

Learning

Generic Linear Learning Algorithm	25
Updates	26
Perceptron Example	27
Perceptron Properties	28

Learning	2
Learning Definitions	2
Supervised Learning	3
Example of Examples	4
Evaluating Predictions	5
Decision Trees	6
Definition	6
Example of a Decision Tree	7
Algorithm for Learning Decision Trees	8
Selecting a Feature: Information Gain	9
Plot of Information Function	10
Plot of Information Gain	11
Example of Feature Selection	12
Feature Selection, Continued	13
Other Choices	14
Special Cases in Decision Trees	15
Iris Dataset	16
Naive Bayes	17
Numerical Learning	17
Naive Bayes	18
Naive Bayes Learning	19
Naive Bayes Example	20
Naive Bayes Example Continued	21
Linear Regression and Classification	22
Linear Functions	22
Linear Classification	23
Example of Numeric Examples	24

Learning

2

Learning Definitions

- *Learning* is improvement of performance (time, accuracy).
- In *supervised learning*, from training examples of input-output pairs, predict the output of a new input.
- In *unsupervised learning*, examples do not have outputs. The most common task is *clustering*.
- In *semi-supervised learning*, some examples have outputs. For example, in *reinforcement learning*, an input is a series of actions, and the output is intermittent feedback.

CS 3793/5233 Artificial Intelligence

Learning – 2

Supervised Learning

- Assume the learner is given the following:
 - a set of input features, X_1, \dots, X_n ;
 - a target feature, Y ;
 - a set of *training examples*, each with values for the X 's and Y
 - a set of *test examples*, each with values only for the X 's.
- The learner finds a *hypothesis* h to predict the target from the inputs.
- Usually, h is restricted to be an element from a *hypothesis space*.
- *Regression* is when the target is continuous.
- *Classification* is when the target is discrete.

CS 3793/5233 Artificial Intelligence

Learning – 3

Example of Examples

No.	Input Features				Target
	Outlook	Temp	Humidity	Windy	
1	sunny	hot	high	false	neg
2	sunny	hot	high	true	neg
3	overcast	hot	high	false	pos
4	rain	mild	high	false	pos
5	rain	cool	normal	false	pos
6	rain	cool	normal	true	neg
7	overcast	cool	normal	true	pos
8	sunny	mild	high	false	neg
9	sunny	cool	normal	false	pos
10	rain	mild	normal	false	pos
11	sunny	mild	normal	true	pos
12	overcast	mild	high	true	pos
13	overcast	hot	normal	false	pos
14	rain	mild	high	true	neg

CS 3793/5233 Artificial Intelligence

Learning – 4

Evaluating Predictions

- Let y_e be the target value for example e .
- Let \hat{y}_e be the predicted value.
- *Error* (or *loss*) measures how close \hat{y}_e is to y_e .
- Zero-One Error: if $y_e \neq \hat{y}_e$, then 1, else 0
- Absolute Error: $|y_e - \hat{y}_e|$
- Squared Error: $(y_e - \hat{y}_e)^2$
- Entropy: $-(y_e \log \hat{y}_e + (1 - y_e) \log(1 - \hat{y}_e))$ (assumes y_e and \hat{y}_e are probabilities.)
- and many variations.
- For classification, use $y_e \in \{0, 1\}$ or $\{-1, 1\}$.
- Secret of machine learning:
update hypothesis to reduce error.

CS 3793/5233 Artificial Intelligence

Learning – 5

Decision Trees

6

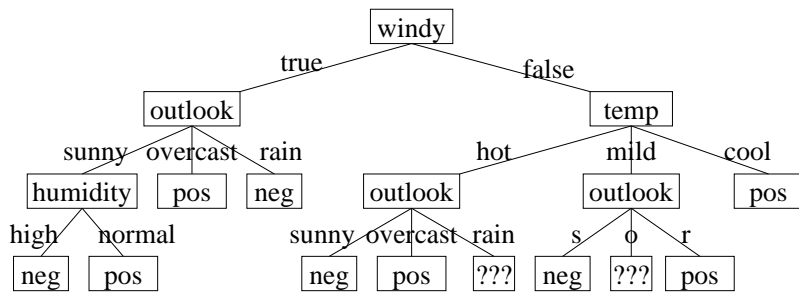
Definition

- Decision trees are a representation for classification.
 - Each nonleaf is labeled by a feature.
 - Edges from nonleaf to children are labeled by feature values.
 - Each leaf is labeled by a prediction.
- Typical Algorithm: Construct the tree top-down.
 - Find the "best" feature.
 - Split examples based on feature's values.

