Review Session

Examples: How I would study

What You Should Know Afterward

You should know the definitions of the following, and be able to use them to solve problems:

- Random variables and events
- The Axioms of Probability
- Independence, binomials, multinomials
- Expectation and variance of a distribution
- Conditional probabilities
- Bayes Rule
- MLE's, smoothing, and MAPs
- The joint distribution
- How to do inference using the joint distribution
- Density estimation and classification

Can you explain these to a friend?
Can you find them in your notes?

Can you find a problem in the sample midterms you can't do or don't understand?

Why are these important later on?

Examples: How I would study

- Remember your test taking skills
  - Skim all the questions first and plan your strategy
  - Go with what you know – skip hard/long questions
  - If you make an assumption write it down
  - If you're not sure ask
- Open-book open-notes but no web access/ laptops
  - Don't assume you'll have time to do research
  - You should be familiar with the terms and important concepts

Examples: How I would study

)-601 Decision Trees

s a lecture used in the Syllabus for Machine Learning 10-601 in Fall 2014

ides

[ec

Ziv's lecture: Slides in pdf

advantages and disadvantages of decision-tree learning in specific, and eager learning in general,

eadings

[ec

How about pros/cons of DTs vs naïve Bayes? Or DTs vs logistic regression?

Mitchell, Chapter 3.

hat You Should Know Afterward

[ec

What a decision tree is, and how to classify an instance using a decision tree.

What the canonical top-down algorithm is for learning a decision tree.

- What heuristics are used for choosing a decision-tree split.
- What entropy is, and what information gain is.

What reduced-error pruning is, and why it might improve classification performance.

Some of the advantages and disadvantages of decision-tree learning in specific, and eager learning in general, compared to K NN learning.

What reduced-error pruning is, and why it might improve classification performance.

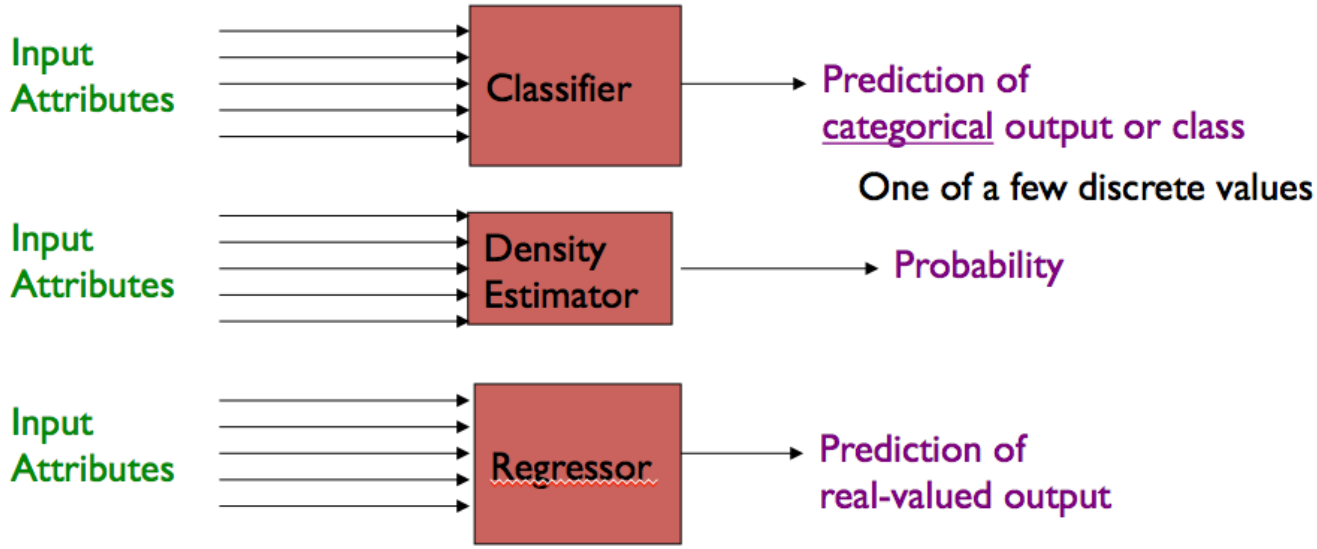
Good question: Does it change bias/variance to do REP?

