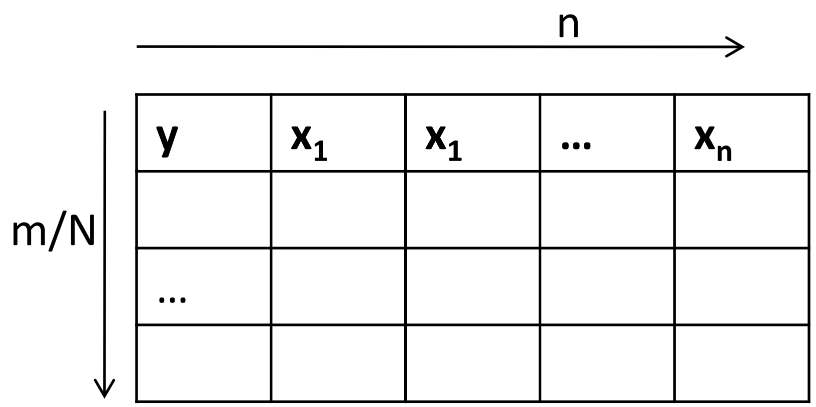
Mid-term overview

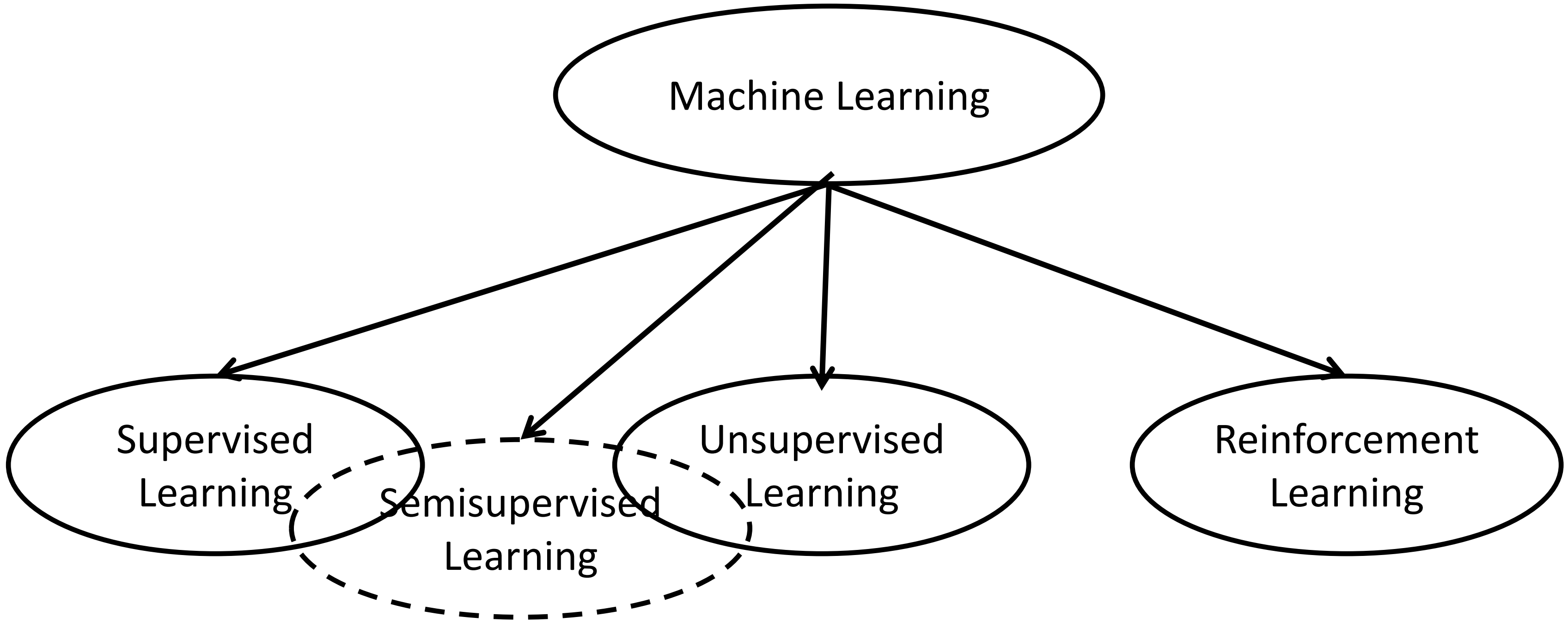
Machine Learning

- Data is i.i.d
- Goal is to learn a function f that maps **x** to y
- Data is generated using an unknown function *f*
- Learn a function *h* that minimizes some notion of distance/error wr.t *f*
- New test example is classified using the learned *h*