CS 3793/5233 Artificial Intelligence

Learning - 6

Example of a Decision Tree



CS 3793/5233 Artificial Intelligence

Learning - 7

Algorithm for Learning Decision Trees

Procedure *DTLerner*(*X*, *Y*, *E*)

Inputs *X*: set of input features, $X = \{X_1, \dots, X_n\}$

Y: target feature

E: set of training examples

if stopping criterion is true then

 return a leaf labeled with prediction of *Y*

Select feature $X_i \in X$, with domain *V*

let *T* = nonleaf node labeled X_i

for each $v \in V$

 let $E' = \{e \in E : X_i = v\}$

 let $T' = \text{DTLerner}(X, Y, E')$

 add edge from *T* to *T'* labeled *v*

return *T*

CS 3793/5233 Artificial Intelligence

Learning - 8

Selecting a Feature: Information Gain

- *p* positive examples and *n* negative examples
- The information contained is:

$$I(p, n) = -\frac{p}{p+n} \log_2 \frac{p}{p+n} - \frac{n}{p+n} \log_2 \frac{n}{p+n}$$
- Feature X_i has *v* values, p_j positive examples and n_j negative examples when $X_i = v_j$
- The *Remainder* of X_i is:

$$\text{Remainder}(X_i) = \sum_{j=1}^v \frac{p_j + n_j}{p + n} I(p_j, n_j)$$

- The information gain of X_i is:

$$\text{Gain}(X_i) = I(p, n) - \text{Remainder}(X_i)$$

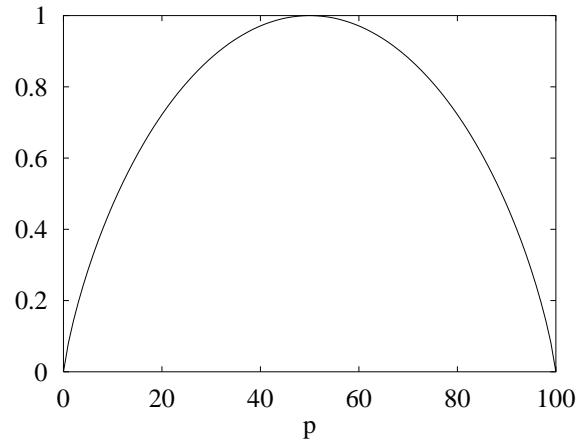
CS 3793/5233 Artificial Intelligence

Learning - 9

Plot of Information Function

p positive examples and n negative examples

$I(p, n=100-p)$

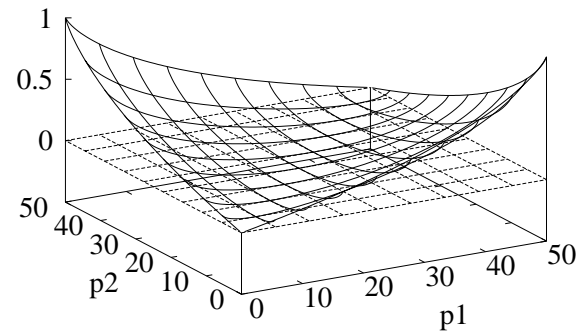


Plot of Information Gain

p_1 positive and n_1 negative ex. when $X_i = v_1$

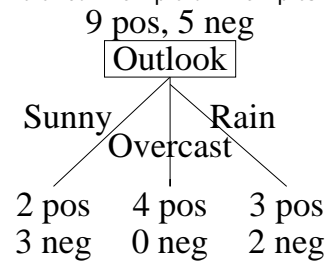
p_2 positive and n_2 negative ex. when $X_i = v_2$

$gain(p_1, n_1=50-p_1, p_2, n_2=50-p_2)$

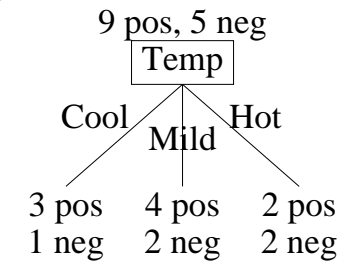


Example of Feature Selection

Refer to Example of Examples earlier.