More Training Data

+1

Density Estimation – looking ahead

- Compare it against the two other major kinds of models:



More Training Data

Decision tree learning

-1

Google scholar decision tree quinlan Search

Scholar Articles and patents anytime include citations Create email alert

Induction of **decision** trees
JR **Quinlan** - Machine learning, 1986 - Springer
... Several studies have been carried out to see how this modified procedure holds up under varying levels of noise (**Quinlan** 1983b, 1985a ... each experiment, the whole set of objects was artificially corrupted as described below and used as a training set to produce a **decision tree**. ...
Cited by 10584 - Related articles - Library Search - All 75 versions

Cited by 13602

More Training Data

-1

```
bash-3.2$ ripper onion
Final hypothesis is:
fromOnion :- wordsInArticle ~ enlarge (173/0).
fromOnion :- wordsInArticle ~ monday (9/1).
fromOnion :- wordsInArticle ~ added, wordsInArticle ~ play (6/1).
fromOnion :- wordsInArticle ~ mocking (2/0).
fromOnion :- wordsInArticle ~ manhattan (2/1).
default fromEconomist (530/0).
================================ summary ================================
Train error rate:   0.41% +/- 0.24% (725 datapoints)    <<
Hypothesis size:    5 rules, 11 conditions
Learning time:      1.09 sec
```

Translation:

if "enlarge" is in the set-valued attribute wordsArticle
then class = fromOnion. *this rule is correct 173 times, and never wrong*

...

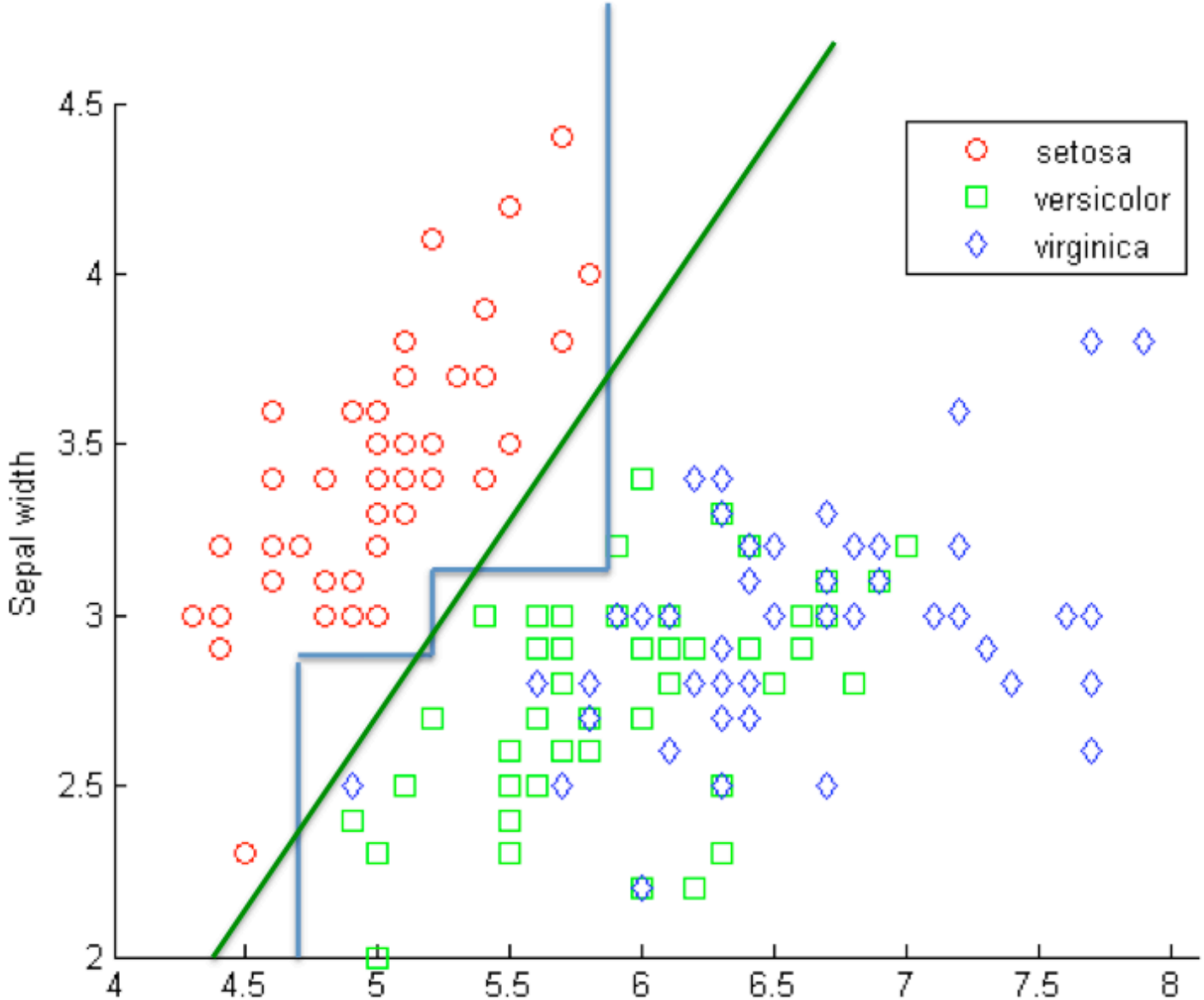
if "added" is in the set-valued attribute wordsArticle
and "play" is in the set-valued attribute wordsArticle
then class = fromOnion. *this rule is correct 6 times, and wrong once*

