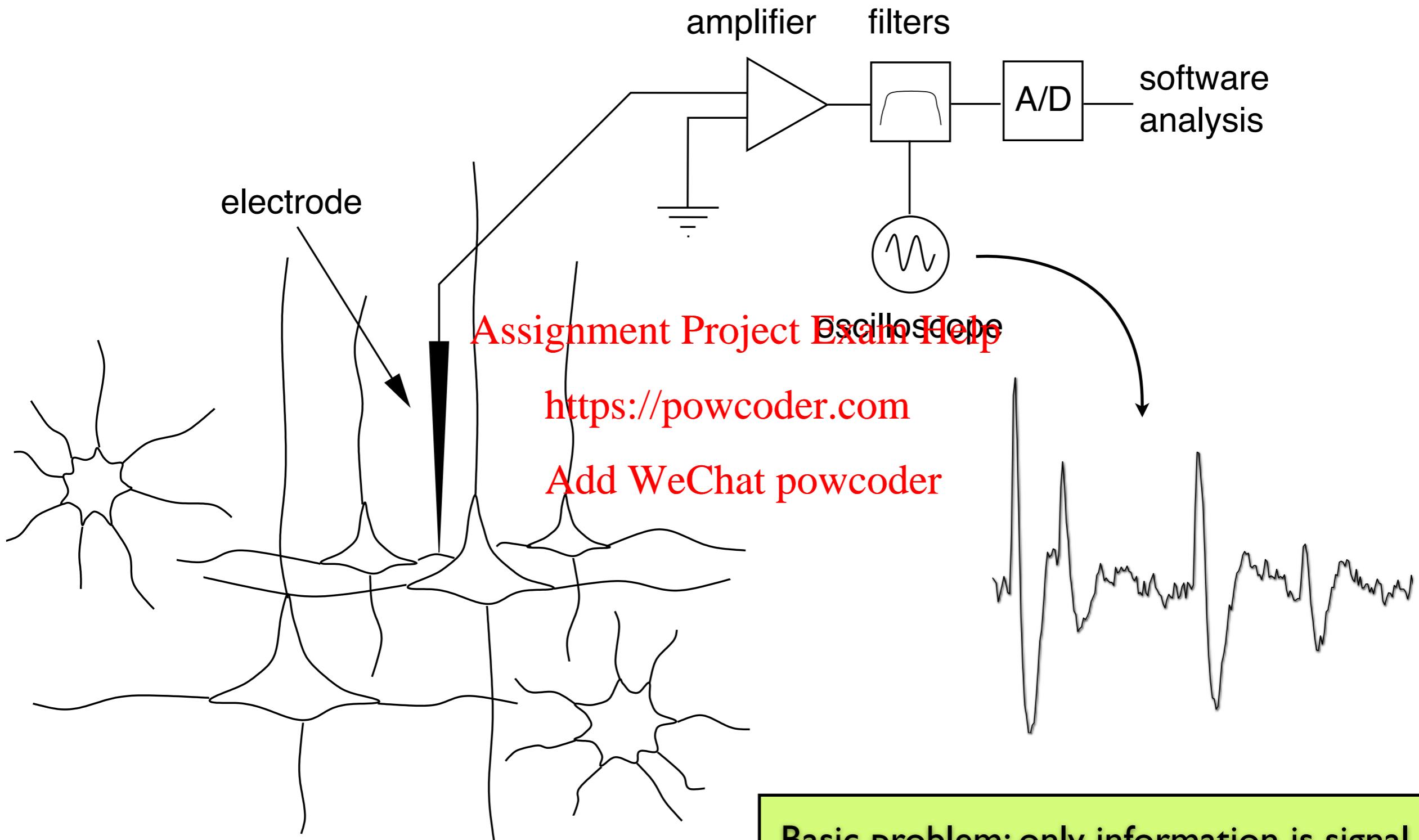
Classifier	error rate on test data (%)
linear	12
k=3 nearest neighbor (Euclidean distance)	5
2-layer neural network (300 hidden units)	4.7
nearest neighbor (Euclidean distance)	3.1
k-nearest neighbor (improved distance metric)	1.1
convolutional neural net	0.95
best (the conv. net with elastic distortions)	0.4
humans	0.2 - 2.5

Assignment Project Exam Help

<https://powcoder.com>

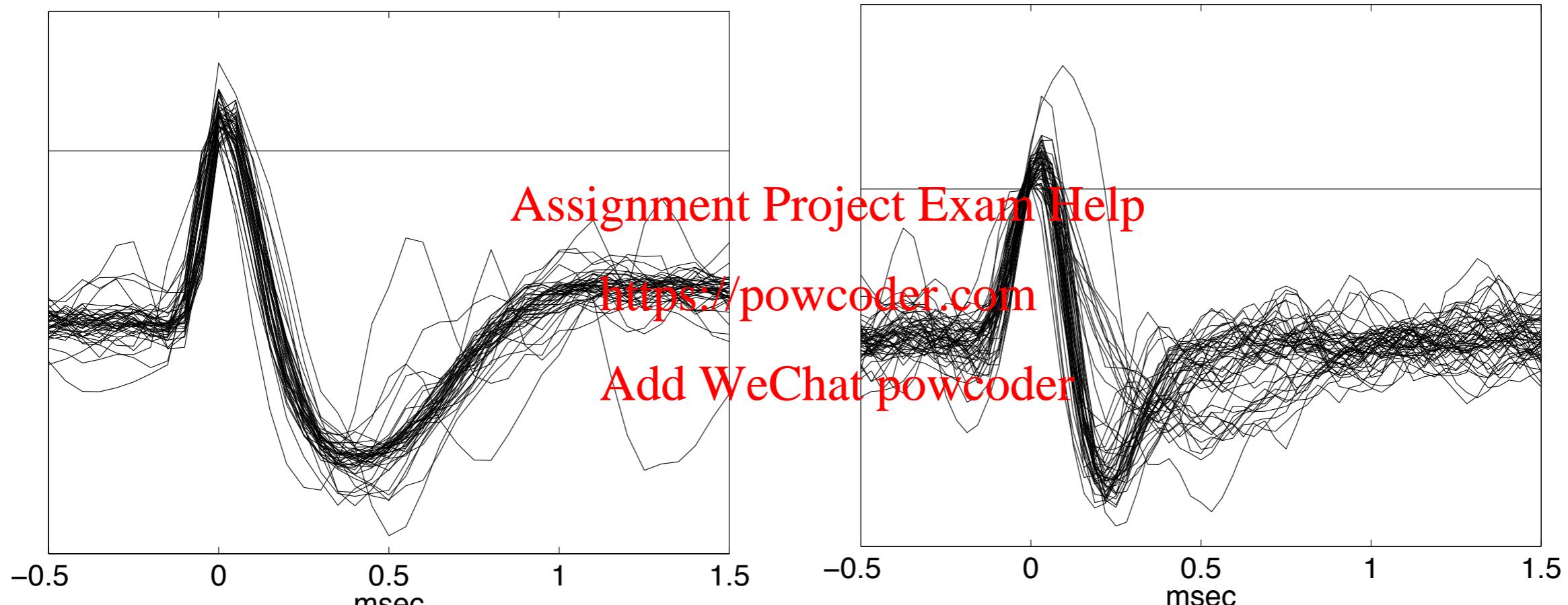
- Euclidean distance only cares about how many pixels overlap.
- Could try to define a distance metric that is insensitive to small deviations in position, scale, rotation, etc.
- Digit example:
  - 60,000 training images,
  - 10,000 test images
  - no “preprocessing”

