Artificial Intelligence 2: Applications of Information Theory to Machine Learning

Shan He

School of Computer Science
University of Birmingham

Outline of Topics

1. Entropy and decision tree learning
2. Mutual Information for feature selection
3. Maximum Entropy Principle

- Welcome to lecture 11. This lecture will be a bit long since I will cover three different topics, decision tree learning, feature selection, and maximum entropy principle.
- However, they are all related because of information theory.
- Our next lecture will be shorter in which I will show you the Python implementation of what we learn in this lecture.

Decision tree learning

Decision trees are one of the most popular machine learning algorithms because of their:

- Simplicity
- Interpretability: human can read and use decision tree.
- Extendability: Gradient boosting such as XGBoost and Random forest

Applications:

- Decision Trees for Credit Card Fraud Detection: "*It then creates an if/then decision tree for that account of features that do and don?t point to fraud.*"
- Google Purchases Customer Service Automation Firm Onward: "Onward uses a unique decision tree algorithm that allows its bots to be exceptionally functional in delivering world-class customer care."
- AI Reveals The Perfect Pancake Recipe.

- Decision trees are one of the most popular machine learning algorithms. They are popular due to two reasons, first, it is very simple, as you will see in the lecture. Some of the online materials such as wikipedia are over complicated, but using the knowledge we learned, you can see it is very simple.
- Second, because of its simple model representation, i.e., a tree structure, we can easily understand and use a trained decision tree.
- Thirdly, this simple method is the base learner of many most power and popular algorithms, such as random forest and XGBoost, which are the winners of many machine learning competitions such as Kaggle competitions.
- There are also many successful industrial applications of decision trees. Here are I listed a few.
- For example, this Customer Service Automation Firm Onward use decision tree algorithm to construct a chatbot for customer service. The company Onward was acquired by Google in 2018.
- Last week, Feb. 16th was the pancake day, which was also my older son's birth day, and here is an relevant news: an AI company used random forest, a decision tree ensemble algorithm to find the perfect pancake receipt. Maybe you can try it yourself.

Decision tree learning

Predictive model: a tree structure which consists of:

- Root or internal node: a independent variable (aka. a feature or an attribute)
- Leaf: an outcome of the depedent variable
- Branch: a rule (or decision)

Types:

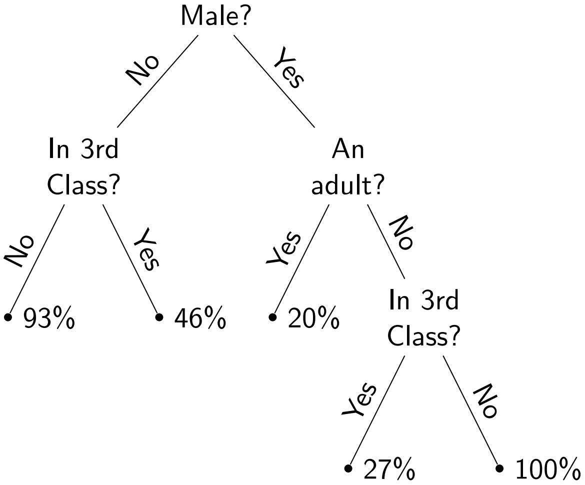
- Classification trees: dependent variables are categorical or qualitative (e.g. male/female).
- Regression trees: dependent variables are continuous or quantitative (e.g., Temperature)

Popular algorithms:

- ID3 (Iterative Dichotomiser 3)
- C4.5 (successor of ID3)
- CART (Classification And Regression Tree)

- So what is a decision tree?
- It is essentially a tree structure with nodes, leafs and branches that connect them.
- Each node, either a root node or an internal node represents an independent variable, also know as a feature or an attribute
- Each leaf node, also known as the terminal node, presents an outcome of the dependent variable.
- These nodes are connected by branches or edges or links. Each branch presents a rule (or decision).
- Based on the dependent variables, or prediction or output in neural networks, we have two types of decision trees,
- Classification trees whose dependent variables are categorical or qualitative, for example, to predict a patient is cancer or normal.
- Regression trees are used when the dependent variables are continuous or quantitative, for example, to predict Temperature of tomorrow.
- Here a few most popular decision tree learning algorithms. After learning the basic knowledge of this lecture, you can follow the links to learn them.

