ECE M146 Introduction to Machine Learning

Lecture 5 - Spring 2021

Prof. Lara Dolecek

ECE Department, UCLA

Today's Lecture

Recap:

- Perceptron and linear regression; gradient descent

New topics:

- Logistic regression
- More on loss functions

Recap

- Perceptron is on-line algorithm for binary classification; at test time, outputs one of two choices
- Linear least squares for linear regression; at test time, outputs a real-valued number
- (Stochastic) gradient descent can be used for both problems.

Today's Lecture

Recap:

- Perceptron and linear regression; gradient descent

New topics:

- Logistic regression
- More on loss functions

Logistic regression

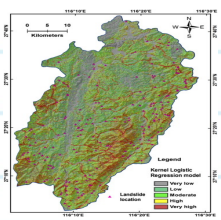
- Can be viewed as an in-between method in the sense that it outputs a real value, but it is used for classification.
- Like the methods studied so far, it represents a linear model.
- It is an instance of a probabilistic discriminative model, where we model p(y|x). As such, it is modeling-wise more complex than those e.g., perceptron, that are described by a discriminant function.
- Later on, we will see generative models such as naïve Bayes, that model p(x|y).

$$p(x, y) = p(x|y)p(y)$$

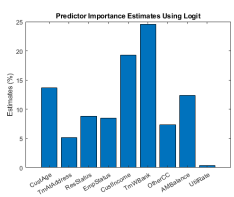
Logistic Regression Applications

- Used where other parametric binary classification methods can be used: Perceptron (so far). Also: SVM, LDA (soon)
- Applications:

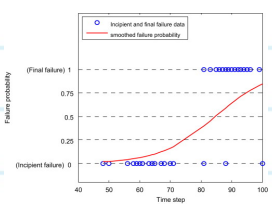
Land slide hazard prediction



Credit default prediction



Machine failure prediction

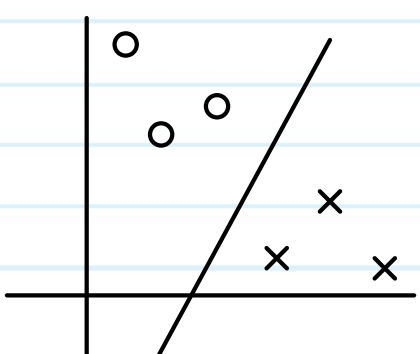


https://www.mathworks.com/help/risk/creditscorecard-compare-logistic-regression-decision-trees.html

Hong et al., "Spatial prediction of landslide hazard at the Yihuang area (China) using two-class kernel logistic regression, alternating decision tree and support vector machines," 2015

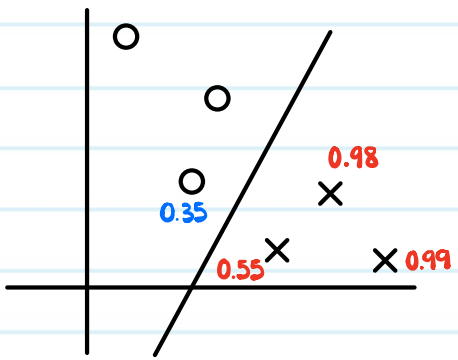
Logistic regression – key idea

- Probabilistically capture the confidence of the classified point.
- The closer the point is to the decision boundary, the less confidence the model has in its value; the further away the point is from the decision boundary, the more confidence the model has in its value.
- Contrast with perceptron.



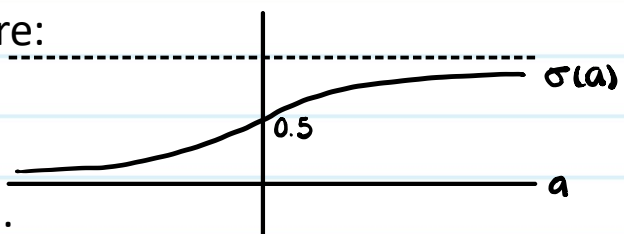
Contrast with perceptron

- 0/1 vs. probabilities



Logistic sigmoid

- A convenient function that we will use in logistic regression.
- Picture:



- Math:

$$\sigma(a) = \frac{1}{1+e^{-a}}$$

Inference set-up

- Label is again binary, but we switch to {0,1} for mathematical convenience.
- This does not make any difference conceptually as it is still binary classification.
- Define the following functions:

$$f(x) = P(y=+1 \mid x)$$
$$1 - f(x) = P(y=0 \mid x)$$

- These functions are unknown. Our goal is to model these conditional probabilities.