What are these? *training examples, features, classes, hypotheses, hypothesis classes, loss functions, adjustable parameters, VC dimension*

Machine Learning - Classification



So far: Supervised learning

Given a data set, learn a function h and predict on unknown test example

Key assumption: All the data points are labeled

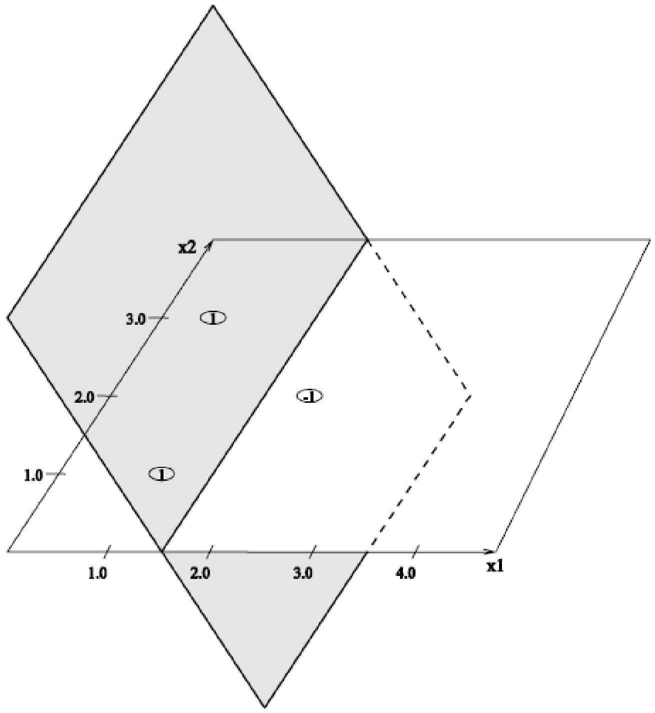
Labels can be <0,1> or <0,1,2..k> or <0,∞> or <red,blue,…> or <low,med, high>

Machine Learning – Classification II

- Linear vs Non-linear models
  - Linear models learn a threshold function

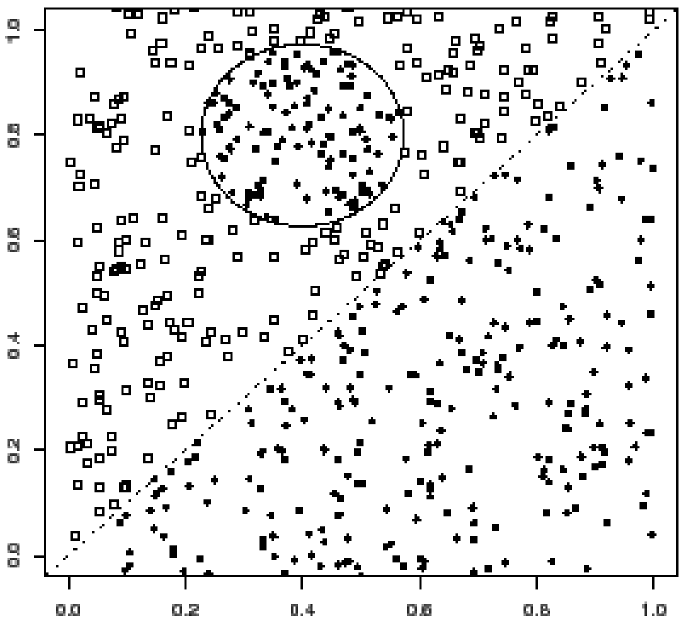
$$h(x) = \begin{cases} +1 \; if \;\; w_1 x_1 + \ldots + w_n x_n \geq 0 \\ -1 \qquad\qquad\qquad\quad otherwise \end{cases}$$

  - Problem description – Given features **x** and y, learn weights ***w***
  - Perceptron, Logistic Regression, Linear Discriminant Analysis, SVMs (with no kernels)
  - Advantages and disdvantages?