Decision tree example: survival of Titanic passengers



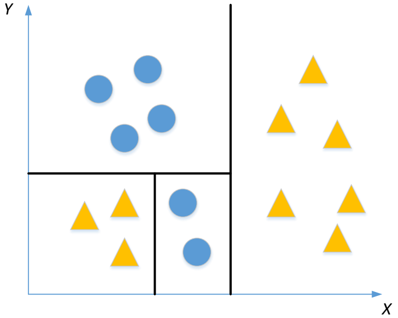
Titanic - Machine Learning from Disaster

- Here is an example decision tree. It is learned from the passenger data of Titanic, the famous ill-fated cruise liner.
- The dataset was originally compiled by the British Board of Trade to investigate the ship's sinking. The data used in this example is a subset of the original, which can be download from Kaggle by following this link.
- The task is to predict if certain groups of passengers were more likely to have survived.
- The decision tree learned from the data is shown here. Starting from the root node, we can check each unseen test sample. For an passenger who was not in the training set, we first use the independent variable Gender, to check if the passenger is male or female, and then use the branches to make decision, for example, if the passenger is a female, then we go to this internal node, to check if the passenger is in the 3rd class cabin or not. If this lady is luck, or more precisely, not poor, then she has a probability of 93% to survive.
- From the result, it seems that you would have a good chance of being rescued from the Titanic if you were a female from 1st/2nd class cabin, or a male child from 1st/2nd class cabin.

Entropy for decision tree

Question: how to construct a decision tree? How to learn the structure from data?

Answer: given a data set, the algorithm groups and labels samples that are similar between them, and look for the best rules that split the samples that are dissimilar between them until they reach certain degree of similarity.



- As you can see from our previous example, decision tree essentially is a top-down approach – given a data set, the algorithm groups and label observations that are similar between them, and look for the best rules that split the samples that are dissimilar between them until they reach certain degree of similarity.

- This can be illustrate by this toy problem with two classes, blue circles and yellow triangles.

- Intuitively, we can first split the samples by the value of X, which is in the middle. Now the samples on the right hand side are all yellow triangles, which we do not need to split further.

- For the samples on the left hand side, we can split the samples by the value of Y, after which the top half are all blue cycles, and finally, we split the bottom half by the value of X. Now all the regions contain the same samples.

Gini index and Information gain

Question: How to look for the best rules that split the samples?
Answer: two main methods:

- Gini index (aka Gini impurity): the simplest one, used in CART (Classification And Regression Tree)
- **Information gain** ✓: used in
  - ID3 (Iterative Dichotomiser 3)
  - C4.5 (successor of ID3)

Gini index: Give a training dataset of J classes, it is defined as

$$I_G(p) = 1 - \sum_{i=1}^{J} p_i^2,$$

where p_i is the fraction of items labeled with class i in the dataset. For more details, please read this Nature Methods 2017 paper: CART (Classification And Regression Tree)

- However, the above example is extremely simple, only two independent variables, so that we can visualise the samples and eyeball the visualisation to make decisions ourselves.
- However, with a complex dataset, for example, higher dimensional data that consists of more than three, or even hundreds and millions of independent variables, how to find the best rules to split the samples?
- There are two main methods, the first one is called Gini index or Gini impurity, which is the simplest. I shall only mentioned here briefly since it is very simple.
- However, it is worth mentioning that Gini index is related to Tsallis Entropy, a new entropy measure that is still controversial in the scientific literature.
- The other one is called information gain, which is used in ID3 and C4.5 decision tree training. It is more complex than Gini index, but closely related to what we learned in our last two lectures, information theory, so I will spend more time on this.
- Ok, just briefly, let me introduce Gini index. **READ SLIDES**.