A Test Case

BACKGROUND: ENTROPY AND OPTIMAL CODES
Information theory
Problem: design an efficient coding scheme for leaf colors:
green
yellow
gold
red
orange
brown
$E[nbits] = \frac{1}{2}*1+\frac{1}{8}*3+\frac{1}{8}*3+\frac{1}{16}*4+\frac{1}{16}*4+\frac{1}{8}*3 = 2.125$
$Entropy(P) = H(P) = -\sum_x p(x)\lg_2 p(x)$
$H(P) = -\sum_x p(x)\lg_2 p(x) = -p\lg p-(1-p)\lg(1-p)$
DECISION TREE LEARNING: THE ALGORITHM(S)
+1/2
+1

More Training Data

+1



More training data

Would foo(D) be a good split criteria or not?

1. Given dataset D:
   - return *leaf(y)* if all examples are in the same class *y* … or nearly so…
   - pick the **best** split, on the best attribute *a*
     - $a=c_1$ or a
     - $a<\vartheta$ or a
     - a or *not*(a
     - a *in* $\{c_1,\ldots,c_k\}$ or not
   - split the data into D_1 ,D_2 , … D_k and recursively build trees for each subset
2. "Prune" the tree

What if I used sampling to estimate the best split?

When does this greedy search not work? (HWs)

Popular splitting criterion: try to lower *entropy* of the *y* labels on the resulting partition

i.e., prefer splits that have skewed distributions of labels

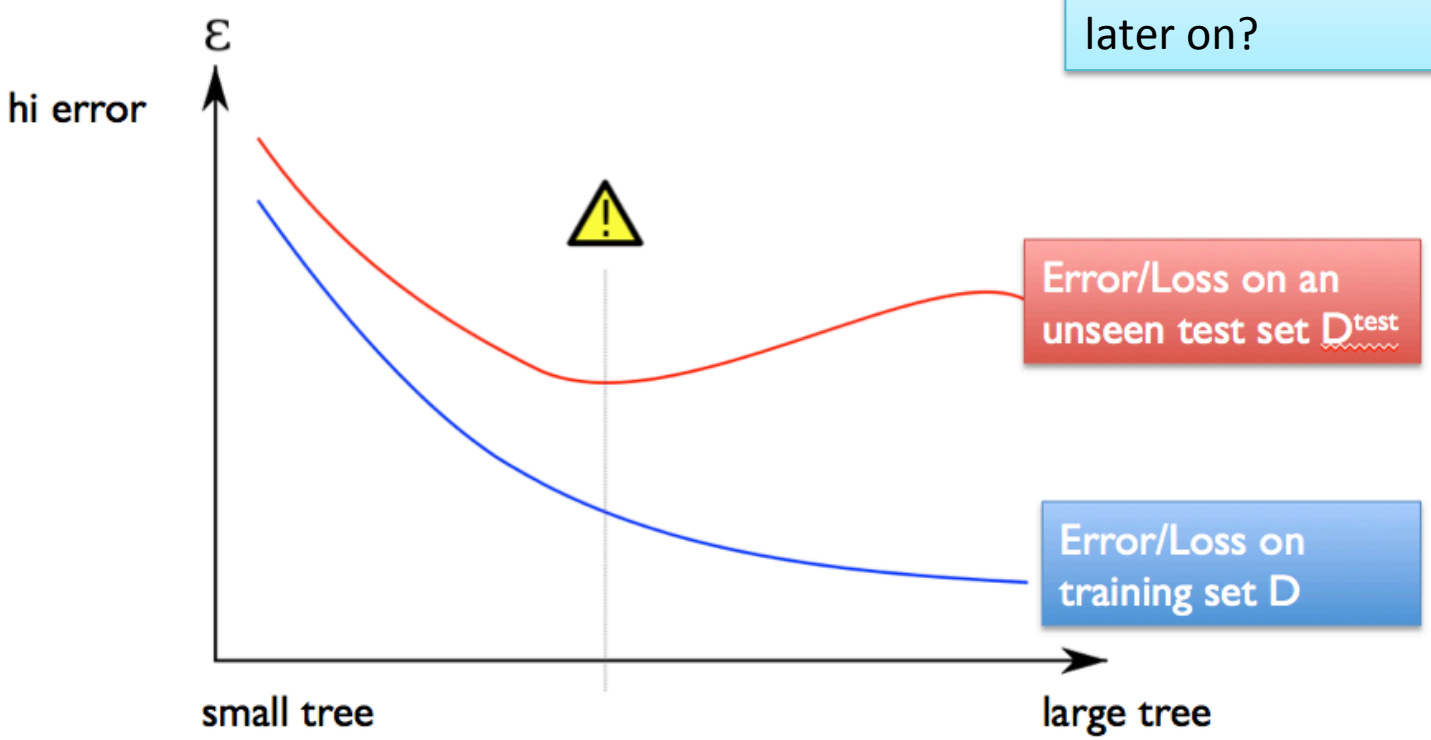
$$H(D) = \sum_k \Pr_D(Y = y_k) \log\left[\Pr_D(Y = y_k)\right]$$