Logistic regression modeling

- In logistic regression, we use the following modeling for the conditional probabilities:

$$P(y=+1|x) = \sigma(\vec{w}^T\vec{x})$$
$$P(y=0|x) = 1-\sigma(\vec{w}^T\vec{x})$$

- Again, as in perceptron and in linear least squares, the goal is to find the best vector w under an appropriately chosen loss function.

A useful property of the logistic sigmoid

$$\sigma(a) = \frac{1}{1+e^{-a}}$$

$$= 1 - \frac{1}{1+e^{a}}$$

$$\sigma(a) = 1 - \sigma(-a)$$

Intuition on why this functional format

- Consider the conditional probability P(y=1|x).

$$P(y=1 \mid x) = \frac{P(x \mid y=1) \cdot P(y=1)}{P(x)}$$

$$\downarrow$$

$$P(x \mid y=1)\,P(y=1) + P(x \mid y=0)\,P(y=0)$$

$$a = \ln\left(\frac{P(x \mid y=1)\,P(y=1)}{P(x \mid y=0)\,P(y=0)}\right)$$

Maximization of the likelihood

- Since we now focus on P(y|x), the goal is to maximize the following:

$(x_1, y_1), \dots (x_n, y_n)$

$$\prod_{i=1}^{N} P(y_i|x_i)$$

- We have N data points.
- How ?

$$\max \prod_{i=1}^{N} P(y_i|x_i) \quad \Leftrightarrow \max \sum_{i=1}^{N} \log P(y_i|x_i)$$

$$\Leftrightarrow \max \log \prod_{i=1}^{N} P(y_i|x_i) \quad \Leftrightarrow \min -\sum_{i=1}^{N} \log P(y_i|x_i)$$

- We now have a function to **minimize**.

Optimization details

- Unfortunately, we cannot take the derivatives and get a closed form solution as in LLS, but we can apply gradient descent!
- Recall the expressions for the conditional probabilities:

$$P(y=1|x) = \sigma(w^T x)$$

$$P(y=0|x) = 1 - \sigma(w^T x)$$

- Mathematical trick:

$$P(y_i|x_i) = [\sigma(w^T \cdot x_i)]^{y_i} \cdot [1 - \sigma(w^T \cdot x_i)]^{1-y_i}$$

- Check:

$y_i = 0$ $\quad 1 \cdot [1 - \sigma(w^T \cdot x_i)]^1$

$y_i = 1$ $\quad [\sigma(w^T \cdot x_i)]^1 \cdot 1$

Back to our set up

- Then, and this is why we applied log, bring the exponents down to get:

$$\min -\sum_{i=1}^{n} \log P(y_i|x_i)$$

$$= \min -\sum_{i=1}^{n} [y_i \log \sigma(w^T \cdot x_i) + (1-y_i) \log(1-\sigma(w^T \cdot x_i))]$$

- Take the gradient with respect to w.

First, an auxiliary result

- A result from matrix calculus we'll need:

$$\frac{\partial}{\partial w}\sigma(w^T \cdot x) = (1 - \sigma(w^T \cdot x))\,\sigma(w^T \cdot x) \cdot x$$

- Because: $\frac{\partial}{\partial a}\sigma(a) = \underbrace{\frac{e^{-a}}{1+e^{-a}}}_{1-\sigma(a)} \cdot \underbrace{\frac{1}{1+e^{-a}}}_{\sigma(a)}$

First, an auxiliary result

- And because: $\frac{\partial}{\partial w}(w^T \cdot x) = x$

Back to our minimization problem

$$\min \; -\sum_{i=1}^{N}\left[y_i \log \sigma(w^T x_i) + (1-y_i)\log(1-\sigma(w^T x_i))\right]$$

$$\uparrow$$

$$L(w)$$

$$\frac{\partial L(w)}{\partial w} = -\sum_{i=1}^{N}\Big[y_i \frac{1}{\sigma(w^T x_i)} \cdot \sigma(w^T x_i)\cdot(1-\sigma(w^T x_i))\cdot x_i - (1-y_i)\frac{-1}{1-\sigma(w^T x_i)}\cdot\sigma(w^T x_i)\cdot(1-\sigma(w^T x_i))\cdot x_i\Big]$$

$$\frac{\partial L(w)}{\partial w} = -\sum_{i=1}^{N}\left[y_i(1-\sigma(w^T x_i))x_i + (y_i-1)\sigma(w^T x_i)x_i\right]$$