Information gain for decision tree

Information gain: the information we can gain after spiting the samples based on a independent variable (internal node)
Formally, information gain is defined as the change in information entropy H from a prior state to a state that takes some information as given:

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

where Y is a random variable that represent the dependent variable and X is one of the independent variable, and $H(Y|X)$ is the conditional entropy of Y given X .

- Now, let's learn information gain.
- Informally, as the name suggested, information gain is the information we can gain after spiting the samples based on one particular independent variable (internal node).
- Formally, information gain is defined as the change in information entropy H from a prior state to a state that takes some information, which is given by this equation $IG(Y, X)$ eqauls to the entropy $H(Y)$ minus the conditional entropy $H(Y|X)$, where Y is a random variable that represent the dependent variable and X is one of the independent variable.
- I shall explain this equation using a toy example of classification.

Example: Cancer diagnosis

We took some biopsy of some tissues of 5 patients who are suspected to have breast cancer. We measured three biomarkers, which are either positive (T) or negative (F). 3 out of these 5 patients are confirmed as cancer by oncologists, and labelled as 'T'. The three measures and labels are a set of training samples which are tabulated below:

Tissue ID	Biomarker A	Biomarker B	Biomarker C	Label (Cancer?)
1	T	T	T	F
2	T	F	T	T
3	F	F	T	T
4	F	T	T	F
5	F	T	F	T

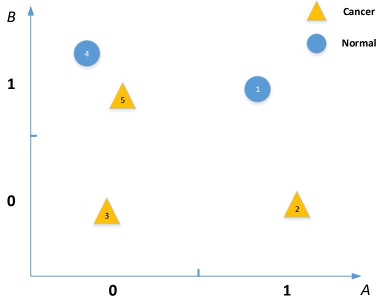
Question: How to train a decision tree? Or more precisely, how to split the samples by independent variables?

- Here is the toy example. **READ SLIDES**
- The question is, how to train a decision tree so that we can classify patients into normal or cancer based on the three biomarkers?
- This essentially is a question about how to split the samples according to the independent variables. Or more precisely, which independent variable to choose so that based on the value of this independent variable, after splitting, the samples of the two parts are dissimilar between them?
- We have seen a similar situation in page 6 **(TURE PAGE)**, where to do the first split? X or Y? In that toy example, it is obvious since there are only two independent variables. But here is slightly more complex since there are three independent variables, Biomarkers A to C.
- Is there a systematic and algorithmic way to decide how to split?

Example: Cancer diagnosis

Solution: We first use a random variable Y to model the depedent variable (i.e., label) with the range as $R_Y = \{0, 1\}$ and 0 means normal and 1 means cancer, and each of the independent variables, i.e., the biomarkers is a random variable, denoted as A, B and C, with range $R_A = R_B = R_C = \{0, 1\}$, respectively. Then we follow the following steps:

- Step 1: Calculate the entropy of random variable Y before the split



- To construct a systematic and **then** algorithmic way to split, we first need to have mathematical rigorous in our problem formulation.
- Let's use what we learned in our previous 3 weeks' lectures.
- We first use a random variable Y to model the depedent variable (i.e., label) with the range as $R_Y = \{0, 1\}$ and 0 means normal and 1 means cancer, and each of the independent variables, i.e., the biomarkers is a random variable, denoted as A, B and C, with range $R_A = R_B = R_C = \{0, 1\}$, respectively
- We now have a rigorous mathematical formulation of our problem to find out hwo to choose a independent variable for splitting.
- Now, we can derive some algorithmic steps to solve the problem.
- The first step is to calculate the entropy of random variable Y before the split
- Here I visualise this dataset but only for two independent variables, biomarkers or random variables A and B. Please remember there is the 3rd dimensions, biomarker C, which we omitted here. You can regard this plot as a projection from 3d space to 2d plane.
- Just by eyeballing, you might already notice, splitting samples using B is better than using A. Can the information gain tells us this?

