[Music]

Welcome to the third lecture of the fourth week to the course of a machine learning this lecture will be about decision tree learning algorithms. For practical reason we'll break this lecture down into two parts, this is part one. The agenda for this lecture is as follows you'll first talk about decision trees in general and then we will focus on a specific kind of algorithms called top-down induction of decision trees algorithms, and then we will talk about some information ,theoretical measures needed for making crucial decisions in these algorithms. Then we will have this break for part two and then in part two we will look into the ID3 algorithm which is a default or prototypical algorithm of this kind and after that we will talk a little about one of the key problems for this kind of algorithms which is over-fitting and how to manage that problem through something called pruning, and then finally we say a few words of some alternative algorithms within the same category.

The challenge here is to design a decision tree such that the tree optimizes the fit of considered data items and the predictive performance for still unseen data items, we recall the minimum prediction error. Decision trees analysis are two main types, so on one hand we have classification tree analysis when the leaves are labelled according to the **k** Target classes included in the data set. Regression tree analysis when the leaves are real numbers or intervals. Decision trees represent the disjunction of conjunctions of constraints on the future value of our instances, but the decision tree can also be seen as an equivalent to a set of if-then rules I mean if the rules are also many times used for decision situations. So in this case each branch represents one if-then rule where the if part responds to the conjunction of each tests on the nodes from the root to the leaf and the then part corresponds to the class label or numerical range of that branch.

Decision trees and the corresponding learning algorithms have some positive and negative properties if we looked at the positive side first, this kind of formalism is very easy to interpret for humans and as a consequence there's also many aspects of the algorithms are easy to grasp and follow. It's a very compact formalism it's also very natural to handle irrelevant attributes you will see that when we go into the early one of the key things to handle the choice of attributes. And also there are reasonable ways of handling missing data and other kinds of noise, and these are ways are also very fast at testing time, of course there are some restrictions so one thing is of actually that the only way the decision tree can split the data items is by following the axis of the feature space. So the divisions of the feature space is actually rectangles that follows the dimensions of the feature space. Another thing is

that typically these algorithms are greedy in the sense that I always try to find an optimal local decision, which means that and they don't backtrack , they don't they never go back and change earlier decisions. So this means that because they are greedy, they may not find the globally optimal tree.

So we're learning from this representation, we will now focus on a particular category of learning techniques called top-down induction of decision trees TDIDT. The scenario for learning here is supervised non-incremental data-driven learning from excerpts and we have touched all these subjects earlier so I hope you get the picture. The systems are presented with a set of instances and develops a decision tree from the top down guided by frequency information in the examples. The trees are constructed beginning with the root of the tree and proceeding down to its leaves. The order in which instances our handle are not supposed to influence the build up of the tree. The systems typically examine and re-examine all of the instances as many stages during learning, so actually all the instances have to be stored in this process. Building the tree from the top and downward, the main issue is to choose and order features that discriminate data items in an optimal way. Subtopics that we will touch is the follow are the following, use of information theoretic measures to guide the selection and ordering of features, how to avoid underfitting and overfitting by pruning on the tree, generation of several decision trees in parallel is another approach an example of that is called random forest, and finally we will say something about how you can build in some kind of inductive bias in the algorithms. One property of nodes in a decision tree that is important for the rest of our discussion, is the concept of purity or homogeneity. So when we start a buildup of a decision tree, and we start with a data set and we start with the root of a tree and initially the entire data set that all training instances the entire dataset is associated with a root so this you can see in the picture the small example to the right. But for every decision split we do, in this process or building up the tree based on the shows and feature and its values, the data set is partitioned and the subsets of the datasets become associated with the nodes of the split, the new nodes and this is repeated recursively down to the leaves through the series of decisions we make. So impurity or homogeneity refers to the distribution of data items of the **k** target classes they are symbolized by colors in simple example to the right, both for the root and for each of the nodes. Less degree of mix of classes for all the particular nodes, implies higher purity for the nodes. So this means that if the elements over an node, the instances associated with the node is of just one class, you purity maximum purity but if you have early even mix, an even balance of all the classes involved, if your two class you can say you have 50/50 if you have three classes you one-

third one-third one-third then you have a minimum purity or maximum impurity. Sometimes you people want to talk about homogeneity instead but in this presentation I highlight the term purity. So most algorithms in this genre aims at maximizing the purity of all nodes and of course then we need some means for measuring the impurity and we will now turn into a few alternative schemes for measuring the purity or impurity.

As you understood, we need some information theoretic measures ways of measuring the decision tree nodes and the associated instances in order to be able to understand which is the optimal choice of features to use for discrimination when building up the tree. So what we will do now a look at a few two very related but slightly different schemes for doing so. One is we call information gain based on entropy and the other scheme we call the Gini approach. There are also other ways of doing this there's something called variance reduction but we will not too further about that I mean. Most of these techniques are more or less equivalent there may be slight difference when we apply that one or the other for very in various applications, there will also be some differences in computation efficiency some may be better but more computational expensive and vice-versa, but I think for the purpose of this course the most important thing is to understand that in order to build a decision tree according to this kind of methodology, you need some kind of measure that we will exemplify with two cases.