# A real example: clustering electrical signals from neurons



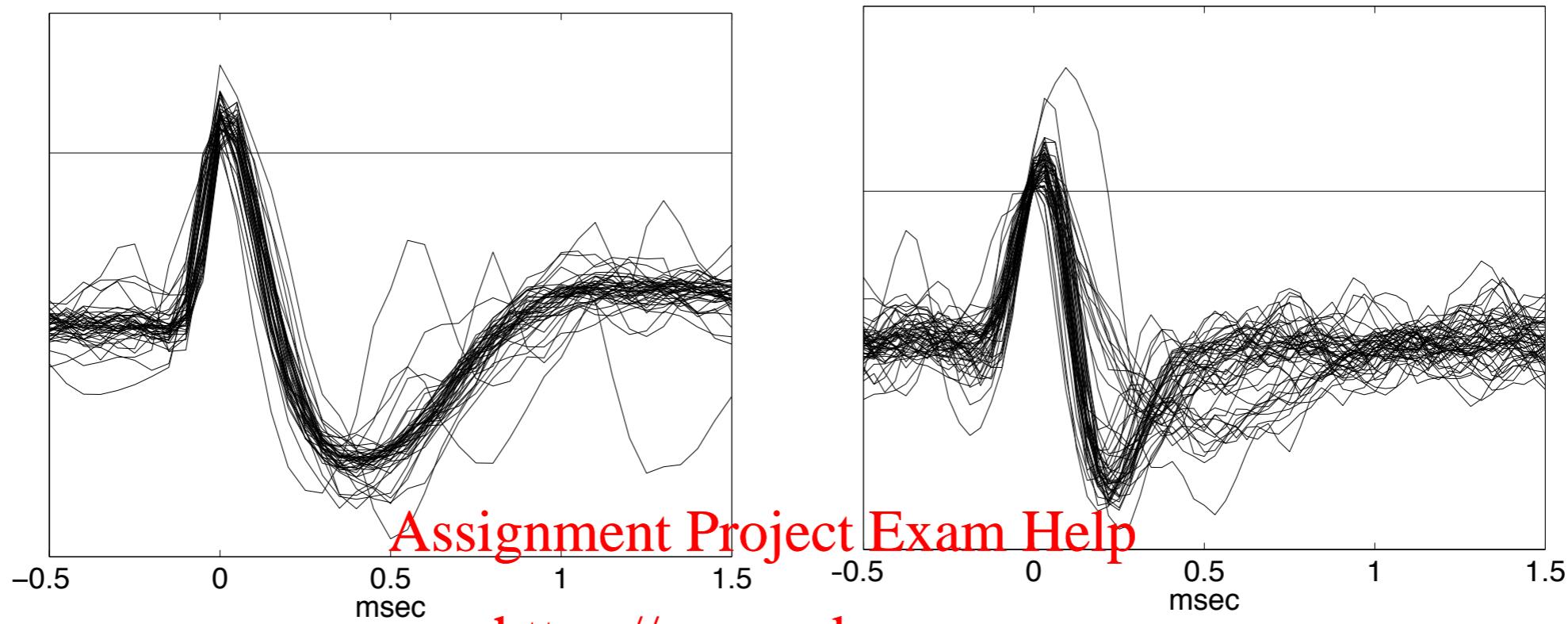
Basic problem: only information is signal.  
The true classes are *always* unknown.

# Sorting with level detection on an oscilloscope



Level detection doesn't always work.