$$= -\sum_{i=1}^{N}\left[y_i x_i - \sigma(w^T x_i)\right]$$

Key result

- Gradient descent (as before)

$$\frac{\partial L}{\partial w} = \sum_{i=1}^{N} (\sigma(w^T x_i) - y_i) \cdot x_i$$

$$w_{new} = w_{old} - \eta \frac{\frac{\partial L(w_{old})}{\partial w_{old}}}{\left\| \frac{\partial L(w_{old})}{\partial w_{old}} \right\|}$$

$$y_i \in \{0, 1\} \qquad \sigma(w^T x_i) \in [0, 1]$$

(Stochastic) gradient update rule

evaluate at just one data point (x_j, y_j)

$$(\sigma(w^T x_j) - y_j) \cdot x_j$$

Connection to cross-entropy

- Consider two RVs, X and Y. Suppose X is distributed as a Bernoulli RV with parameter p and Y is distributed as a Bernoulli RV with parameter q.

$$X = \begin{cases} 1 & n.p & p \\ 0 & n.p & 1-p \end{cases} \qquad Y = \begin{cases} 1 & n.p & q \\ 0 & n.p & 1-q \end{cases}$$

- Cross-entropy is defined as:

$$p \log q + (1-p) \log(1-q)$$

- Note that the loss is a scaled version of the cross-entropy.

$$y_i \ln(\sigma(w^T \cdot x_i)) + (1-y_i) \ln(1-\sigma(w^T \cdot x_i))$$

Connections to quadratic loss

- Recall that in linear regression with quadratic loss we saw:

$$L(w) = \sum_{i=1}^{N} (w^T x_i - y_i)^2$$

- Derivatives were of the same format:

$$\frac{\partial L(w)}{\partial w} = \sum_{i=1}^{N} 2(w^T x_i - y_i) x_i$$

Further discussion on logistic regression

- Mathematical derivations thus far were for the batch gradient descent; this method can also be done with stochastic gradient descent using one data point at the time.

drop the Σ in $\frac{\partial L}{\partial w}$

At test time

- Once we have the weight vector, at testing time, we perform the following.

compute w using (S)GD

test point x_{TEST}

$$\sigma(w^T \cdot x_{TEST}) = P(y_i = 1 \mid x_{TEST})$$
$$1 - \sigma(w^T \cdot x_{TEST}) = P(y_i = 0 \mid x_{TEST})$$

At test time

- Once we have the weight vector, at testing time, we perform the following.

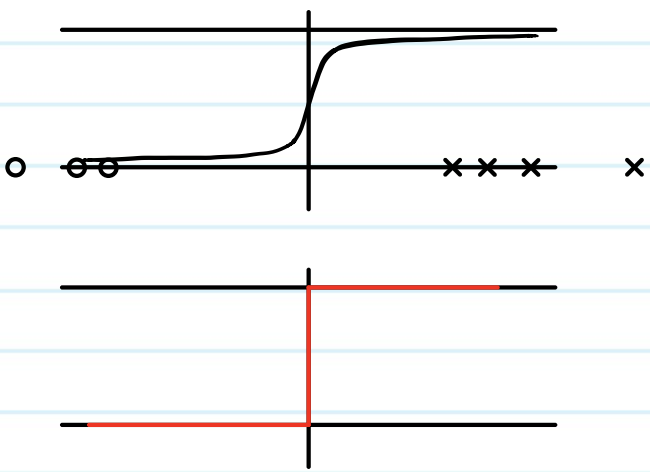
$$\text{compare} \quad \sigma(w^T \cdot x_{TEST}) \underset{y=0}{\overset{y=1}{\gtrless}} 1 - \sigma(w^T \cdot x_{TEST})$$

$$\text{on boundary:} \quad \sigma(w^T \cdot x_{TEST}) = 0.5$$

- Don't confuse logistic sigmoid with the decision boundary.
- **Decision boundary is still linear**.

When training data is linearly separable

- Overfits! Defeats the purpose of probabilistic modeling.