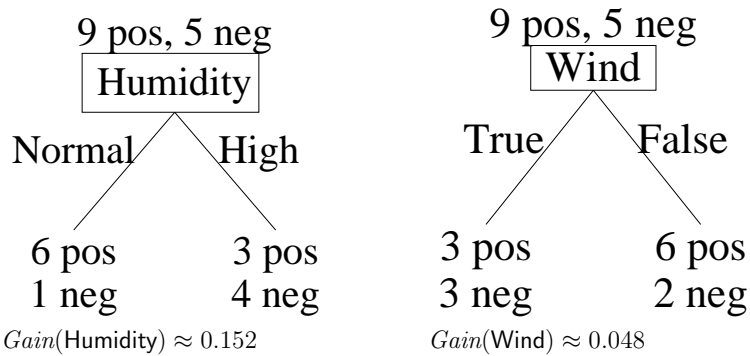


$Gain(Outlook) \approx 0.246$



$Gain(Temp) \approx 0.029$

Feature Selection, Continued



Outlook has the highest gain.
 Overcast branch is pure.
 Need to construct DTs for two branches.

CS 3793/5233 Artificial Intelligence

Learning – 13

Other Choices

- When to stop:
 - all examples are classified the same
 - all examples have the same feature values
 - too few examples
- *Overfitting* occurs when the algorithm tries to fit noise in the training data (outliers, random fluctuations, approx. decision boundary).
- Handling overfitting: use part of training set as a *validation set*.
 - create decision tree with *training set*
 - prune decision tree with *validation set*

CS 3793/5233 Artificial Intelligence

Learning – 14

Special Cases in Decision Trees

- Feature X_i is numeric.
 - Find best $X_i \leq v$ test. Requires sorting.
 - Or: Discretization. Partition X_i into ranges.
- Feature X_i has missing values.
 - Pretend missing is just another value.
 - Or: Ignore missing values. Split examples with missing values across branches.
- Feature X_i has many discrete values.
 - Find best $X_i = v$ test. Forms binary tree.
 - Or: Partition values into subsets.

CS 3793/5233 Artificial Intelligence

Learning – 15

Iris Dataset

No.	Input Features				Target
	Sepal length	Sepal width	Petal length	Petal width	
1	5.8	4.0	1.2	0.2	I. setosa
2	5.7	4.4	1.5	0.4	I. setosa
3	4.8	3.4	1.9	0.2	I. setosa
4	5.0	3.5	1.6	0.6	I. setosa
5	6.7	3.0	5.0	1.7	I. versicolor
6	6.0	2.7	5.1	1.6	I. versicolor
7	5.9	3.2	4.8	1.8	I. versicolor
8	6.0	3.4	4.5	1.6	I. versicolor
9	7.6	3.0	6.6	2.1	I. virginica
10	4.9	2.5	4.5	1.7	I. virginica
11	7.3	2.9	6.3	1.8	I. virginica
12	6.7	2.5	5.8	1.8	I. virginica

CS 3793/5233 Artificial Intelligence

Learning – 16

Naive Bayes

17

Numerical Learning

- Numerical learning methods learn the parameters or weights of a model, often by optimizing an error function. Examples include:
- Calculate the parameters of a probability distribution.
- Separate positive from negative examples by a decision boundary.
- Find points close to positive but far from negative examples.
- Update parameters to decrease error.