# Why level detection doesn't work



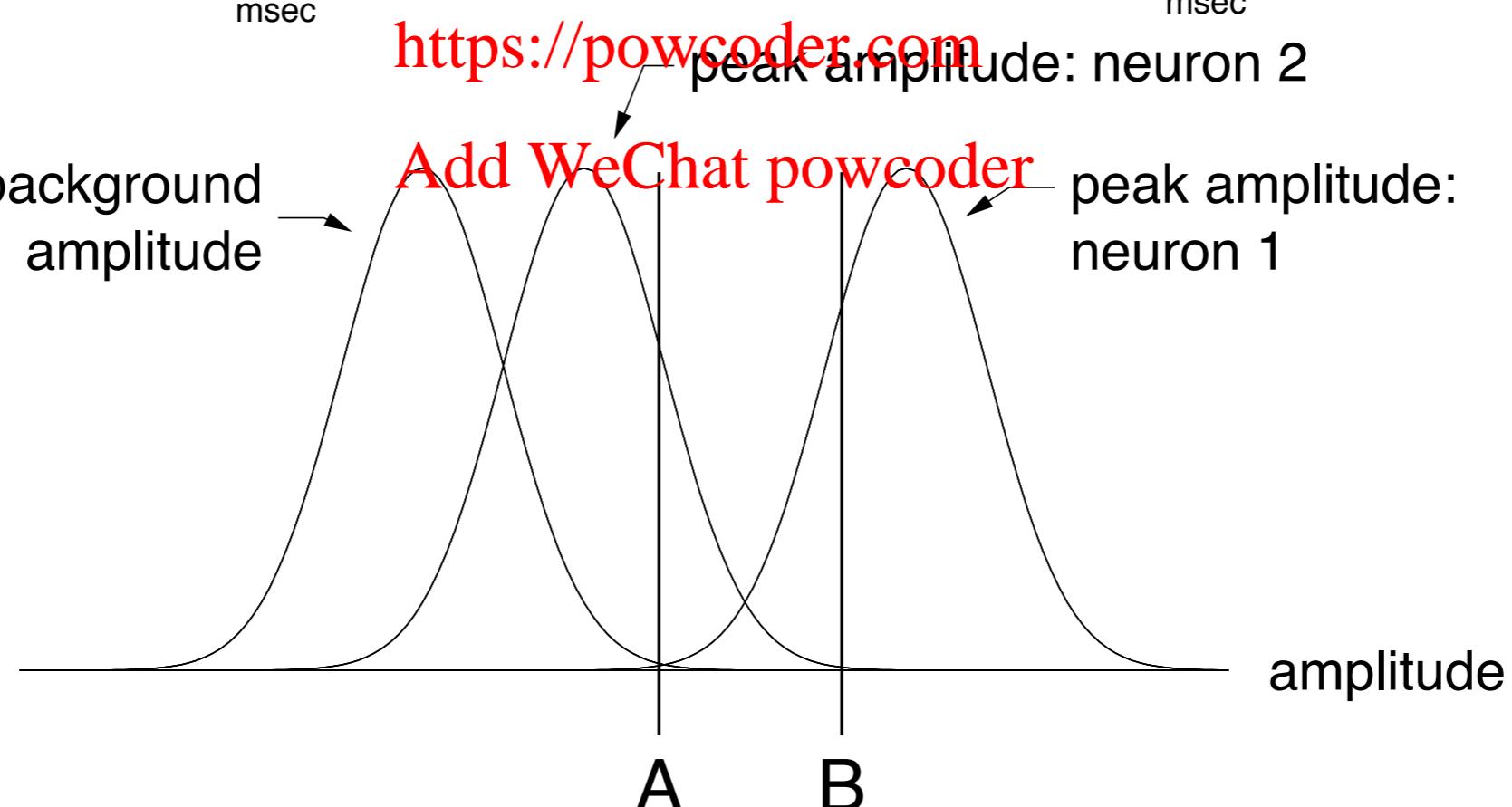
Assignment Project Exam Help

<https://powcoder.com>  
peak amplitude: neuron 2

background  
amplitude

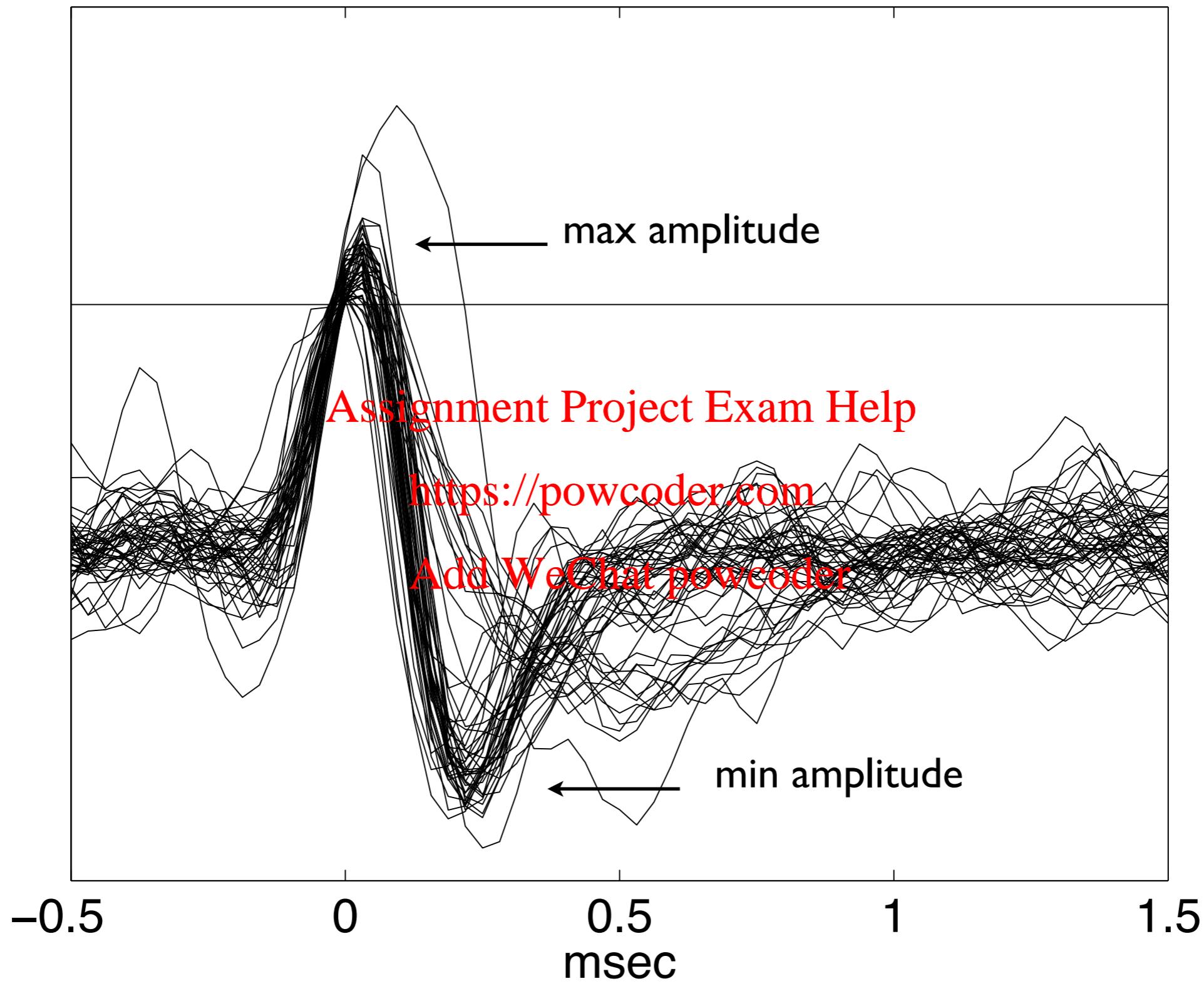
Add WeChat powcoder

peak amplitude:  
neuron 1

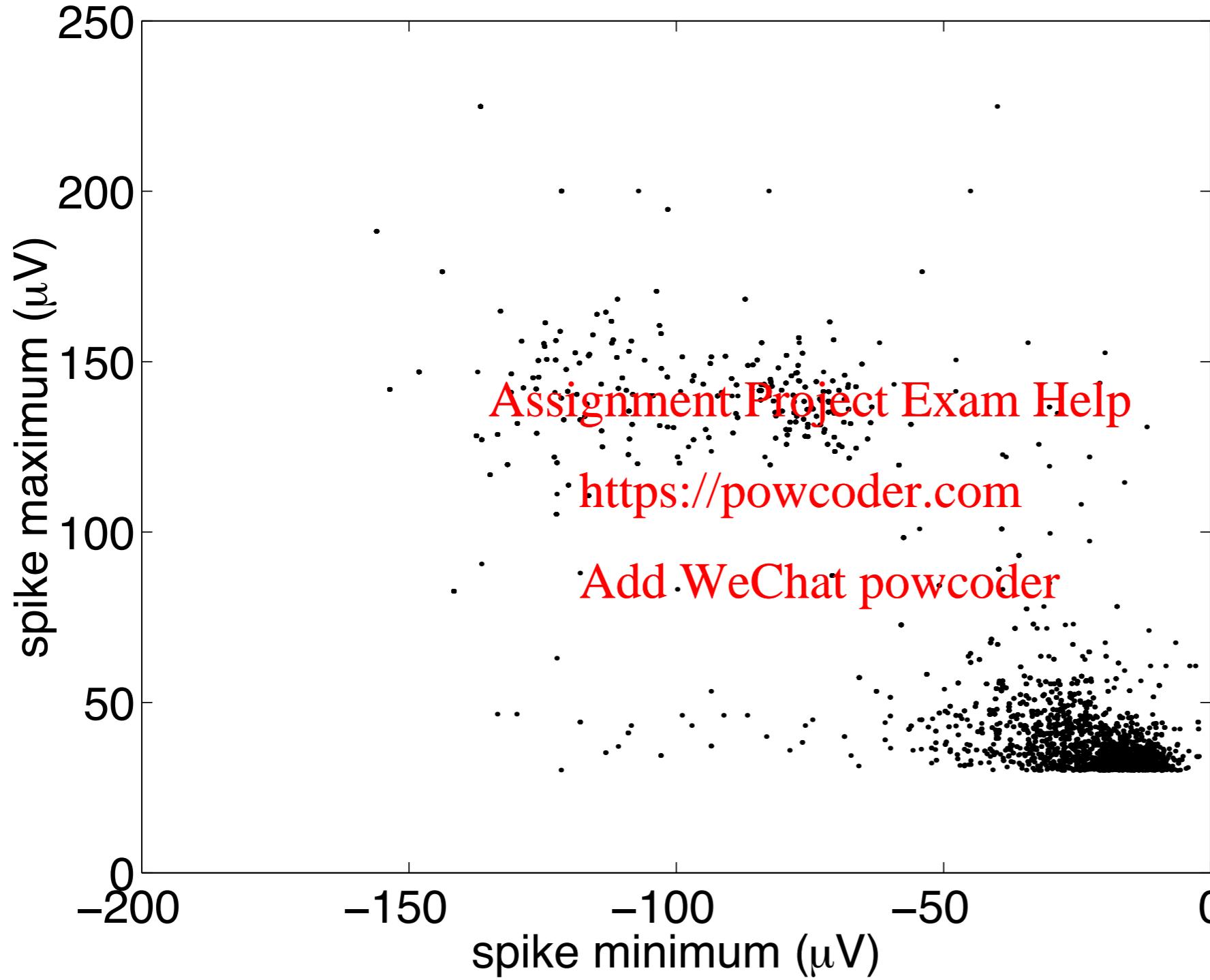


One dimension is not sufficient to separate the spikes.

# Idea: try more features

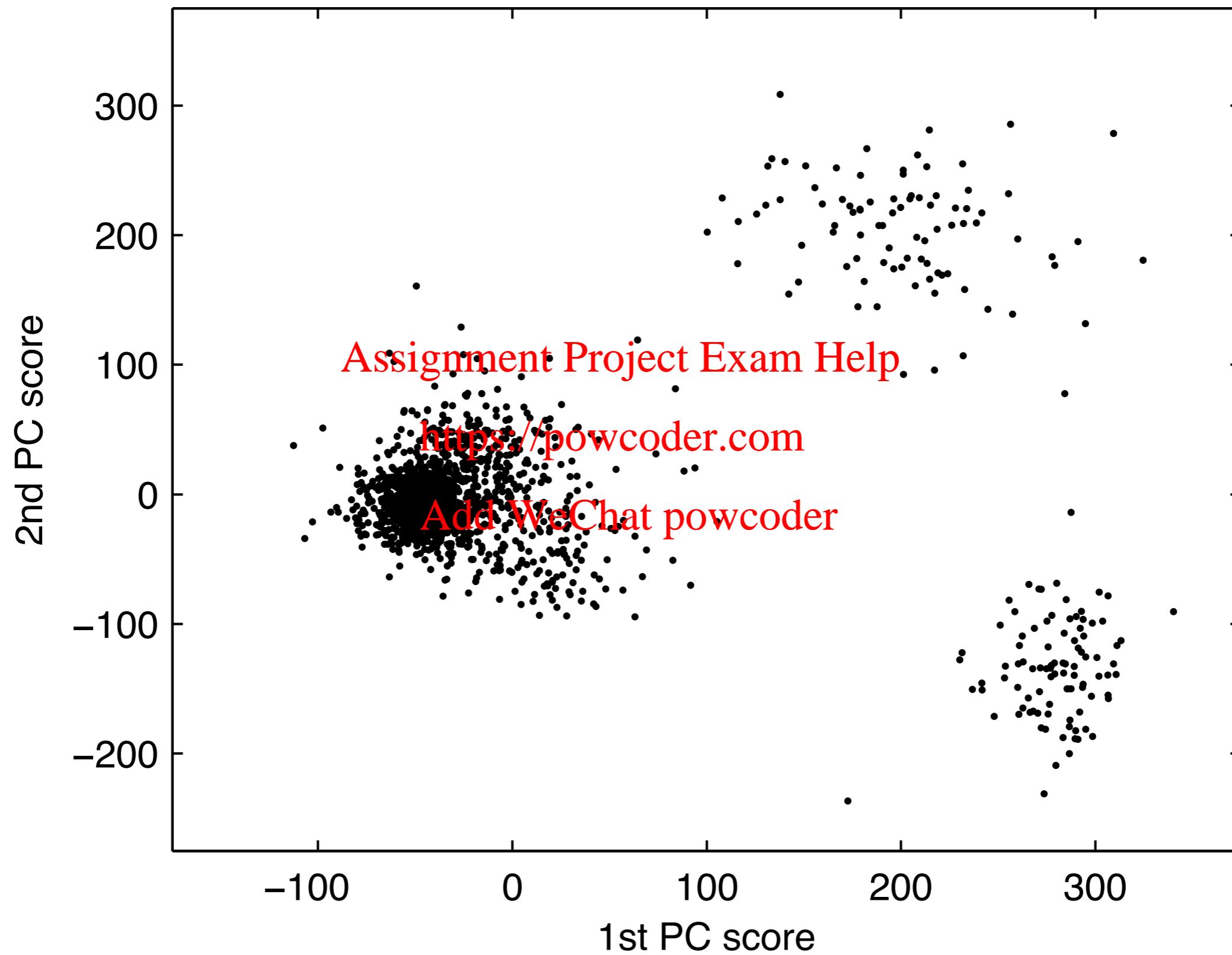


## Maximum vs minimum



This allows better discrimination than max alone, but is it optimal?