Example: Cancer diagnosis

Step 1: Calculate the entropy of random variable Y before the split

$$H(Y) \equiv -E\left[\log P_Y(y)\right] \equiv -\sum_{i}^{n} P(Y = y_i) \log P(Y = y_i)$$
$$\equiv -\sum_{i}^{n} p(y_i) \log p(y_i)$$

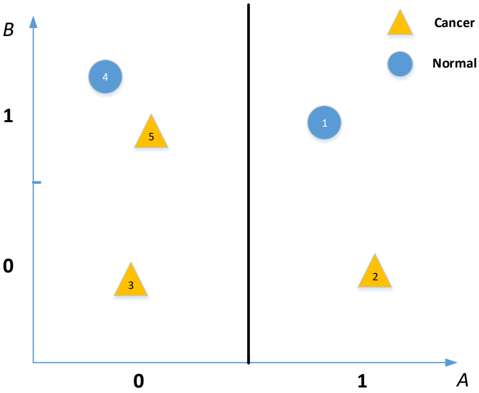
$p(0) = \frac{2}{5}$ and $p(1) = \frac{3}{5}$, we have

$$H(Y) = -[p(0) \log p(0) + p(1) \log p(1)]$$
$$= -\left[\frac{2}{5} \log \frac{2}{5} + \frac{3}{5} \log \frac{3}{5}\right] = 0.971$$

- Let's follow the steps to calculate the information gain.
- The first step is to calculate the entropy of random variable Y before the split.
- This is the entropy equation as we learn in lecture 9, week 3.
- We need to know the probability of $y = 0$ and $y = 1$, which is easy, since there are 2 normal and 3 cancer cases, we can obtain probability $P(Y = 0)$ and $P(Y = 1)$
- Then use the entropy equation, we can calcualte the entropy is 0.971, very close to 1, that is random.

Example: Cancer diagnosis

Step 2: Calculate the conditional entropy after a split. Since only three independent variables, we use brutal force, i.e., go over all three biomarkers:



- The second step is to use the value of three independent variables or the random variables A to C to split the samples and calculate the information gain after splitting.
- We can do so is because we have only 3 independent variables, we can use this brutal force method to calculate all 3 independent variables. If there are too many independent variables, then we might need to use some heuristic search to speed up the calculation.
- We start with random variable A or biomarker A. Since A is a binary discrete random variable, it is very straightforward, we split samples into two parts according to the value of A, that is 0 or 1.
- This is can be visualised by this figure.

Example: Cancer diagnosis

Step 2: Calculate the conditional entropy after the split. Since only three independent variables, we use brutal force, i.e., go over all three biomarkers and starting with A:

$$\begin{aligned} H(Y|A) &\equiv \sum_{a_i \in R_A} P_A(A = a_i)\, H(Y|A = a_i) \\ &= -\sum_{a_i \in R_A} P_A(A = a_i) \sum_{y_j \in R_Y} p(y_j|a_i) \log\, p(y_j|a_i) \end{aligned}$$

Since $P_A(A = 0) = \frac{3}{5}$, $P_Y(Y = 1|A = 0) = \frac{2}{3}$, $P_Y(Y = 0|A = 0) = \frac{1}{3}$,$P_A(A = 1) = \frac{2}{5}$, $P_Y(Y = 1|A = 1) = \frac{1}{2}$, and $P_Y(Y = 0|A = 1) = \frac{1}{2}$, we have

$$H(Y|A) = -\left[\frac{3}{5}\left(\frac{2}{3}\log\frac{2}{3} + \frac{1}{3}\log\frac{1}{3}\right) + \frac{2}{5}\left(\frac{1}{2}\log\frac{1}{2} + \frac{1}{2}\log\frac{1}{2}\right)\right] = 0.951$$

- Here is the calculation. First, we write down the conditional entropy equation, which we learned from our previous lecture, lecture 10.

- Since the shorthand notation will be very confusing, I use the full math notation here.