CS 3793/5233 Artificial Intelligence

Learning – 17

Naive Bayes

- For target class Y and features X_i , assume:
$$P(Y, X_1, \dots, X_n) = P(Y)P(X_1|Y)\dots P(X_n|Y)$$
- This corresponds to a Bayesian network where Y is the sole parent of each X_i .
- To calculate the belief in Y :
$$P(Y | X_1, \dots, X_n) = \frac{P(Y, X_1, \dots, X_n)}{P(X_1, \dots, X_n)}$$
- The denominator is the same for all values of Y , so to compare only the numerator needs to be calculated.

CS 3793/5233 Artificial Intelligence

Learning – 18

Naive Bayes Learning

- Estimate prior and conditional probabilities by counting, e.g.,
 - $Y = \text{pos}$ in 9 of the 14 examples
 - $X_1 = \text{sunny}$ in 2 examples where $Y = \text{pos}$.
- If an outcome occurs m times out of n examples, Laplace's law of succession recommends the estimate $(m + 1)/(n + k)$ where k is the number of outcomes.
 - Estimate $P(Y = \text{pos}) = (9 + 1)/(14 + 2) = 10/16$
 - Estimate $P(X_1 = \text{sunny} | Y = \text{pos}) = (2 + 1)/(9 + 3) = 3/12$

CS 3793/5233 Artificial Intelligence

Learning – 19

Naive Bayes Example

Using Laplace's law of succession on the 14 examples:

$$P(Y = \text{pos}) = (9 + 1)/(14 + 2) = 10/16$$

$$P(Y = \text{neg}) = (5 + 1)/(14 + 2) = 6/16$$

$$P(X_1 = \text{sunny} | Y = \text{pos}) = (2 + 1)/(9 + 3) = 3/12$$

$$P(X_1 = \text{overcast} | Y = \text{pos}) = (4 + 1)/(9 + 3) = 5/12$$

$$P(X_1 = \text{rain} | Y = \text{pos}) = (3 + 1)/(9 + 3) = 4/12$$

CS 3793/5233 Artificial Intelligence

Learning – 20

Naive Bayes Example Continued

For the first example:

$$\begin{aligned} P(Y = \text{pos} | \text{sunny, hot, high, false}) \\ &= \alpha (10/16) (3/12) (3/12) (4/11) (7/11) \\ &\approx \alpha 0.00904 \end{aligned}$$

$$\begin{aligned} P(Y = \text{neg} | \text{sunny, hot, high, false}) \\ &= \alpha (6/16) (4/8) (3/8) (5/7) (3/7) \\ &\approx \alpha 0.02152 \\ &\approx \frac{0.02152}{0.00904 + 0.02152} \approx 0.704 \end{aligned}$$

CS 3793/5233 Artificial Intelligence

Learning – 21

Linear Regression and Classification

22

Linear Functions

- A *linear function* of the input features is a dot product of the *weights* and the inputs.

Inputs: $\mathbf{x} = (1.0, x_1, \dots, x_n)$

Weights: $\mathbf{w} = (w_0, w_1, \dots, w_n)$

Dot product: $\mathbf{w} \cdot \mathbf{x} = w_0 + w_1x_1 + \dots + w_nx_n$

- If y is the target and $\hat{y} = \mathbf{w} \cdot \mathbf{x}$:

Regression:

Squared error: $(y - \hat{y})^2$

Absolute error: $|y - \hat{y}|$

Classification (assume $y \in \{-1, 1\}$):

Hinge loss: $\max(0, 1 - y\hat{y})$

Logistic loss: $-\log(1/(1 + e^{-y\hat{y}}))$

CS 3793/5233 Artificial Intelligence

Learning - 22

Linear Classification

Why $y * \hat{y}$ all over the place?

- The goal of linear regression is $y = \hat{y}$.
- The goal of linear classification is not $y = \hat{y}$, but $\text{sign}(y) = \text{sign}(\hat{y})$.
- If y and \hat{y} have the same sign, then $y\hat{y} > 0$.
- The hinge loss includes a *margin*. Its goal is $y\hat{y} \geq 1$.
- The logistic loss is for interpreting \hat{y} probabilistically (as a log-likelihood). It has larger values for negative $y\hat{y}$.
- The global minimum can be found for the hinge and logistic loss.