# Features using Principal Components (not covered)



# k-means clustering

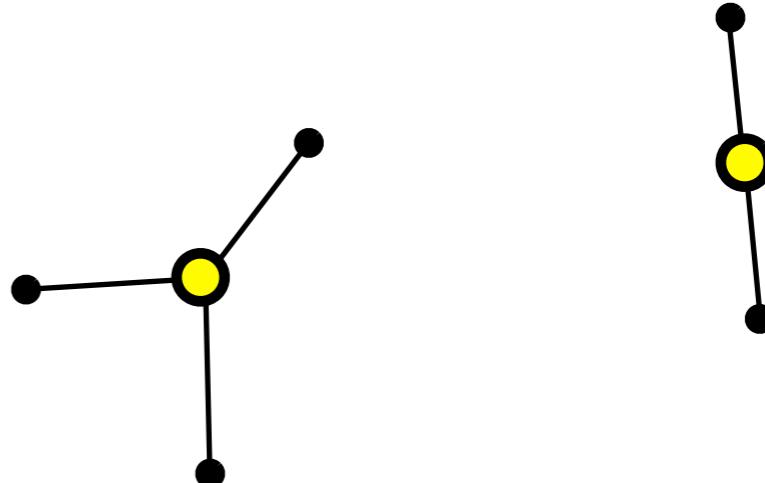
- Idea: try to estimate k cluster centers by minimizing “distortion”
- Define distortion as:

$$D = \sum_{n=1}^N \sum_{k=1}^K r_{nk} \| \mathbf{x}_n - \boldsymbol{\mu}_k \|^2$$

$r_{nk}$  = 1 if  $\mathbf{x}_n \in$  Assignment Project Exam Help  
cluster  $k$ , 0 otherwise.

<https://powcoder.com>

- $r_{nk}$  is 1 for the closest cluster mean to  $\mathbf{x}_n$ .  
[Add WeChat powcoder](#)
- Each point  $\mathbf{x}_n$  is the minimum distance from its closet center.
- How do we learn the cluster means?
- Need to derive a **learning rule**.



# Deriving a learning rule for the cluster means

- Our objective function is:

$$D = \sum_{n=1}^N \sum_{k=1}^K r_{nk} \| \mathbf{x}_n - \boldsymbol{\mu}_k \|^2$$

- Here, we can solve for the mean:

$$\boldsymbol{\mu}_k = \frac{\sum_n r_{nk} \mathbf{x}_n}{\sum_n r_{nk}}$$

- Differentiate w.r.t. to the mean (the parameter we want to estimate):

Assignment Project Exam Help

$$\frac{\partial D}{\partial \boldsymbol{\mu}_k} = 2 \sum_{n=1}^N r_{nk} (\mathbf{x}_n - \boldsymbol{\mu}_k)$$

<https://powcoder.com>  
Add WeChat powcoder

- This is simply a weighted mean for each cluster.

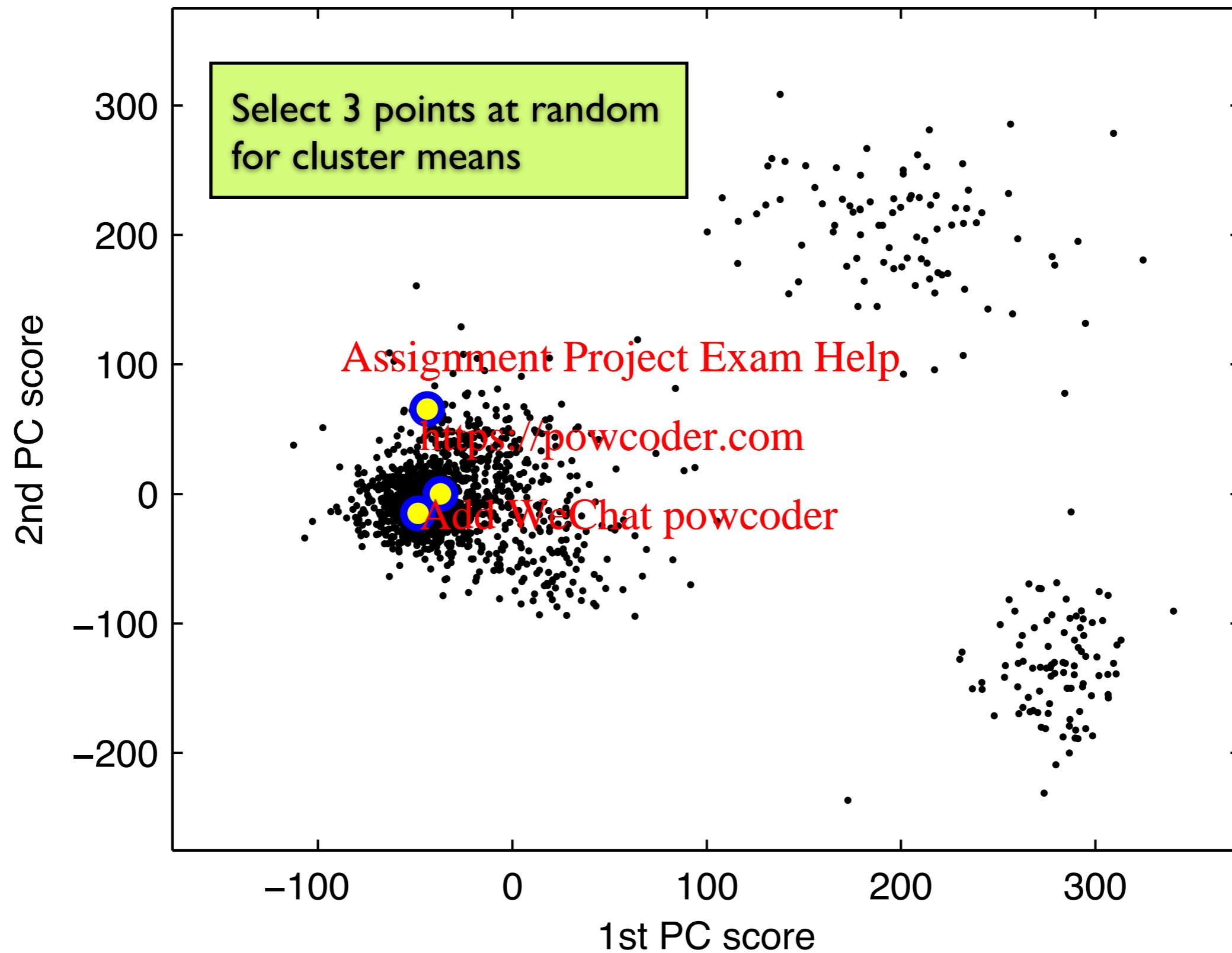
• Thus we have a simple estimation algorithm (*k-means clustering*)