Today's Lecture

Recap:

- Perceptron and linear regression; gradient descent

New topics:

- Logistic regression
- More on loss functions

Loss functions

indicator function: $\mathbb{1}[\text{expression}] = \begin{cases} 1 & \text{if expression true} \\ 0 & \text{if expression false} \end{cases}$

- There are different loss functions, and we have already seen some. Consider:

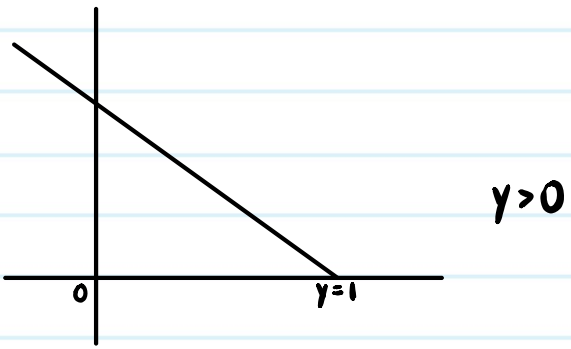
1. 0/1 loss $L: [y \cdot \hat{y} < 0]$
2. Squared loss: $L: (y - \hat{y})^2$
3. Logistic loss: $\frac{1}{\log 2} \cdot \log(1 + e^{-y \cdot \hat{y}})$
4. Exponential loss: $e^{-y \cdot \hat{y}}$
- Hinge loss: $\max\{0, 1 - y \cdot \hat{y}\}$

assume true $y > 0$

- They differ in the kind of penalty they incur.

Loss functions

- Hinge loss – used in SVM



- Logistic regression uses cross-entropy loss, which is equivalent to logistic loss, up to scaling.

Summary so far

- The methods we have introduced so far all have the property that the goal is to produce a weight vector w, under a suitable loss.
- Dimensionality of w does not depend on how many data points we have (parameter N).
- These methods are referred to as **parametric** methods.

- Vector w succinctly captures everything we need to know about the data.

Next: non – parametric methods for classification

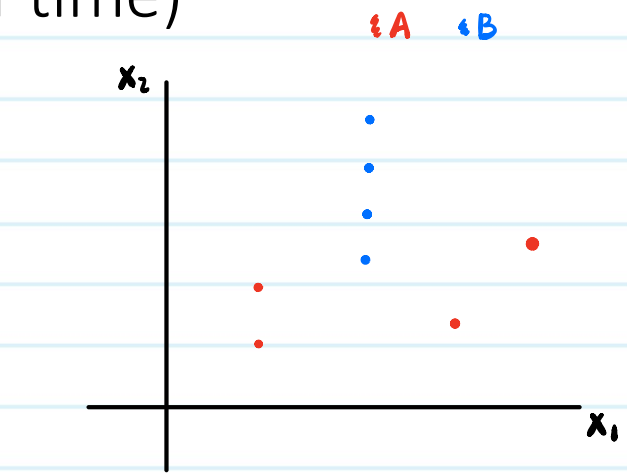
- An alternative approach is to develop a non-parametric model.
- In this case, there is not simple parameter that captures the data set.
- Complexity grows with the size of the data set.

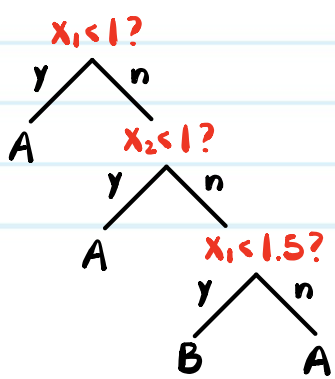
- But, there are other advantages (simplicity for example).

- Decision Trees and K-Nearest Neighbors (KNN) for classification

Decision Trees (if time)

X_1	X_2	y
0.5	0.25	A
0.5	0.5	A
1.5	0.35	A
1.75	1.25	A
1.25	1.2	B
1.25	1.4	B
1.25	1.6	B
1.25	1.8	B





Decision Tree Overview

- Performs sequential inquiries on data attributes to arrive at the classification.
- Can be used for binary or multiclass classification.
- Can also be extended to regression.
- Pros:
  - Has human interpretability.
  - Fairly easy to prepare (doesn't need complex calculations).
  - Typically robust to missing data and outliers.

ECE M146 Introduction to Machine Learning

Lecture 6 - Spring 2021

Prof. Lara Dolecek

ECE Department, UCLA

Today's Lecture

Recap:

- Linear methods: Perceptron, logistic regression, linear regression

New topic:

- Decision Trees

Techniques we have learnt so far

- Perceptron algorithm for binary classification
- Linear regression via linear least squares or gradient descent
- Logistic regression for binary classification via gradient descent
- All these methods derive weight vector w, and the output value is a function of $w^T x$.