More Training Data

+1

- Small tree ➔ a smooth decision boundary
- Large tree ➔ a complicated shape
- What's the best size decision tree?

How does this relate to concepts we've learned later on?



How to guess what's important

- If I say something twice, or come back to it in multiple lectures
- If it's on a HW
- If I take the time to "break something down" in detail
  - Know the algorithms we've discussed!
  - Could you find a bug? Complete an incomplete algorithm?
  - What do the pieces/parameters of the algorithm do? What are they called?
- If it helps compare/contrast/relate learning methods, or formal results
  - "learning as optimization"
  - "bias/variance and overfitting"
- If it demonstrates/gives a good intuition about an algorithm (when it work, how, why….)
  - Representational power of some H

[2pts] Suppose that in answering a question in a multiple choice test, an examinee either knows the answer, with probability p, or he guesses the answer with probability $1-p$. Assume that the probability of answering a question correctly is $1-\delta$ for an examine who knows the answer and $1/m$ for an examinee who guesses, where m is the number of multiple choice alternatives. What is the probability that an examinee knew the answer to a question given that he correctly answered it.

What is this testing? How do you solve it?

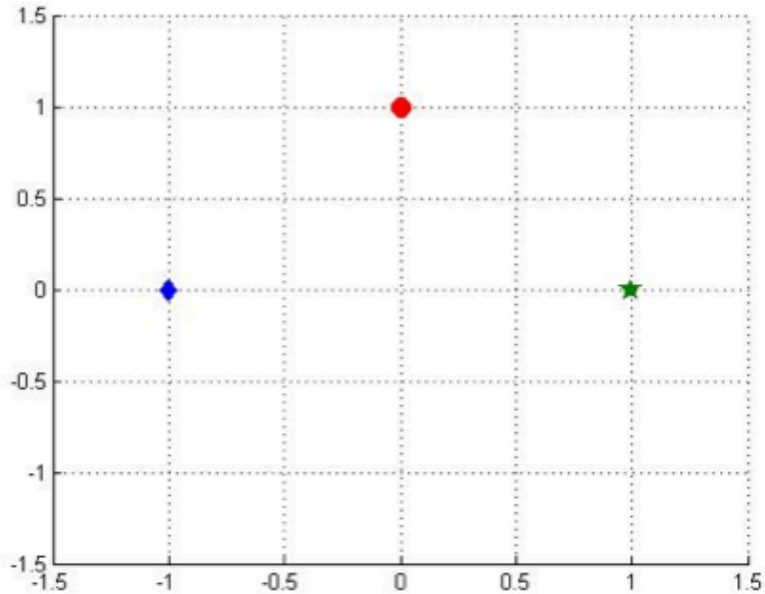


Figure 3: Points on two dimentional plane.

6. [2pts] For the data given in figure 3 draw the 1-NN decision boundary. Different shape represents Different classes.

What is this testing?

3. [2pts] Suppose multinomial naive Bayes gets zero *training* error on a binary dataset D. Does that imply that D is linearly separable? Justify your answer.