- We know the optimum is when

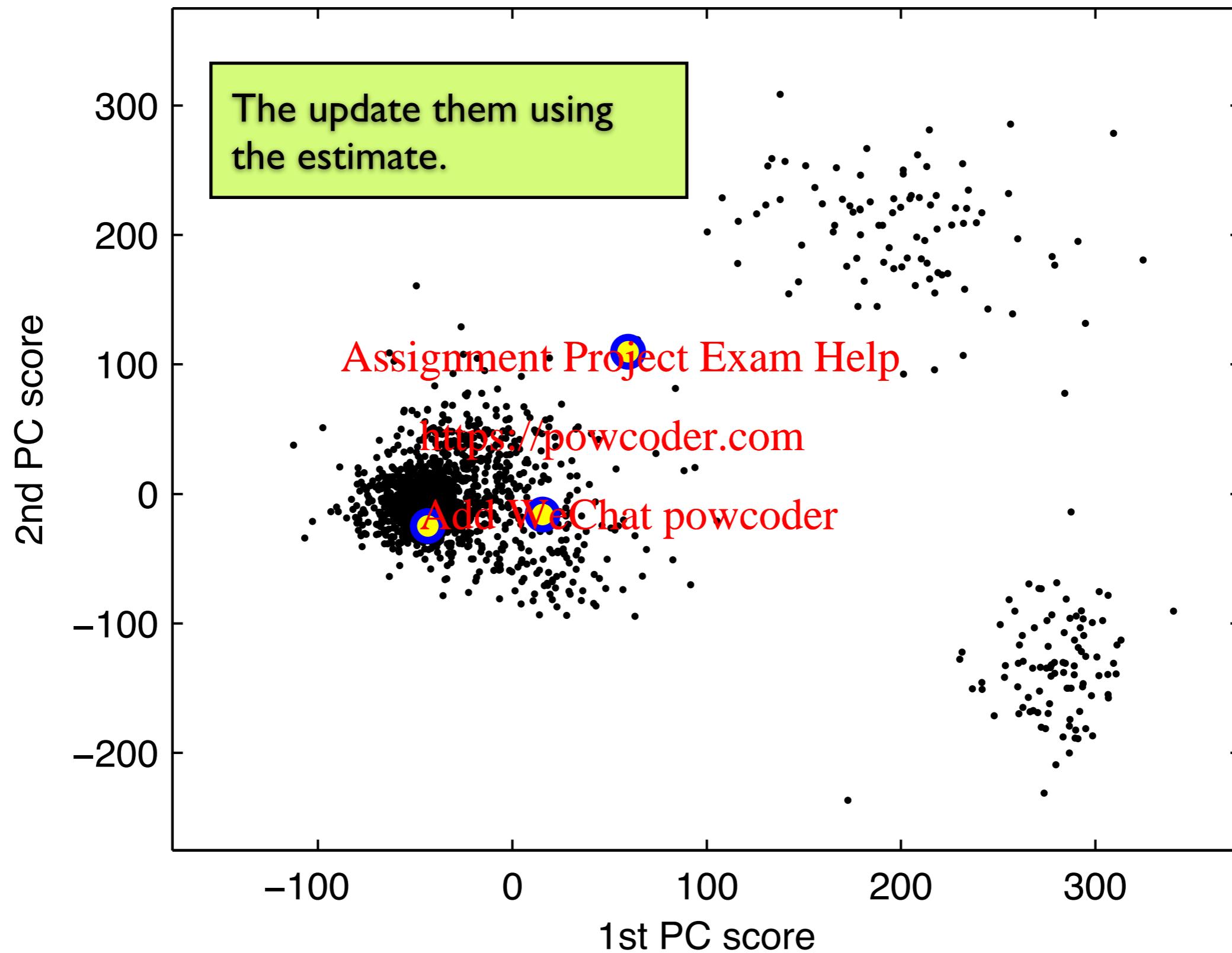
$$\frac{\partial D}{\partial \boldsymbol{\mu}_k} = 2 \sum_{n=1}^N r_{nk} (\mathbf{x}_n - \boldsymbol{\mu}_k) = 0$$

1. select k points at random
  2. estimate (update) means
  3. repeat until converged
- convergence (to a local minimum) is guaranteed

