[Music]

Welcome to the first lecture on this fifth week of the course in machine learning. So the theme of this week is called machine learning enabled by prior theories, and the purpose of this first lecture is to explain the background for the choice of theme and give you an overview of what will happen this week. When studying machine learning, it's easy to get the impression that machine learning is typically learning from a lot of examples in a kind of vaccum but from these examples the goal is to induce some abstractions or hypothesis but with very little or I was a minimal guiding information. Of course there are a lot of techniques that we look at and that we covered on the course so far that is of this character, but there are also a lot of work going on where you actually look at learning algorithms that rather learn on the border of existing knowledge which means that you actually start with a more or less substantial theory and augment or debug or this theory. Also I would say that is the case that when time goes by and this area becomes machine learning gets more mature, we will use all the experience we have had by developing these more stand-alone techniques starting with one examples but it will be more typical in the future that these techniques will be applied also in settings where we have a lot of surrounding already existing domain knowledge. So in this week I've chosen a number of sub theme to discuss, which are examples of this situation where we have a prior theory and where we're learning in way improves in various fashion that are domain theory. And I sorted these sub teams there are six of them in kind of three groups, so the first group of sub-themes have to do with the situation where we look at mixed inductive, deductive and Abductive scenario, and essentially this kind of work stems out of computer science logic tradition. And those are explanation based learning an inductive logic program and I will come back to that. So the next part which is just one theme is a reinforcement learning and I will spend considerable time on that because it's first of all it fits the theme of the week because actually here you're already have a system which is designed but you want to improve that system and also they're being considerable success in this area lately for example in gameplay, after that there will be three other techniques that I will mention more briefly. So I will talk about case based reasoning and there will be a separate lecture for that, but then in the end of this initial lecture here talk about two other sub teams learning a Bayesian belief networks and something called model-based clustering which were already touched last week in in the lecture on clustering.

Starting with the first block of sub themes I will say something now about Inference techniques. So there are three classical kinds of inference techniques, deduction, in deduction which is the classical way inference in logic you derive a conclusion from given axioms, axioms which constitute domain main knowledge and facts which are typically observations for a specific case, and then the conclusion can be derived by applying inference schemes working from the axioms and the facts using forms of inference like natural deduction for example modus ponens as a classical thing, or resolution. Then on the other hand we induction which is the other in the focus of this course, in induction you derive an axiom a general rule from the observations typically many observations and possibly some background knowledge that can guide the generalization process from the observations. And as you have hopefully understood induction is the main inference mechanism for learning but then we have a third classical inference technique called abduction, which means that from a known axiom theory and some observation you can derive a premise using a rule backwards actually. So you start from the conclusion and you and you make an argument about the premise and actually is a core element of all kinds of explanations, I mean as a great example you can look at diagnosis in medicine, and abduction also have a strong relation to causation actually use looking at fossil relations and use them backwards. So if we look at the two techniques we are going to look at now. Explanation based learning enables a spectrum of inferences from abduction to deduction. Explanation based learning is very much abduction but there are also cases where abduction in the presence of complete knowledge turns into deduction. While in inductive logic programming we start from a full deductive framework of logic programming based on resolution as the inference technique, and what has then been done to expand or generalize logic program to inductive logic programming is to see to that algorithms are developed that also enables induction to be made in a convenient way, so that you can easily in the same framework combine both deduction and induction. Explanation based learning abbreviated EBL creates general problem-solving schemata by observing and analyzing solutions to specific problems. EBL spans the spectrum from deduction to abduction, it uses prior knowledge domain theory, to explain why a training example is an instance of a concept. The explanations and identify what features this case predicates are relevant to the target concept. Prior knowledge is used to reduce the hypothesis space and focus the learner on hypotheses that are consistent with that knowledge. Accurate learning is possible even in the presence of very few training instances. There is an obvious trade-off between the need to collect many examples to be able to induce without prior knowledge and the ability to explain single examples in the presence of a strong domain theory.