What is this testing? What sort of connections?

This is related to two themes we've repeated:

- learning and optimization – how do you rederive an optimization strategy?
- regularization and bias/variance – what does regularization do?

A common regularization term for logistic regression is

$$\mu \sum_{j=1}^{d} (w_j)^2$$

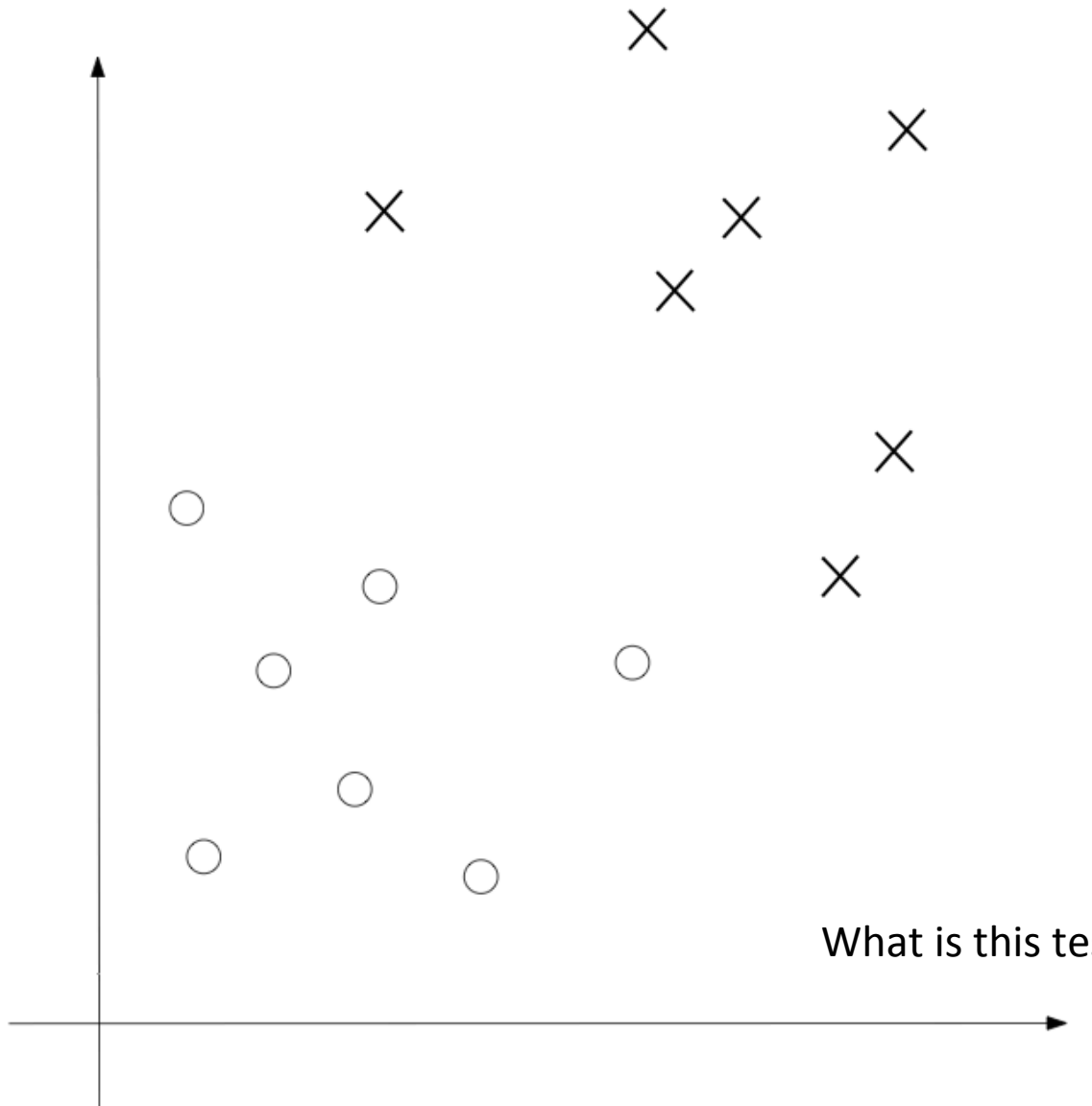
where d is the number of dimensions of the data, and w_j is the weight for the j-th parameter.