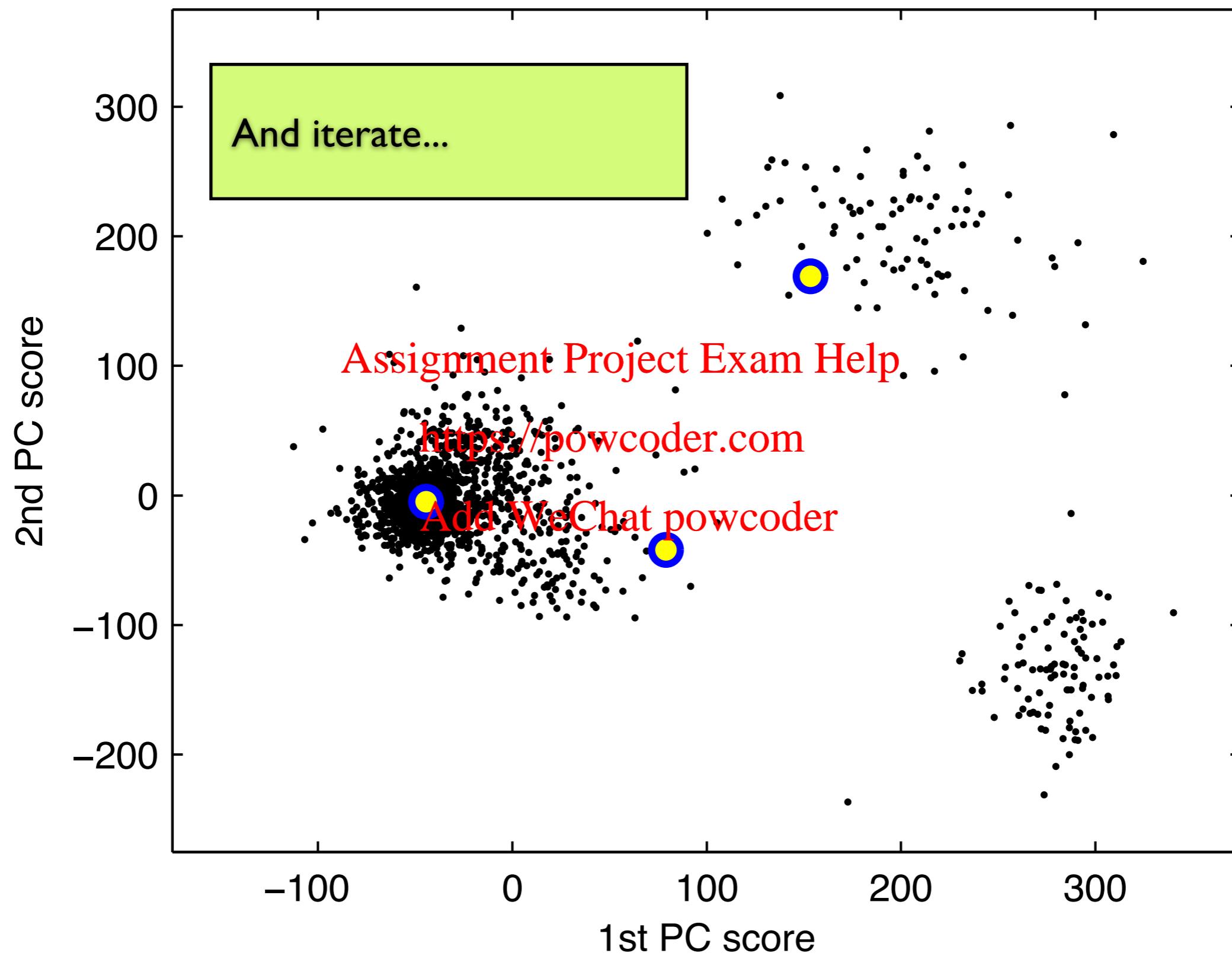
# k-means clustering example



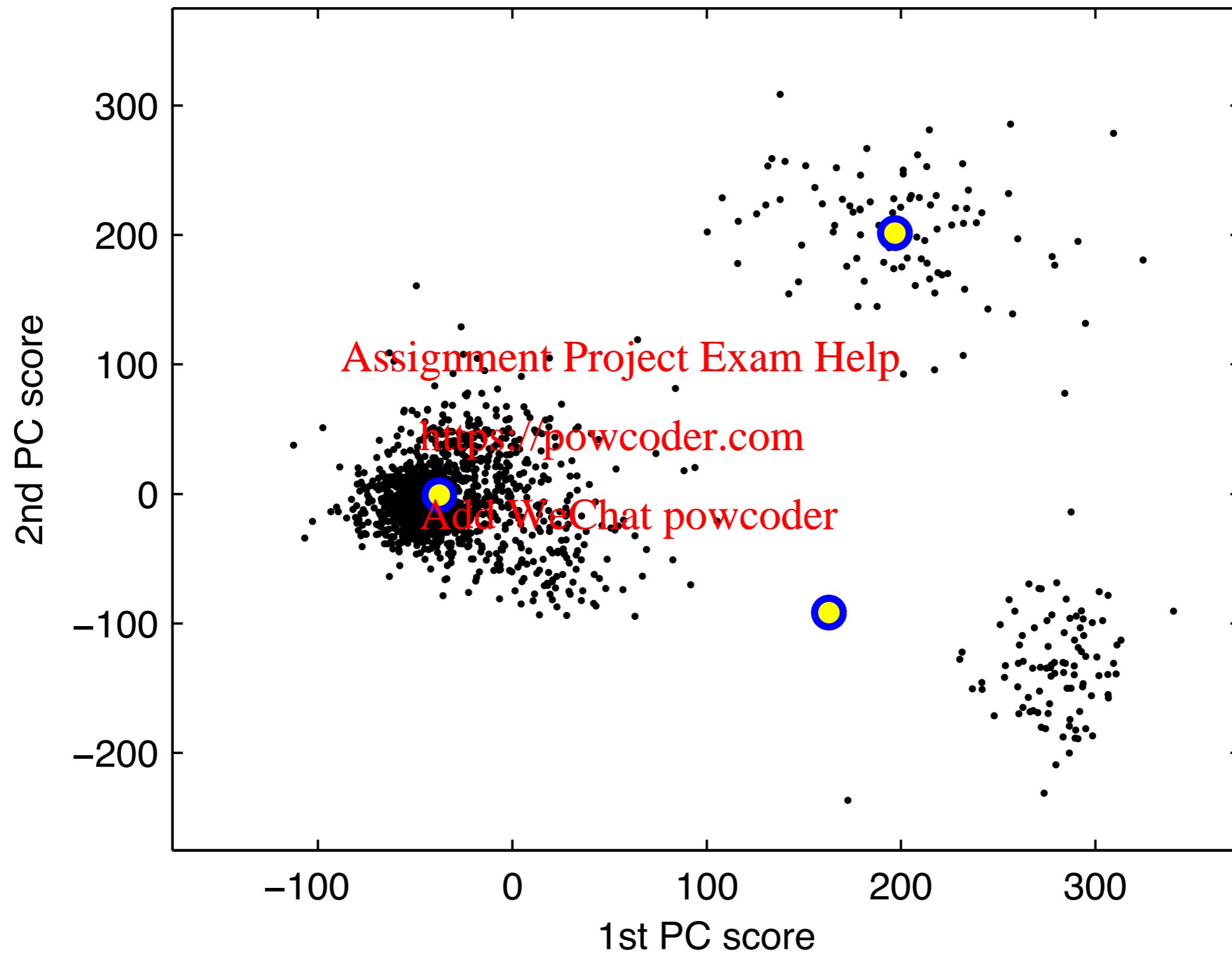
# k-means clustering example



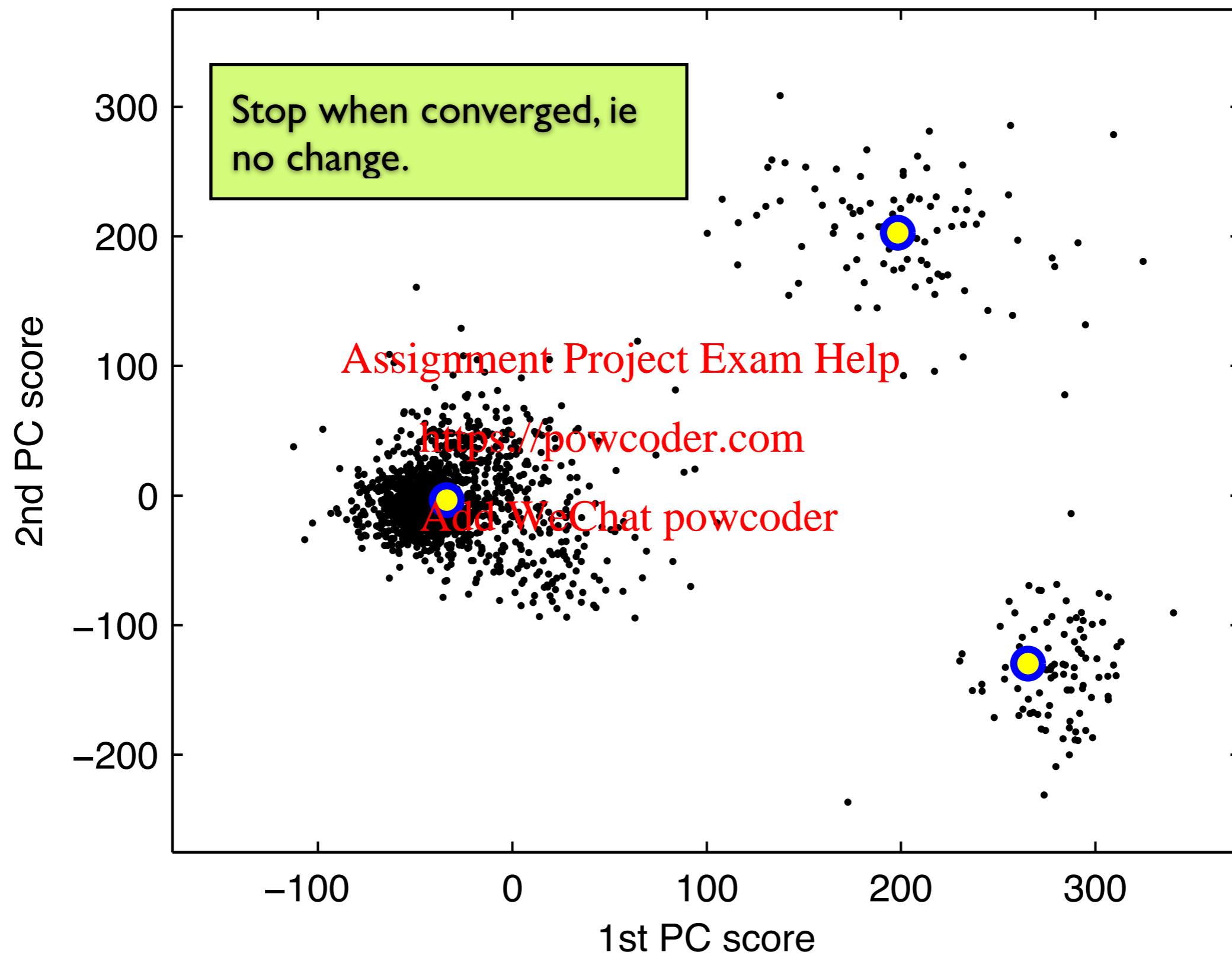
# k-means clustering example



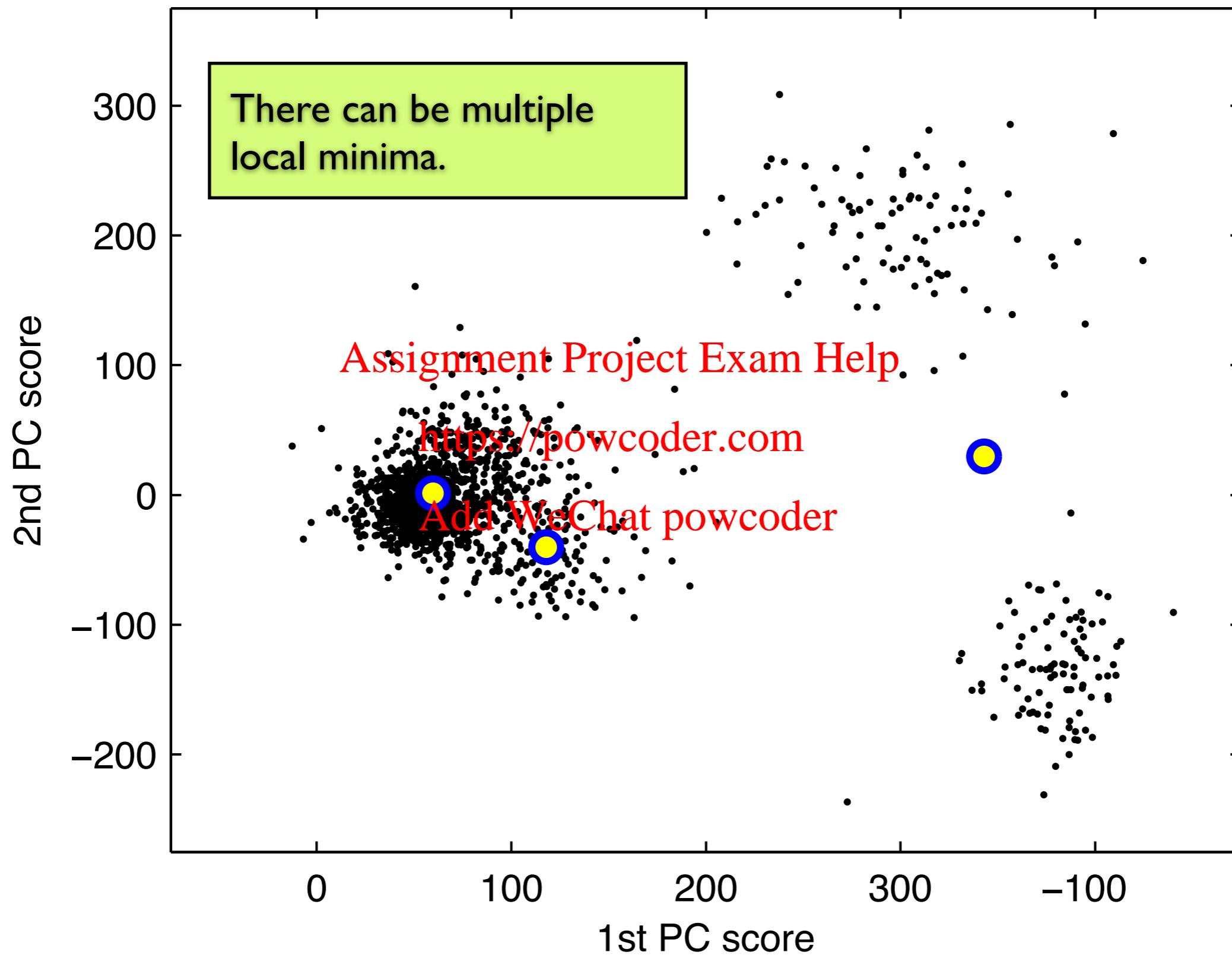
# k-means clustering example



# k-means clustering example

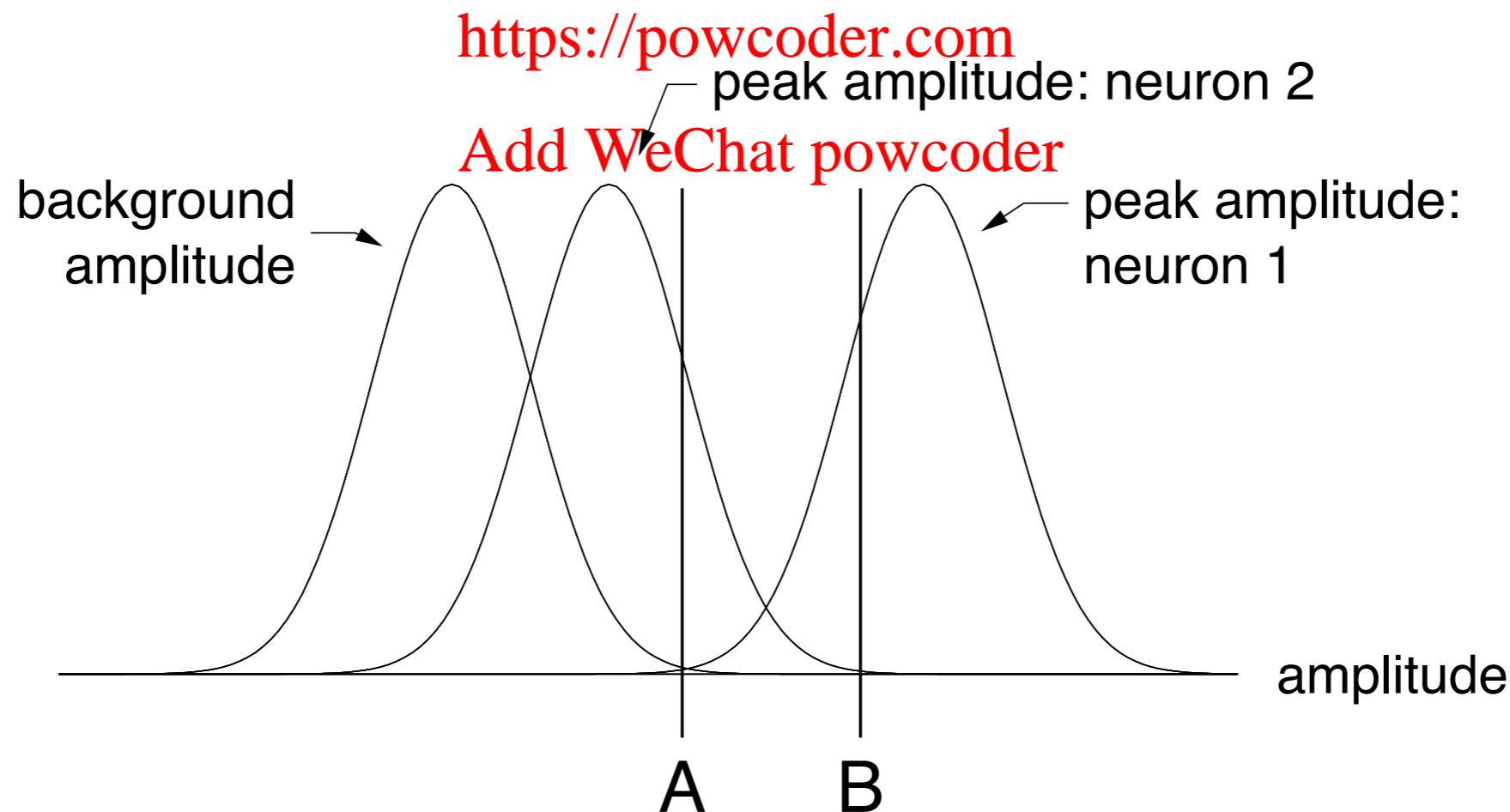


# An example of a local minimum



# A probabilistic interpretation: Gaussian mixture models

- We've already seen a one-dimensional version
- This example has three classes: neuron 1, neuron 2, and background noise.
- Each can be modeled as a Gaussian
- Any given data point comes from just one Gaussian
- The whole set of data is modeled by a *mixture* of three Gaussians
- How do we model this?



# The Gaussian mixture model density

- The likelihood of the data given a particular class  $c_k$  is given by

$$p(x|c_k, \mu_k, \Sigma_k)$$

- $x$  is the spike waveform,  $\mu_k$  and  $\Sigma_k$  are the mean and covariance for class  $c_k$ .

- The marginal likelihood is computed by summing over the likelihood of the  $K$  classes  
<https://powcoder.com>

$$p(x|\theta_{1:K}) = \sum_{k=1}^K p(x|c_k, \theta_k) p(c_k)$$