Let's go to the second theme of this week Inductive logic programming abbreviated ILP. The goal of ILP is to find general hypotheses expressed as logic programming clauses, from a set of positive and negative examples also expressed in the same formalism and all these possibly in the presence of a domain theory likewise described in the same notation. Advantages of ILP as I already said now I think one of the main advantages is the possibility to express all these crucial items for learning examples hypothesis domain background the theory in the same formulas. It's also an advantage that you can a very convenient way handle deduction and induction within the same framework. It's also the case that because of the structure of logic programming where the big strength is that is easy to handle multiple relation and structured data, also inducted programming from logic programming inherent this capability which could be beneficial in many areas, one particular examples are applications in chemistry where there are more many complex stretch striker instances to handle. Also there is an opportunity in ILP but that's not unique for ILP and there is the possibility of invent new predicate sets and add them to the domain theory. So the scenario move only for inductive logic programming is that you start from a set of positive and negative training examples, expressed in an observation language. And we have a domain theory, we have a hypothesis language than the limits the kind of clauses that are allowed to express our hypothesis in and we have some kind of relation covers which works as a detector on whether a particular example is covered by a hypothesis also taking into the consideration the available background theory. And the goal given that all that is to find an hypothesis that covers all positive examples and no negative examples. The next sub theme for the week is reinforcement learning. So now we turn to a very different area and it may be it's important to say at this point that in contrast to the earlier two sub areas, stemming from computer science logic area, we now go into a methodology that is very much grounded on and inspired by long-term work in control theory. So the intuitive scenario for reinforcement learning is that you have an agent that learns from interaction with an environment to achieve some long-term goals related to the state of the environment including of course the agents itself and it learns through by performing a sequence of actions but in every step of action receiving some feedback. From the agent point of view we call the mapping from a particular state with respect to the action that can take and we call that a policy that every moment major has as a political issue see actions relative to the state they are in. And also with respect to the terminology the satisfaction of goals is defined by something called rewards. So after each single action and the environment provides a reward or rewards signal which constitutes the feedback on the appropriateness of that action. Then we have another a very important concept for all this area which we call a Return, which is it's not the feedback on a single action but it's actually the cumulation of all the rewards for a whole episode, which is the term for a sequence of actions from a state to something consider the terminal state. And the goal of most of the working area is to establish a policy that maximizes the return for all positive way of going from one state to a terminal state. And to the right you can see a simple depiction of the general framework you can also see a practical example of a maze, where a small robot is supposed to find its way, considering that that robot is embodies a variant of reinforcement learning algorithm.

The next sub team of the week is called Case Based Reasoning CBR and that's the process of solving new problems based on the experiences from solutions of similar past problems expressed in an alternative fashion seabird does solve a new problem by remembering previous similar problems and by reusing knowledge of successful problem-solving for those problems. Case based reasoning can be motivated by some such as similar problems of similar solutions normally, and that many domains are regular in the sense that successful problem-solving schemes are invariant over time. So what can say there are some considerations of invariance here as a starting point for you for using this method. And case based reasoning typically our to contrast it with rule-based reasoning so in a rule-based system you solve a problem by a fixed or maybe dynamic set as a set of rules while in case based reasoning everything starts from the cases and there is in many examples no explicit rule base, we only have a memory of cases. An interesting analogy that is always comes to mind if you look at legal systems in every country, so in many times called the German tradition or the Central European tradition law is very much based on rules while in the anglo-saxon tradition law is very much based on cases. So typically in case based reasoning cases are stored in a case space or a case memory to be retrieved and used. When a successful solution to the new problem is found an adapted case can be stored in the case space to increase the competence. Actually this is this incremental growth of the case base that constitutes the learning behavior of this kind of system. Technically case based reasoning primarily supported by the techniques described in the previous lecture on similarity based learning in an earlier week we'll also call memory based learning, and particular examples of what is really useful to inherit from that technology is these schemes for distance and similarity measures that are very crucial for retrieval of the similar cases we want to based on our reasoning on.