Recap: Linear models for classification and regression

- These methods are called **parametric methods**. They are parametrized by w and the choice of the function that relates it to the output.
- In particular, the complexity of vector w, and the decision boundary, does not grow with the number of points, N.

- There are also non-parametric methods.
- Main examples are decision trees and k nearest neighbors.
- Complexity of the decision boundary can grow with N.

Today's Lecture

Recap:

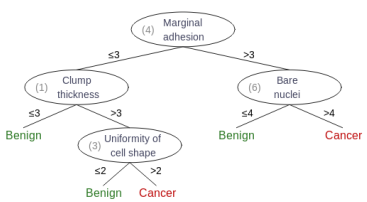
- Linear methods: Perceptron, logistic regression, linear regression

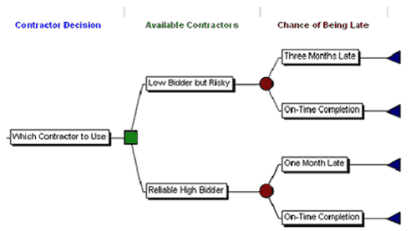
New topic:

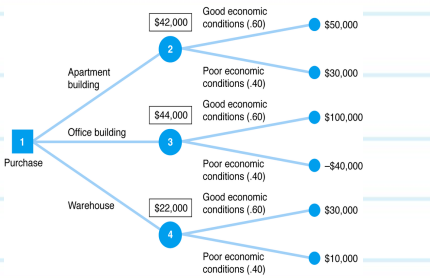
- Decision Trees

Decision Tree Overview

- Broad applications in real life in many areas, such as medicine, engineering, civil planning, law, business.







https://artofcomputing.cecs.anu.edu.au/lab/decisiontree/

http://www.projectrisk.com/decision_trees_for_important_project_decisions.html

http://amsdecisiontreeanalysis.weebly.com/how-to-solve-problems.html

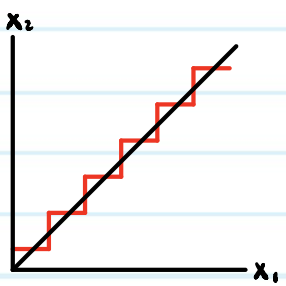
Decision Tree Overview

- Performs sequential inquiries on data attributes to arrive at the classification.
- Can be used for binary or multiclass classification.
- Can also be extended to regression.
- Pros:
  - Has human interpretability.
  - Fairly easy to prepare (doesn't need complex calculations).
  - Typically robust to missing data and outliers.

Decision Tree Overview

- Not good for parity functions
  - Example: k binary attributes

    check 2^k choices to arrive at the output value
- Not good for non-axis aligned decision boundary
  - Example:



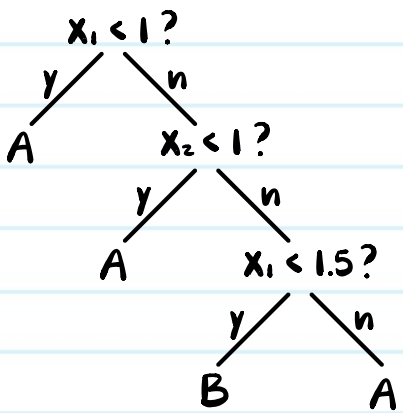
Decision Tree Example

- Creating a decision boundary on two attributes, x_1 and x_2.
- Data Set:

X1	X2	label
0.5	0.25	A
0.5	0.5	A
1.5	0.35	A
1.75	1.2	A
1.1	1.2	B
1.2	1.2	B
1.3	1.3	B
1.2	1.1	B

$(1.25, 1.25) \rightarrow B$

Decision Tree:



Decision Tree Example

- Creating a decision boundary on two attributes, x_1 and x_2.
- Once we have the tree built, at testing time, go down the tree until a leaf node corresponding to the test point is reached.

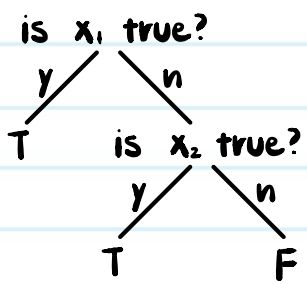
Properties of the Decision Tree Classification

- This approach allows for modeling of fairly complicated decision boundaries.

- This approach is expressive in the sense that any Boolean combination of attributes can be represented.