- $\theta_{1:K}$  defines the parameters for all of the classes,  $\theta_{1:K} = \{\mu_1, \Sigma_1, \dots, \mu_K, \Sigma_K\}$ .
- $p(c_k)$  is the probability of the  $k$ th class, with  $\sum_k p(c_k) = 1$ .
- What does this mean in this example?

# Bayesian classification with multivariate Gaussian mixtures

- How do we determine the class  $c_k$  from the data  $x$  ?
- Again use Bayes' rule

$$p(c_k|x^{(n)}, \theta_{1:K}) = p_{k,n} = \frac{p(x^{(n)}|c_k, \theta_k)p(c_k)}{\sum_k p(x^{(n)}|c_k, \theta_k)p(c_k)}$$

Assignment Project Exam Help

- This tells us the probability that waveform  $x^{(n)}$  came from class  $c_k$ .

Add WeChat powcoder

# Estimating the parameters: fitting the model density to the data

- The objective of density estimation is to maximize the likelihood of the data
- If we assume the samples are independent, the data likelihood is just the product of the *marginal* likelihoods

$$p(x_{1:N}|\theta_{1:K}) = \prod_{n=1}^N p(x_n|\theta_{1:K})$$

<https://powcoder.com>

- The class parameters are determined by optimization.  
[Add WeChat powcoder](#)
- Is far more practical to optimize the log-likelihood.
- One elegant approach to this is the EM algorithm.