- We need to get the probabilities when $A = 0$ and $A = 1$, which can be easily see from the table, or from the figure **(TURN TO PAGE 12)**. There are 5 samples, 3 samples with $A = 0$, and 2 samples with $A = 1$, which gives us $P(A = 0) = 3/5$ and $P(A = 1) = 2/5$, respectively.

- We then calculate the conditional probability $P(Y = 1)$ given $A = 0$. From the figure **(TURN TO PAGE 12)**, we can see that, when $A = 0$, the probability of a sample is cancer, that is $Y = 1$, is $2/3$, and being normal, that is the conditional probability $P(Y = 0)$ given $A = 0$ is $1/3$. Similarly, we can obtain the conditional probability $P(Y = 1)$ given $A = 1$ as $1/2$ and the conditional probability $P(Y = 0)$ given $A = 1$ as $1/2$ as well.

- Then we can use the conditional entropy equation to obtain its value as 0.951.

Example: Cancer diagnosis

Step 3: Calculate the information gain for split A:

$$IG(Y, A) = H(Y) - H(Y|A) = 0.951 - 0.971 = 0.02$$

Step 4: Repeat Step 2-3 to calculate the information gain of all splits. **Results**:

$$\begin{aligned} &\text{Split 1: } IG(Y, A) = 0.02 \\ &\text{Split 2: } IG(Y, B) = 0.419 \\ &\text{Split 3: } IG(Y, C) = 0.171 \end{aligned}$$

Candidate Split 2 has the highest information gain, so it will be the most favourable split for the root node.

- Now we can calculate the information gain after splitting the samples according to the value of random variable A or Biomarker A, that is 0.02.

- In contrast, after calculating the information gains for the other two random variables, B and C, we found out that they are both larger than that of splitting using A. The information gain from splitting on B is the largest, which you make sense from the figure **(NEXT PAGE)**

Your first decision tree

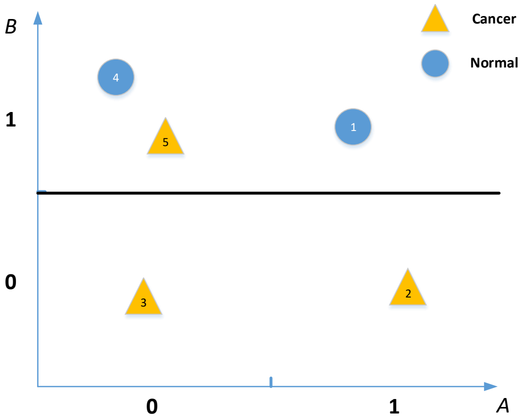
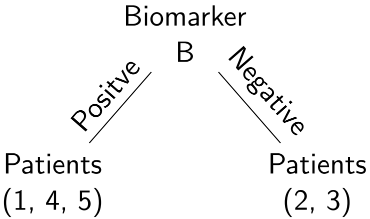


Figure 1: Splitting samples based on the values of random variable B generates the largest information gain.

Your first decision tree



Note: This example is oversimplified, e.g., the decision rules for splitting of samples, i.e., the branches, are directly based on the value (Positive/Negative) of the independent variable, because it is a binary discrete random variable as same as the dependent variable. However, for general discrete random variables with more than two values (called categorical variables) and continuous random variables, we need to search the best value, called cut-off, cut-point or threshold that maximise the information gain.

- Now, we will use random variable A as the the root node, and you obtained your first decision tree by hand calculation.
- However, I should point out that, this example is a oversimplified, e.g., the decision rules for splitting of samples, i.e., the branches, are directly based on the value (Positive/Negative) of the independent variable, because Biomarker B is a binary discrete random variable as same as the dependent variable, the label cancer or not cancer.
- However, for general discrete random variables with more than two values (called categorical variables) and continuous random variables, we need to search the best value, called cut-off, cut-point or threshold that maximise the information gain.
- **(GO TO PAGE 6)**. Here, in this example, we need to decide the value of X, for example, $X > 0.5$ then right-hand side part of those yellow triangles $X \leq 0.5$ for the left hand side part to split the samples into two parts.