Now we come to the last two sub themes for this week and those are the sub themes where we will spend very little time on and will only show you a couple of slides here and comment on those themes. In the reading reading recommendations for this week you will also get some material on this kind of topics, so you may voluntarily look more into those themselves but there is actually no time this week to go further into these. So the first of these two are is learning on Bayesian belief networks. So we talked about that some weeks ago and I will shortly recapitulate for you. So Bayesian belief network is a probabilistic graphical model that represents a set of variables and their conditional dependencies describing effects in terms of courses. So structurally BBN is a directed acyclic graph or DAG you can see it smaller to the right. And in that kind of Network inferences typically aims to update beliefs concerning courses in the light of new evidence. So you do design or build up the network in some direction so we can say you could in the forward direction you could have a very typical deductive reasoning, but the more important part of it that is what you typically want is that you want to make inferences about courses in the light of new observation, new evidence. And the major theorem that supports the backward reasoning part here is called the Bayes theorem which we already discussed in the other week and this theorem makes is possible then to make valid probabilistic inferences about whether course may hold in the light of some evidence and of course then Bayesian rule controls the risk inference for one step in the network but the same procedure can be recursively applied throughout the whole structure. But then the issue is what is learning here actually one way of handling the Bayesian belief network is to statically design them so we have a fixed structure you set up a number of variables you have a node for each variable, you defined exactly the dependencies in terms of the errors and you also define fully and the conditional probabilities that guide the kind of micro reasoning in each node. But of course you can also have a situation where you start with a rudimentary structure. So for example you can have a case where you have defined the variables, you have defined the connection, so the structure is clear. However the conditional probabilities are not know so this is what is called here parameter learning, so this means that we can have a learning situation where we build up the conditional probabilities given a fixed structure from actual observations or observational pairs of all the pairs or variables involved or all the combinations of variables involved. So this we call parameter learning, and then we can of course use different kinds of specific learning techniques we already looked at for that purpose. The more ambitious task is to also handle a situation where not even the set of variables and not even the structure is known, so that we call structure learning and then there are two cases we can learn the variable structure which means that variables are known but the connections are not known. And then finally we can have a case where we can also learn new variables and you can next week see an interesting parallel here when we talk about neural networks where it's also an issue whether it's possible to learn new structures in the same fashion. So this is very shortly the goals of learning in this context and as I said you will get some reference to some material for this field and if your is interested feel free to dwell into that.

Finally I want to say something about something called model-based clustering and this I already you mentioned but pretty briefly last week when we talked about clustering on one of the lectures. So model-based clustering means that clustering is based on some model or background knowledge and actually this knowledge is for clustering typically statistical but its knowledge about the domain from which we harvest the data set. In the basic kinds of clustering techniques which we looked at last week, actually we build up cluster structure just from instances with very little or non-existent background knowledge, because of the fact that most of the time the background knowledge for this kind of in the clustering case is of statistical nature, one also call this kind of techniques distribution based clustering or statistical based clustering. The model or the knowledge variable can be more or less extensive but will in all cases guide the clustering process to some extent in contrast to the non informed variance. But it's important to say mainly it can be a little confusing because we introduced categories of clustering techniques last week but for model based clustering essentially you can start from any of the categories of clustering techniques and augment it by adding domain knowledge as a guiding principles. The most common case I would say in this area is that what we have the knowledge we have are statistical distributions regarding the objects that we look at, essentially then we about statistical distributions for the different kinds of objects which in a way then indirectly refers to the potential clusters we want to discover. So examples of these kind of methods are Gaussian mixture models where actually we have a fixed number of Gaussian distributions, that are initial randomly in the beginning but whose parameter iteratively optimized to better fit the data set, but this Gaussian distribution in a way till then are related to the different hypothesis groupings of the dataset. We also have some clustering techniques based on Bayesian statistics and it is a very well-known systems use very much practical actually in the industry called AUTOCLASS, and then finally we have something called conceptual clustering techniques with - well one well very well-known system called COBWEB. By this I am end initial lecture thanks for your attention so we will now go to the different subtopics and the next time lecture 5.2 will be on the topic of explanation based learning thank you