(a) [2pts] In Kevin's experiments with logistic regression, he used cross-validation to pick the best value of μ among these values: -1, -0.5, -0.1, 0, 0.1, 0.5, 1. He asks your advice about this selection—which set of values will you ask him to use among his suggested values and why?

Another good question: fix mu, vary k.
Does higher k have higher BIAS or VARIANCE?

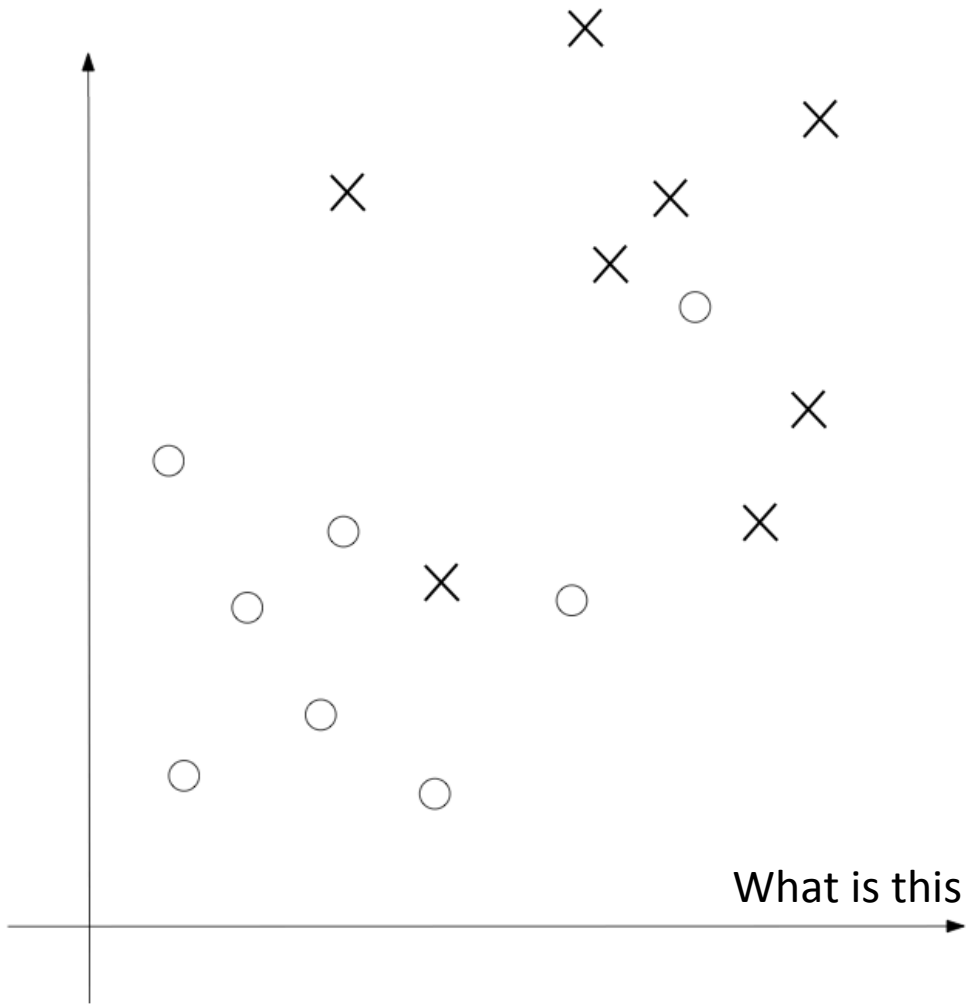
(b) [2pts] Kevin wants to explore using a regularization term of the form $\mu \sum_{j=1}^{d} (w_j)^k$ for $k = 1, 2, 3, 4$. He asks your advice on this proposal—what do you say?

. [2pts] Given the data points shown in the following figure with two different labels. Draw the linear SVM decision boundary for this binary classification problem. Also circle the support vectors (data points) for the boundary you find out



What is this testing?

3. [2pts] Draw the SVM decision boundary for another set of data points on the following figu and also explain the reason for achieving such a boundary. Please simply describe any ot additional conditions or constraints you want to use, for example slack variable.



What is this testing?

5. [2pts] You have implemented stacking using 5 base classifiers and 10-fold cross-validation. One of the five base classifiers is used as the final meta-classifier (the "top" of the stack). If each of your classifiers takes about the same time T to train on your dataset, and time t to evaluate a single example, how long does it to train the stacked learner? How long does it take to evaluate an example with the learned classifier? Assume nothing has been parallelized.

Do you understand the algorithm?