- Example:

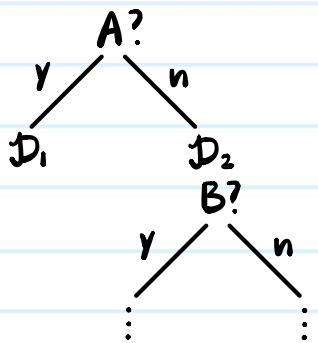
X_1	X_2	X_1 or X_2
T	T	T
T	F	T
F	T	T
F	F	F



Now we have a sense of what a decision tree can do. But how do we create one ?

- What is a sequence of questions that should be asked ?
  - Say we have K possible binary attributes. That is 2^K combinations!
- How do the size and the profile of the training data set impact the answer ?
  - There could be several solutions that agree on the available part of the combinations.
- Finding the smallest decision tree that correctly classifies all training points is NP hard.
- We build the tree **greedily**.

Procedure



- Notation: Node n = root of the decision tree.

  Set $\mathcal{D}$ is the set of unclassified examples in the training set.
- While $\mathcal{D}$ is not an empty set:
  - Pick A as the "best" decision attribute (typically binary) to query on
  - Assign A to n
  - For each value of A create a new descendant of n (typically two)
  - Assign class values to descendants based on A
  - Remove from $\mathcal{D}$ examples that are perfectly classified
- If $\mathcal{D}$ is empty, stop. Else, recurse over new leaf nodes.

How to pick the best attribute ?

1. Random choice: query on any attribute, chosen at random

2. Least/most value: choose an attribute with least/most possible values.
   - Example:

   least: $A \in \{0, 1\}$ most: $A \in \{0, 1, \ldots, M\}$

3. Maximum gain: choose the attribute that has the largest expected **information gain**.
   - Unlike 1 and 2, it captures how informative attributes are (statistical measure of goodness).

Capturing information gain -- example

- Suppose we have 80 data points, each specified by the vector $x=(x_1,x_2)$ of attributes and its label y. Assume that x_1,x_2, and y are all in the {T,F} set.

		x_1 = T	x_1 = F
y	T	40	10
	F	0	30

		x_2 = T	x_2 = F
y	T	20	20
	F	20	20

- If we split on x_1:
  - Under T, we can conclusively say that y=T. We have reduced uncertainty in y.
- If we split on x_2:
  - Under either T or F, we have not reduced uncertainty in y. Useless split.

Formalize the notion of the information gain

- Mathematical expression for the information gain (IG) for random variables Y and X:

$$IG(X,Y) = H(Y) - H(Y|X)$$
$$IG(X,Y) = H(X) - H(X|Y)$$

- H(Y) is called the entropy of Y and represents intrinsic uncertainty in Y.
- H(Y|X) is called the conditional entropy of Y given X, and represents the uncertainty in Y once X is revealed.

Entropy – binary case

- Consider a Bernoulli RV Z with parameter p.

$$Z = \begin{cases} 1 & \text{w.p. } p \\ 0 & \text{w.p. } 1-p \end{cases}$$

- Entropy H(Z) is a measure of surprise.
- Formula and picture:

$$H(Z) = -p\log_2 p - (1-p)\log_2(1-p)$$

note: log is base 2; $0\log 0 = 0$

Entropy – binary case

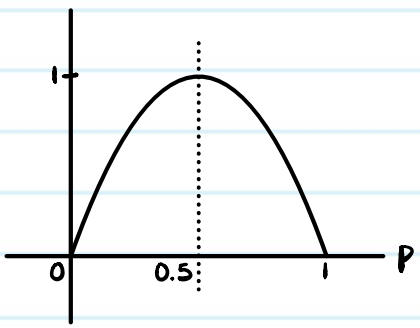
- Formula and picture:

$$H(Z) = -p\log_2 p - (1-p)\log_2(1-p)$$

$$H(0) = -0\log 0 - 1\log 1 = 0$$

$$H(0.5) = -\tfrac{1}{2}\log\tfrac{1}{2} - \tfrac{1}{2}\log\tfrac{1}{2} = 1$$

$$H(1) = -1\log 1 - 0\log 0 = 0$$



Entropy – non binary case

- Suppose now Z has PMF given as $P(Z = z_k) = p_k$ for $1 \leq k \leq K$.
- Entropy of H(Z) is written as:

$0 \leq p_k \leq 1$

$$H(Z) = -\sum_{k=1}^{K} p_k \log p_k$$

- Note that the entropy does not depend on values z_k, but only on probabilities p_k.