Machine Learning – Classification II

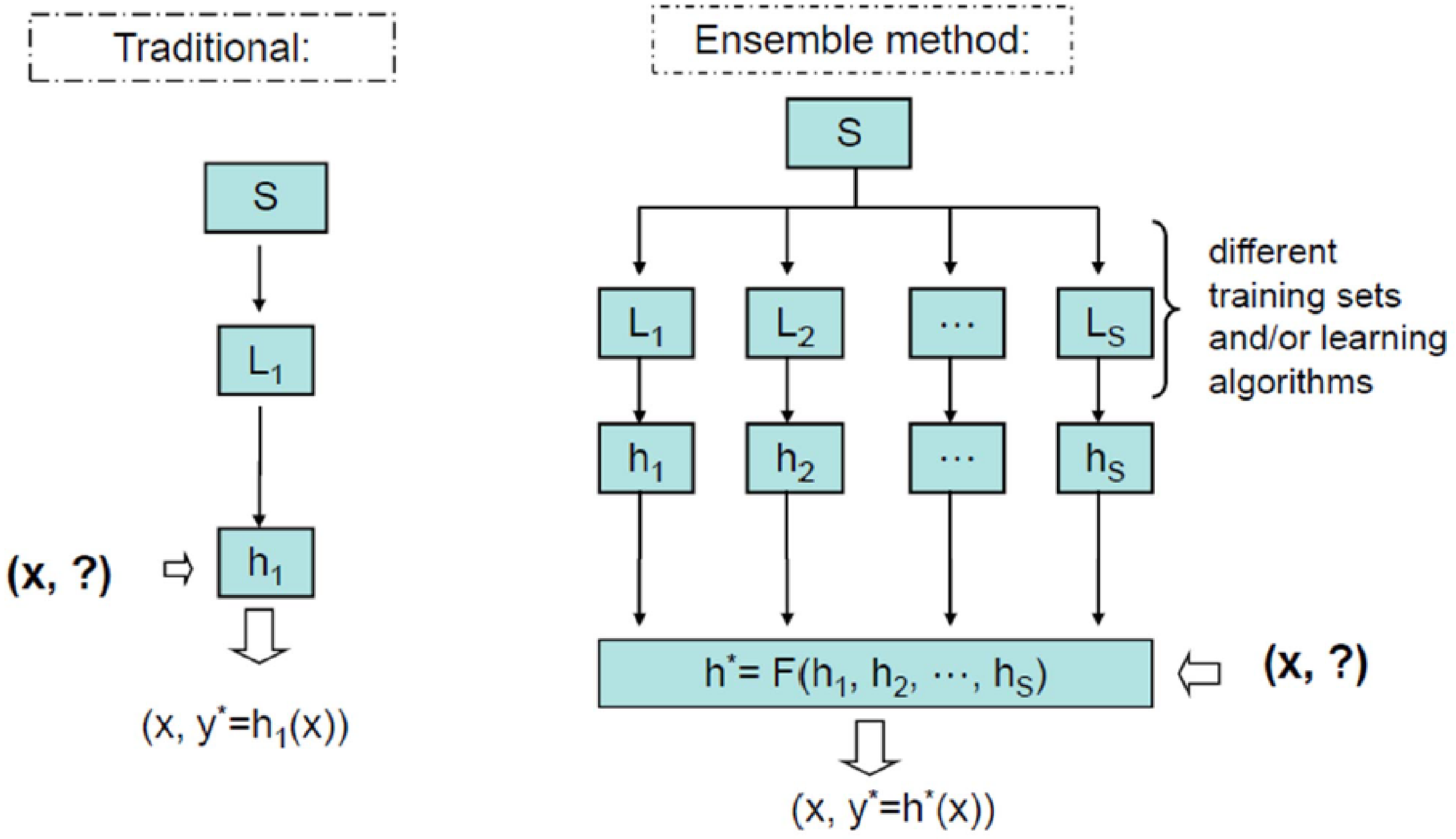
- Non-linear classifiers
  - Decision-Trees
  - SVM with Kernels
  - Neural Networks (later)
  - Naïve Bayes
- Complex decision boundaries
- Type of boundary depends on the classifier
- Advantages and disadvantages?



Machine Learning – Classification III

- **Directly** learn a mapping y = f(**x**)
  - No uncertainty is captured
- Learn the **joint distribution** i.e., learn p(y**,x)**
  - Captures uncertainty about both the attributes **x** and the target y
- Learn the **conditional distribution** i.e., learn p(y|**x**)
  - p(**x**,y) = p(y|x)p(x)
  - Hence this avoids modeling the distribution of **x**
  - In general, this is akin to assuming an uniform distribution over **x**
  - Can also be considered as saying "I do not care about **x** but only P(y|**x**)
- Once we learn p, how do we choose y? This is called as **decision-theory**

Machine Learning – Classification IV



Machine Learning – Classification V

- Deterministic Vs Probabilistic
  - Perceptron, SVMs, Nearest Neighbors, Decision-Trees etc classify a point as either positive or negative
  - Naïve Bayes, LDA, Logistic(?) etc return a distribution over the target class
  - Discussion: How can we make the following probabilistic?
    - Decision-Trees
    - SVMs?
  - Reverse: How can the probabilistic methods allow for decisions?