I wanna talk about one of the most frequently used information to take measures in this kind of algorithms and is often referred to as information gain and entropy measure. So that two parts, here so the first part if we start from the top is the information gain. So really we want to measure that guides us in choosing the feature with the best discriminative power in each decision point, and as we said earlier and actually the goal is to maximize the purity of nodes, minimize the impurity of nodes. And we want to a measure that tells us whether which feature is as the greatest discriminative of power. So actually the idea with the information gained is to say oh look if we have a measure of purity on node where we are let's look at what happens if we take this feature and split into in respect to its value, then look at the subsets or instances that we determine attributed to each a new node that we branch from the node we are. And let let's look at what happens to the entropy, if we do this split, then looking at the entropy of course of overall the new subsets of nodes associated with the split values. So this is essentially the idea of information gain but first of all of course need a measure of purity or impurity. So the measure of purity here is in this approach is called entropy, borrowed from information theory and actually it's defined as for the binary classification case we have two classes or one class and then we are positive example of that

and negative example of that. So the measure or purity here is chosen to be the negation of the probability of a positive example times the binary logarithm of that probability and plus the probability of having a negative example time to a binary logarithm of that probability. So this is the entropy definition.

And for binary reclassification we have many classes is just the negation of the sum of all the probabilities for having a specific class times logarithm of each of those classes. Okay, so then the only thing that we need then is to create a formula where we infer the gain, to show the future at a certain point based on the split of this feature. So you can see on the slide you can see this formula saying the gain at a certain point which corresponds to a certain set of instances, with respect to a feature is the entropy of the original point, the original node minus the sum of all the entropies for the subsets of nodes given that we split the instance with respect to the feature values of the selected feature. But moderated by the division of the cardinality of each little subset divided by the cardinality of the instances. So this is the definition of information gained based on a name to be purity measure. Actually the most frequently used measure and in the literature of this kind is the information gained measure with entropy as we described in the in the last slide. The alternative used by some people are called the Gini impurity measure and then we have something we call the Gini gain, which as you can see on the slide the differences is not so much in how we decide how we define the information gain, because the information gained is we define with the same principle doesn't matter what kind of measure we have of the impurity. So the difference here is how we measure the impurity of a certain set of instances. So you can see that more or less the nation of Gini gain instead of the general information gain is the same, the difference here is that there is no Gini impurity. So Gini impurity is a measurement of the likelihood of an incorrect classification of a new instance of a random variable if that new instance were randomly classified according to the distribution of class labels from the date set. And this becomes then one minus the sum of the squares of the probabilities for the instance belonging to a certain class. Mmany times the Gini method gives more or less the same result as the other. As you can understand is a little simpler to compute because you don't have to compute lover but I take it up here because it's often mentioned and I think the message is it's not so important it's use if you don't have specific knowledge and really go into the details, but I want to mention it because it occurs in the literature and the important message is there the summary is to say you need this kind of measure in this kind of approach and these are the two the ones I've described here are the two most commonly used.

I will now introduce an example and so I can exemplify how these measures can work out. So this example we have 14 instances, the training instances and each instances is expressed in terms of four features, outlook, temperature, humidity wind, which have had a few feature values and then we have a binary classification, for this task which whether you should perform certain activity or not. So this is the case we have.

Let's now look at the example and how we compute the various measure. So then the first thing to do of course is to look at the root of all potential tree for this dataset, and as you remember there were 14 data items in the data set, nine of them were positive which means that it's a positive outcome they're performing something like playing tennis and five negative. So then we can compute the entropy of the root node and you can see this done here according to the earlier given formulas. Then we illustrate on the slide how we can try to evaluate the information gained for one specific feature let's say the wind, which has a strong value and a weak value. So for the weak value we have to look at which instances would belong to a node branched out based on that feature value and actually it turns out to be eight instances in total, six of them positive two of them negative. In the other case we would have six instances, three of them positive three of them negatives. Okay, so then when we use the formula introduced the gain, the information gained by splitting according to the to the wind feature the that information gain will be the entropy for the whole set that we already calculated, minus the sums of the entropies for these two subsets that we described about moderated by the cardinalities of the two sets divided by the original data set, so we end up with a value of 0.048.

If we do the same thing for all the 4 features, you can see down here for Outlook, for wind for Humidity, temperature day the highest value that we get is 0.246, which we get for outlook and therefore according to this method, outlook becomes the preferred feature to discriminate and the first feature that we are going to split from the root.

So on this slide I exemplify exactly the same thing for the Gini impurity measure as for the entropy measure. I will not spend a lot of time on this because it's more or less equivalent to the other case actually do the same thing instead of calculating the impurity of S, you have to the entropy of S you have to calculate a Gini impurity S with another definition, and then of course you also have to calculate the Gini impurities instead of the entropies for the subsets, of the subsets of the original set based on the feature values in this case, but then in the end when you want to calculate the gain it's a similar formula that you used earlier it's just that we use the Gini impurity measure instead of the entropy instead and typically we get pretty similar results with these two methods. We will now make a short break here and stop this

video, so we will continue to discuss learning for decision trees in part 2 of this video thanks a lot bye