$H(p_1, p_2, \dots p_k)$

$$\log p_k \leq 0$$
$$-\log p_k \geq 0$$
$$-\sum_{k=1}^{K} p_k \log p_k \geq 0$$

Conditional Entropy

- We next discuss conditional entropy H(Y|X).
- Let's first consider $H(Y|X=x_j)$:

$$H(Y|X=x_j) = -\sum_{k=1}^{K} P(Y=y_k|X=x_j) \cdot \log\left(P(Y=y_k|X=x_j)\right)$$

- Average over all values of X to get:

$$H(Y|X) = \sum_{j=1}^{J} P(Y|X=x_j)\, P(X=x_j)$$

Conditional Entropy

- Ctd.

$$H(Y|X) = -\sum_{j=1}^{J}\sum_{k=1}^{K} P(Y=y_k|X=x_j)\cdot\log(P(Y=y_k|X=x_j))\cdot P(X=x_j)$$

$$= -\sum_{j=1}^{J}\sum_{k=1}^{K} P(Y=y_k, X=x_j)\cdot\log(P(Y=y_k|X=x_j))$$

$$IG(X,Y) = H(Y) - H(Y|X)$$

$H(Y|X) \geq 0$: when is $H(Y|X) = 0$?

- when $IG(X,Y) = H(Y)$

Back to our binary example

- Recall the set up.

		X_1 = T	X_1 = F
Y	T	40	10
	F	0	30

		X_2 = T	X_2 = F
Y	T	20	20
	F	20	20

- Interpret as random variables based on the empirical data. We want to compare $IG(X_1,Y)$ and $IG(X_2,Y)$.

$$IG(X_1, Y) = H(Y) - H(Y|X_1) \qquad IG(X_2, Y) = H(Y) - H(Y|X_2)$$

- General rule:

$$IG(X,Y) = H(Y) - H(Y|X)$$

if X uninformative about Y: $IG(X,Y) = 0$

Example, continued.

map empirical values to binary random variables w/ probabilities that correspond to sample frequencies

If we query on X_2

- Convert to probability

		X_2 = T	X_2 = F	marginal of y
y	T	20	20	½
	F	20	20	½
marginal of x_2 →		½	½	

table

- Compute $H(Y|X_2)$: need $H(Y|X_2=T)$, $H(Y|X_2=F)$ or joint PMF

$$H(Y|X_2) = -\overbrace{\sum_{x_2}\sum_{y}}^{4 \text{ terms}} P(Y=y, X_2=x_2)\log P(Y=y, X_2=x_2)$$

$$= -\sum_{x_2 \in \{T,F\}}\sum_{y \in \{T,F\}} \frac{1}{4}\log\left(\frac{1}{2}\right) = 1$$

- What is $IG(X_2,Y)$?

$$P(Y=T|X_2=T) = \frac{P(Y=T, X_2=T)}{P(X_2=T)} = \frac{1/4}{1/2} = \frac{1}{2}$$

$H(Y) = 1$

$IG(X_2, Y) = 1 - 1 = 0$

If we query on X_2

- Compute $H(Y|X_2)$:
- What is $IG(X_2,Y)$?

If we query on X_1

- Convert to probability

		X_1 = T	X_1 = F
y	T	40	10
	F	0	30

→

		X_1 = T	X_1 = F	
y	T	½	⅛	5/8
	F	0	3/8	3/8
		½	½	

- Compute $H(Y|X_1)$:
  - Why do we expect it to be < 1?

$$P(Y=T, X_1=T) \cdot \log_2(P(Y=T|X_1=T)) = \frac{1}{2}\log_2\left(\frac{1/2}{1/2}\right) = 0$$

$$P(Y=F, X_1=T) \cdot \log_2(P(Y=F|X_1=T)) = 0$$

$$P(Y=T, X_1=F) \cdot \log_2(P(Y=T|X_1=F)) = \frac{1}{8}\log_2\left(\frac{1/8}{4/8}\right) = \frac{-1}{4}$$

$$P(Y=F, X_1=F) \cdot \log_2(P(Y=F|X_1=F)) = \frac{3}{8}\log_2\left(\frac{3/8}{4/8}\right) = \frac{3}{8}\log_2\left(\frac{3}{4}\right) = \frac{3}{8}\log_2 3 - \frac{3}{4}$$

If we query on X_1

- Compute $H(Y|X_1)$: add them up $1 - \frac{3}{8}\log 3$
- What is $IG(X_1,Y)$?