Machine Learning – Classification VI

Generative vs Discriminative

Generative

- Generative: Create something that can generate ex's
- Can create complete input feature vectors
  - Describes probability distributions for all features
  - Stochastically create a plausible feature vector
  - Example: Bayes net
- Make a model that *generates* positives
- Make a model that *generates* negatives
- Classify a test example based on which is more likely to generate it

Discriminative

- What differentiates class A from class B?
- Don't try to model all the features, instead focus on the task of categorizing
  - Captures differences between categories
  - May not use all features in models
  - Examples: decision trees, SVMs, & neural nets
- Typically more efficient and simpler

LTU's – 3 approaches

- Directly learn a classifier
  - Perceptron – Based on gradient descent algorithm. Is eager and performs local search on the weight vector. i.e., starts with an initial set of weights and modifies it iteratively based on the example. The gradient is computed using some loss function (usually 0/1 loss or hinge loss – what is the difference?) Online or batch?
- Learn a discriminative function
  - Logistic Regression – Learns P(y|**x**). Transforms a linear function using an exponential function (Why?) Perform gradient descent on the log likelihood. (Can you prove that Logistic regression learns a LTU?)
- Learn a generative function
  - Linear Discriminant analysis – Learns P(y,**x**). Not necessary for mid-term

Decision Trees

- Recursively split the features based on some statistical measure – information gain, Mutual Information, gini index p(1-p)
- Splits are binary in general – can you make multi-way split? What will information gain favor? Binary or multi-way?
- What is a decision stump?
- How does the decision boundary look like?
- Pruning will allow decision trees to have a reduced depth
- When will decision trees overfit? What will you prefer – small depths or a very large depth?
- Expressiveness – Can they represent an arbitrary boolean function? How about a disjunction of conjunctions and negations etc?
- How can you avoid overfitting?

K-Nearest Neighbors

- Lazy algorithm – does not build a classifier from all the training data. Instead builds them lazily as every example comes in
- Decision boundaries are drawn between examples of **opposite** classes. What are the distance measures?
- Complex decision boundaries – Voronoi diagram
- How do they change with k = 1 to 5? When do you choose a small k vs when do you choose a large k? What are the advantages and disdvantages of each choice?
- How can Nearest neighbors overfit? How do you avoid overfitting?
- How does noise affect NN? How can their effect be reduced?

SVMs

- In the simplest case, SVMs search for the hyperplane that maximizes the separation between the two classes (the margin)
- Examples closest to the hyperplane are support vectors and the margin is the distance between support vectors
- The minimization problem is a quadratic optimization with linear inequality constraints. The key idea is to convert this to a dual problem with smaller number of constraints
- Handling noise – Soft margin SVMs. What is the objective here? What are the "support vectors" in this case?
- Minimal change to the optimization function
- Non linear classifiers – Kernels? What is the kernel trick? How do we keep the linear algorithms?
- What can be represented? When can SVMs overfit? How can you prevent that?

Naïve Bayes

- Generative model – Learns the joint distribution of the labels and features P(y,**x**)
- What is the key assumption in NB? When is that a good one? When is it a bad assumption?
- Learning is very simple. Just using MLE. What is the issue with a simple MLE? How can you fix this?
- Can handle a variety of data types. Why?
- First thing to try in most problems – simple yet very efficient

Evaluation

- What is a confusion matrix? What are true positives, false positives, true negatives, false negatives?
- Is accuracy a good measure? When is it desirable and when is it not?
- What are ROC curves? What is precision-recall?
- When do you do cross-validation? What is a hold-out data set?
- Can you test on training data? What will be the result of such a test?

Bias-Variance

- Error = Variance + Bias^2 + Noise^2
- Intuition – When combining multiple independent decisions, random errors cancel each other out
- High bias $\rightarrow$ underfitting
- High variance $\rightarrow$ overfitting
- Reduction in bias achieved by more complex classifiers
- Reduction in variance achieved by simpler classifiers