Drawbacks of decision trees

Drawbacks:

- Unstable: a small change in the data can lead to a large change in the structure of the optimal decision tree.
- Relatively inaccurate: Many other predictors such as Support Vector Machine and Neural Networks perform better than decision trees with similar data.

Solutions: Decision tree ensembles:

- Random forest
- Gradient boosting such as XGBoost

See An Introduction to Statistical Learning: Chapters 8.1-8.2

Questions: Can we go beyond decision trees?

- There are a few notable drawbacks of decision trees.
- The first drawback is they are unstable, meaning a small change in the data can lead to a large change in the structure of the optimal decision tree.
- Researchers have already proposed solutions to address these two main drawbacks.
- The most direct and successful solution is decision tree ensembles, that is, we construct multiple decision trees and then combine them to get better stability and accuracy.
- The theory behind this practice is called ensemble learning theory, which is beyond the scope of this module/
- However, if we think out of the box, do we really need to use decision trees? Can we just learn from decision trees but go beyond decision trees?
- But to do so, we need to get to the bottom of the decision tree learning. In essence, we know it is just a collection of rules to split samples. The splitting is based on some measure, in particular, we learned the information gain measure. So what is it really? Can we have a deeper understanding by relating it to what we learned in information theory?

Information gain actually is...

Question: look at the equation of information gain:

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

what do you remember?

Information gain actually is...

Answer: Mutual Information

$$\begin{aligned} I(X;Y) &\equiv H(X) - H(X|Y) \qquad (1)\\ &\equiv H(Y) - H(Y|X) \qquad (2)\\ &= D_{\mathrm{KL}}(P(X,Y)\|P(X)P(Y)) \qquad (3) \end{aligned}$$

Interpretation: decision tree learning algorithms (ID3 and C4.5) recursively use mutual information to select the independent variable that share the most information with the dependent variable, then split (make decision) the samples based on the value of this independent variable.

- You might remember what we learned from our previous lecture, mutual information. That's exactly equation (2).
- Recall what we learned, mutual information is to measure the information that two random variables X and Y share.
- As we discuss, this mutual information can be also interpreted as the KL divergence between the joint distribution $P(X, Y)$ and the product of marginal distributions $P(X)$ and $P(Y)$. That's why people also regard informaton gain as KL divergence.
- So, the essence of decision tree learning algorithm is to use mutual information to select the most informative independent variable that share the most information with the dependent variable, then make decision to classify the samples. Since after the first split, i.e., the root node, we then recursively split the samples using the same principle, for both split parts until they reach some degree of similarity.
- It is essentially a sequential process. Such a sequential process can explain why decision trees are not stable, the first drawback we mentioned in page 16, that is if the data change, the mutual information might change, the order of those independent variables will change, so that the whole tree structure will change dramatically.

Mutual Information Feature Selection

Question: Why not select a set of most relevant independent variables based on mutual information and use any other powerful machine learning algorithms to build predicted models?

Mutual Information feature selection: Use mutual information to choose a optimal set of independent variables, called features that allow us to classify samples. Formally, given an initial set F with n independent variables, $\boldsymbol{X} = \{X_1, X_2 \ldots, X_n\}$, find subset with $S \subset F$ features that maximises the Mutual Information $I(Y; S)$ between the dependent variable Y (label), and the subset of selected features S.

- Wikipedia: Feature selection
- Scikit-learn: Feature Selection

- Now after understanding the essence of decision tree training and the root of its drawback, we ask, why not select a set of most relevant independent variables, called features based on mutual information and use any other powerful machine learning algorithms to build predicted models?
- This idea is a valid idea, since selecting important features is an important sub-field in machine learning. There are many feature selection algorithms exists which I will not cover in this module. If you are interested, you can follow the links to read more about feature selection.
- However, the most relevant method, which is implicitly related to decision trees learning but directly related to information theory is the mutual information feature selection.
- Informally, the main idea is to use mutual information to choose a optimal set of independent variables, called features to classify samples.
- Formally, given an initial set F with n features, $\boldsymbol{X} = \{X_1, X_2 \ldots, X_n\}$ (independent variables), find subset with $S \subset F$ features that maximises the Mutual Information $I(Y; F)$ between the dependent variable Y, and the subset of selected features S.