$$IG(X_1, Y) = H(Y) - H(Y|X_1)$$
$$= 1 - (1 - \tfrac{3}{8}\log 3)$$
$$= \tfrac{3}{8}\log 3$$

$$IG(X_1, Y) > IG(X_2, Y)$$

Relationship to cross entropy

- Today we discussed entropy/conditional entropy. $H(Y)$, $H(Y|X)$
- Binary entropy:

$$H(p) = -p\log p - (1-p)\log(1-p)$$

- Last time, we discussed cross entropy loss in the logistic regression.
- Binary cross-entropy:

$$-p\log q - (1-p)\log(1-q)$$

Let's now look at a bigger example

- Our first example was fairly apparent. But in practice we don't always have such clear cut choices.
- Bigger example:

$P(X_1 = T) = \frac{1}{2}$

$P(X_2 = T) = \frac{1}{2}$

X_1	X_2	Y	# occurrences
T	T	T	20
T	F	T	20
F	T	T	10
F	T	F	10
F	F	F	20

- Note that some combinations didn't even appear and note that some are contradictory.

How do we build the decision tree now ?

- First, let's compute the entropy of Y, H(Y):

$P(Y=T) = 5/8$

$P(Y=F) = 3/8$

$H(Y) = -\frac{5}{8}\log\frac{5}{8} - \frac{3}{8}\log\frac{3}{8}$

- Next, compare $IG(Y,X_1)$ vs. $IG(Y,X_2)$:

$IG(Y, X_1) = H(Y) - H(Y|X_1)$

$IG(Y, X_2) = H(Y) - H(Y|X_2)$

choose the one w/ max IG $\Leftrightarrow$

choose the one w/ smallest conditional entropy term

Conditional entropy calculation

- To compute $H(Y|X_1)$, let's compute $H(Y|X_1 =T)$ and $H(Y|X_1=F)$ first.

$$H(Y|X_1) = H(Y|X_1=T) \cdot P(X_1=T) + H(Y|X_1=F) \cdot P(X_1=F)$$

$$0 = H(Y|X_1=T) = -P(Y=T|X_1=T) \cdot \log_2 P(Y=T|X_1=T) - P(Y=F|X_1=T) \cdot \log_2 P(Y=F|X_1=T)$$

$$P(Y=T|X_1=T) = \frac{1/2}{1/2} = 1 \qquad P(Y=F|X_1=T) = 0$$

$$H(Y|X_1=F) = -P(Y=T|X_1=F) \cdot \log_2 P(Y=T|X_1=F) - P(Y=F|X_1=F) \cdot \log_2 P(Y=F|X_1=F)$$

$$P(Y=T|X_1=F) = \frac{1/3}{1/2} = \frac{1}{4} \qquad P(Y=F|X_1=F) = \frac{3}{4}$$

$$H(Y|X_1=F) = \frac{-1}{4}\log\frac{1}{4} - \frac{3}{4}\log\frac{3}{4}$$

$$H(Y|X_1) = 1 - \frac{3}{8}\log 3$$

Conditional entropy calculation

- To compute $H(Y|X_2)$, let's compute both $H(Y|X_2=T)$ and $H(Y|X_2=F)$ first.

$$H(Y|X_2) = H(Y|X_2=T) \cdot P(X_2=T) + H(Y|X_2=F) \cdot P(X_2=F)$$

$$H(Y|X_2=T) = -P(Y=T|X_2=T) \cdot \log_2 P(Y=T|X_2=T) - P(Y=F|X_2=T) \cdot \log_2 P(Y=F|X_2=T)$$

$P(Y=T|X_2=T) = 3/4$, $P(Y=F|X_2=T) = 1/4$

$$= -\frac{3}{4}\log\frac{3}{4} - \frac{1}{4}\log\frac{1}{4}$$

$$= 2 - \frac{3}{4}\log 3$$

$$H(Y|X_2=F) = -P(Y=T|X_2=F) \cdot \log_2 P(Y=T|X_2=F) - P(Y=F|X_2=F) \cdot \log_2 P(Y=F|X_2=F)$$

$P(Y=T|X_2=F) = 1/2$, $P(Y=F|X_2=F) = 1/2$

$$= 1$$

$$H(Y|X_2) = \frac{1}{2}\left(2 - \frac{3}{4}\log 3\right) + \frac{1}{2}$$

$$= \frac{3}{2} - \frac{3}{8}\log 3$$

Conditional entropy calculation

- To compute $H(Y|X_2)$, let's compute both $H(Y|X_2 = T)$ and $H(Y|X_2 = F)$ first.