# The Gaussian mixture

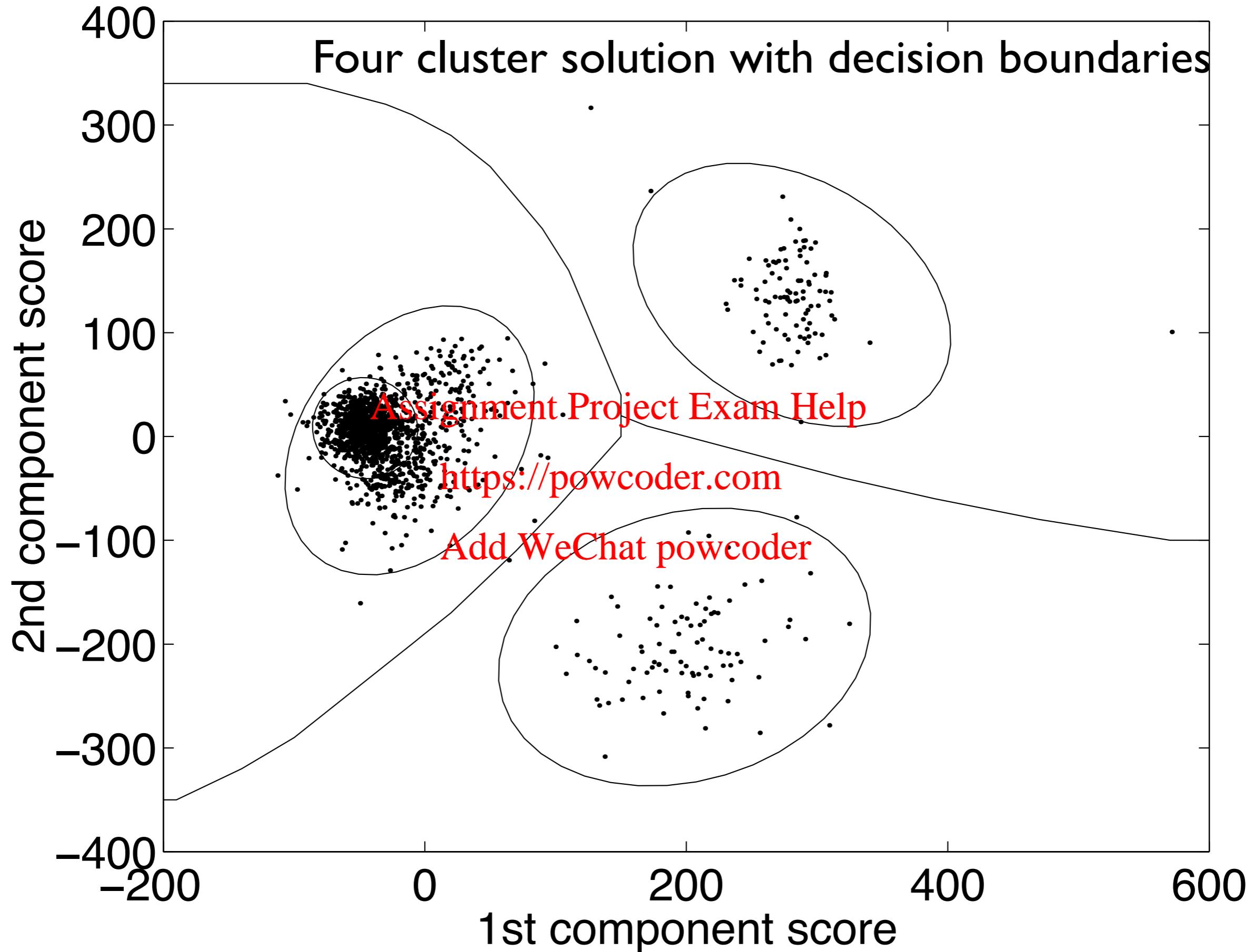
- EM stands for Expectation-Maximization, and involves two steps that are iterated. For the case of a Gaussian mixture model:
  1. E-step: Compute  $p_{n,k} = p(c_k|x^{(n)}, \theta_{1:K})$ . Let  $p_k = \sum_n p_{i,n}$
  2. M-step: Compute new mean, covariance, and class prior for each class:

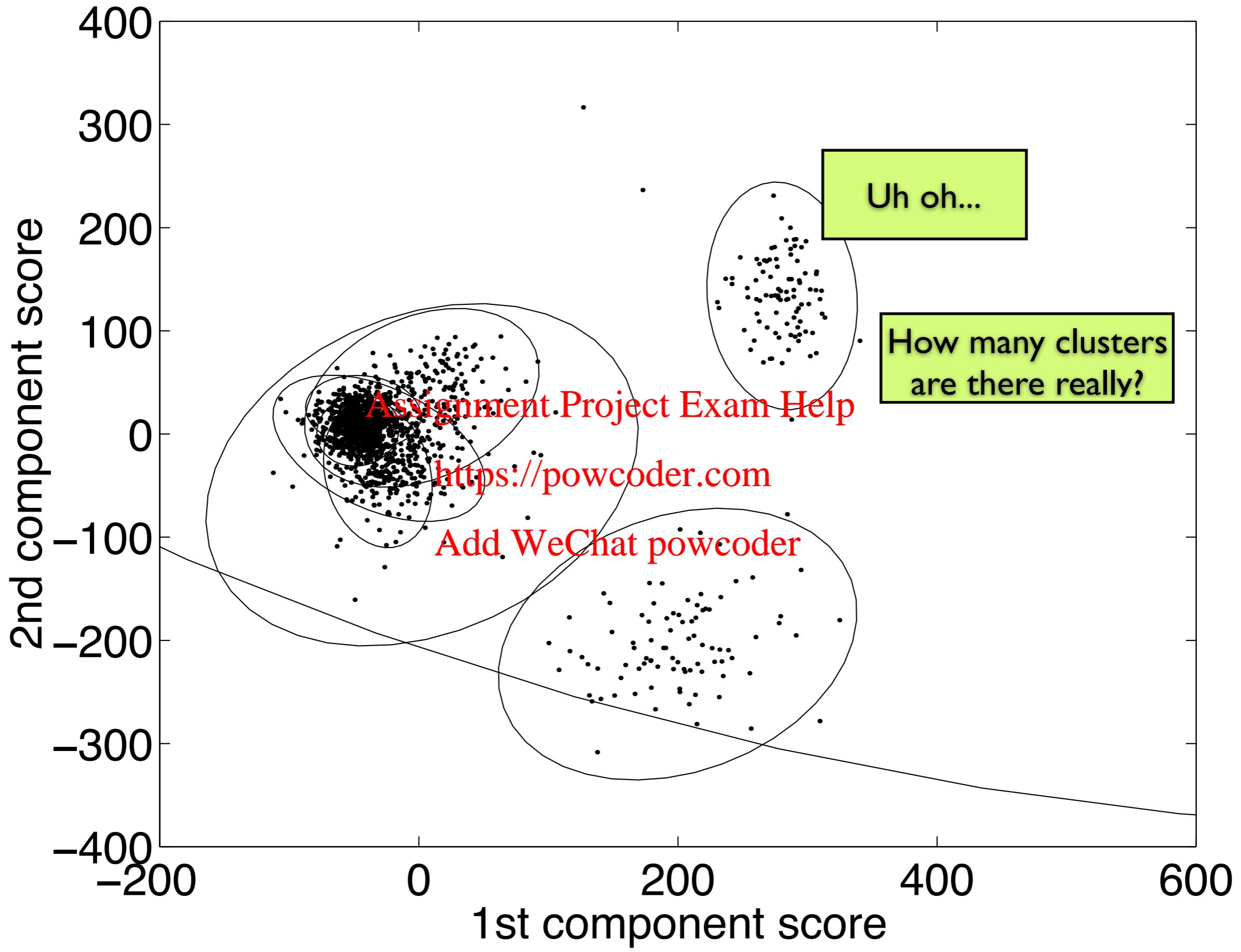
$$\mu_k \leftarrow \sum_n p_{n,k} x^{(n)} / p_k$$

$$\Sigma_k \leftarrow \sum_n p_{n,k} (x^{(n)} - \mu_k)^T / p_k$$

$$p(c_k) \leftarrow p_k$$

- This is just the sample mean and covariance, weighted by the class conditional probabilities  $p_{n,k}$ .
- Derived by solving setting log-likelihood gradient to zero (i.e. the maximum).





# How do we choose k?

- Increasing k, will always decrease our distortion. This will **overfit** the data.
  - How can we avoid this?
  - Or how do we choose the best k?
- One way: **cross validation**
- Use our distortion metric:

$$D = \sum_{n=1}^N \sum_{k=1}^K r_{nk} \| \mathbf{x}_n - \boldsymbol{\mu}_k \|^2$$

<https://powcoder.com>

Add WeChat powcoder

- Then just measure the distortion on a *test data set*, and stop when we reach a minimum.