CS 3793/5233 Artificial Intelligence

Learning - 23

Example of Numeric Examples

No.	Input Features						Target Feature
	Sunny	Rainy	Hot	Cool	Humid	Windy	
1	1	0	1	0	1	0	-1
2	1	0	1	0	1	1	-1
3	0	0	1	0	1	0	1
4	0	1	0	0	1	0	1
5	0	1	0	1	0	0	1
6	0	1	0	1	0	1	-1
7	0	0	0	1	0	1	1
8	1	0	0	0	1	0	-1
9	1	0	0	1	0	0	1
10	0	1	0	0	0	0	1
11	1	0	0	0	0	1	1
12	0	0	0	0	1	1	1
13	0	0	1	0	0	0	1
14	0	1	0	0	1	1	-1

CS 3793/5233 Artificial Intelligence

Learning - 24

Generic Linear Learning Algorithm

Procedure *LinearLerner*(X, Y, E, η)

Inputs X : set of input features, $X = \{X_1, \dots, X_n\}$

Y : target feature

E : set of training examples

η : learning rate

initialize all weights w_0, w_1, \dots, w_n to zero

repeat until termination

for each example $e = (\mathbf{x}, y) \in E$

$\hat{y} \leftarrow \mathbf{w} \cdot \mathbf{x}$

$\delta \leftarrow$ update based on y and \hat{y}

$\mathbf{w} \leftarrow \mathbf{w} + \eta\delta\mathbf{x}$

return \mathbf{w}

CS 3793/5233 Artificial Intelligence

Learning - 25

Updates

Name	$\delta \leftarrow$
Regression:	
Squared error	$y - \hat{y}$
Absolute error	$\text{sign}(y - \hat{y})$
Classification:	
Perceptron	if $y\hat{y} \leq 0$ then y else 0
Hinge loss	if $y\hat{y} < 1$ then y else 0
Logistic loss	$y/(1 + e^{y\hat{y}})$

Except for perceptron, the update is based on the derivative of the error wrt the weights.

Note: For squared error, the optimal solution can be directly computed.

CS 3793/5233 Artificial Intelligence

Learning - 26

Perceptron Example

Using the learning rate $\eta = 1$:

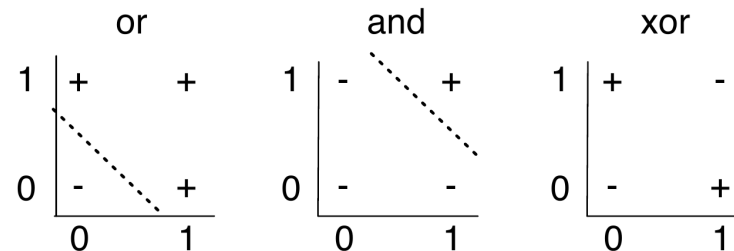
Features								Weights				
X_0	X_1	X_2	X_3	X_4	Y	\hat{y}	$y\hat{y}$	w_0	w_1	w_2	w_3	w_4
1	0	0	0	1	-1	0	0	0	0	0	0	0
1	1	1	1	0	1	-1	-1	0	1	1	1	-1
1	1	1	1	1	1	2	2	0	1	1	1	-1
1	0	0	1	1	-1	0	0	-1	1	1	0	-2
1	0	0	0	0	1	-1	-1	0	1	1	0	-2
1	0	1	0	1	-1	-1	1	0	1	1	0	-2
1	1	0	0	0	1	1	1	0	1	1	0	-2
1	1	0	1	1	1	-1	-1	1	2	1	1	-1
1	0	1	0	0	-1	2	-2	0	2	0	1	-1

CS 3793/5233 Artificial Intelligence

Learning - 27

Perceptron Properties

- The perceptron can learn *linearly separable* examples with zero error. Linearly separable = exists w with zero error on all examples.



- Usually, many *epochs* (passes over the training examples) are needed until convergence.
- If zero error is not possible, use hinge/logistic loss and $\eta \approx 0.1/n$, where n is $\max \mathbf{x} \cdot \mathbf{x}$.

CS 3793/5233 Artificial Intelligence

Learning - 28