Mutual Information Feature Selection

Mutual information feature selection

```
**Initialisation:** Set $F \leftarrow \boldsymbol{X}$ and $S \leftarrow \emptyset$
    $f_{\max} = \operatorname{argmax}_{X_i \in \boldsymbol{X}} I(Y; X_i)$
    Set $F \leftarrow F \setminus \{f_{\max}\}$ and $S \leftarrow f_{\max}$
    **Repeat** until $|S| = K$:
        $f_{\max} = \operatorname{argmax}_{X_i \in F} I(Y; X_i) - \beta \sum_{X_s \in S} I(X_s; X_i)$
        Set $F \leftarrow F \setminus \{f_{\max}\}$ and $S \leftarrow f_{\max}$
    **End**
**End**
```

[0] R. Battiti, "Using mutual information for selecting features in supervised neural net learning," **IEEE Trans. on Neural Networks**, 5(4), 1994.

- Here is the pseudo code of a simple mutual information feature selection algorithm published 26 years ago. Let me explain how it works.
- The basic idea is Sequential Feature Selection, or more precisely, the so-called forward Feature selection, meaning you start with an empty set of important features, called set S and then greedily add the most important feature into this empty set, that's exactly the initialisation step.
- The second step is to find the most important features from the n features $\boldsymbol{X}$, which is the second line. In this equation, we find the feature $f_{\max}$ which achieves the maximum mutual information I among all the independent variables. ||| **(POINT TO 3rd LINE)** Then you add this most important feature into the set of selected features, S and then subtract it from the set of all original features. This line of code select the first and the most important feature.
- Then we execute a loop to select K features. In this loop, we also find the feature $f_{\max}$ which achieves the maximum mutual information I among all the remaining independent variables in set F.
- However, because some features highly correlated with each other, selecting them will increase the number of features but does not improve the prediction. Therefore, we need to make sure there the must be minimal redundancy between the candidate feature X_i and the set of selected features S. That's exactly the second term of the equation (POINT TO $f_{\max}$) ||| You then add this feature into S and then subtract it from set F and repeat until we got K features.

Mutual information feature selection

Further readings:

- Normalized Mutual Information Feature Selection, *IEEE Trans. on Neural Networks*, 20(2), 2009.
- Conditional Likelihood Maximisation: A Unifying Framework for Information Theoretic Feature Selection, *The Journal of Machine Learning Research*, 13, 2012.

Maximum Entropy Principle

Maximum Entropy Principle:

- Jaynes' Maximum Entropy Principle
- Shan's blog
- Duality of Maximum Likelihood and Maximum Entropy

Fundamental question: How information, energy and matter, the three fundamental elements of our universe, give rise to life?

- Unfortunately, we ran out of time to explain an very interesting principle in information theory, that is Jaynes? Maximum Entropy Principle.
- It is also because you need to know a constrained optimisation method called Lagrangian method, which has not been taught in undergraduate modules.
- Therefore, I have to leave this topic as further reading if you are interested. This principle has deep connections between information theory, probability theory, Bayesian learning, which we will learn from our next lecture, physics, life science and even philosophy.
- I recommend this paper which is freely available online. I also wrote a blog about this principle.
- This paper shows the relationship between maximum entropy and maximum likelihood method we learned in lecture 5, that is maximum entropy is the dual problem of maximum likelihood method.
- I hope reading this papers will give us some clue to answer a fundamental scientific question: How information, energy and mass, the three fundamental elements of our universe, give rise to life?

Exercise

Task: After learning how to implement decision tree learning algorithm to predict Titanic passenger survival in my next lecture, implement the mutual information feature selection algorithm and construct a decision tree. Compare it with the decision tree learned from data without feature selection.