Machine Learning,ML
Prof. Carl Gustaf Jansson
Prof. Henrik Boström
Prof. Fredrik Kilander
Department of Computer Science and Engineering
KTH Royal Institute of Technology, Sweden

Lecture 11
Forms of Representation

So welcome to the first lecture of the third week of this course in machine learning this week we will focus on representations. So first a few words about representation I mean this is a course in machine learning so you can wonder why should whole week on representation. So I want to show you this triangle of three abstract concepts that are really the key, the backbone or concepts for any kind of work in machine learning, so we have the concepts of representation, we have the concept of problem solving and problem solving it's a little matter of taste what term you like I prefer problem solving some people would say reasoning other people would say decision making, but it doesn't really matter. And of course the representation part as such is more the core declarative part of the system but the representation in that sense is only there because it should support problem solving, so this means that when we talk about a representation it's absolutely necessary also to talk about that kind of problem solving that representation. So this means you will see that during the week you know that we will have these so called representation is use but essentially when we talk about it if we in the end be a mix or pure declarative aspect and more problem-solving aspects. Also of course because it's a machine learning course it's an important issue what would learning mean what form would machine learning take in that specific form, and this means that in several of the lectures this week we will also touch the learning aspects of these representations. So I want to comment on something that I am personally find very important, so the message here is that computer science is of course an engineering discipline and artificial intelligence therefore our is a engineering discipline as well. So actually the choice is of representation and problems of these skills are and should be based on many pragmatical engineering decisions and of course the same domain and problem can be modeled in any of the alternative form of representation and problem seems theoretically, not necessarily it is as convenient to do it in all these forms but it could be done. Also when we work of a specific domain we may use not only one of these schemes we may use several of them and we may map what we described in one's scheme on to another scheme, because there are many purposes here one purpose is to stand what we are doing is to have a kind of representation that is easy understand for humans for people involved in the whole process of working with the domain, the other aspect is that the representation should be a convenient base for an efficient implementation, so it may be so for example you will see that later that we can have a neural network representation which is then mapped totally on an array representation and then the computation which will happen in the array representation. So problem solving a machine learning for domain may be more feasible to implement in some of these schemes while some other forms may be useful for other purpose, so the bottom line here and the summary of this line is that nothing is absolute here, everything is relative representations and problems on these schemes are too in our toolbox and we have to pragmatically choose the most suitable a combination for a particular problem area. So of course one always have to make choice what is the most relevant sub team to focus on and I've made a decision to spend most of these weeks this week on five kind of representations and problem-solving schemes and the fives I have chosen are Decision trees, Bayesian networks, Neural networks, Genetic algorithm and Logic program.

So there will be one lecture on each of these things. So for each of these lectures I decided to upon a structure that I would will try to stick to, so for each of all of these representation schemes I will talk some about something general about the general characteristics about them, I will talk also a little about the kind of sources of inspiration, how they come to be developed, I will try to describe the kind of core components and in most of the cases also say something about structural aspects of the representation, finally I will look into what kind of problem solving is typical for this representation and in the end comment on what could it mean to do machine learning for this kind of representation.

So even if I did choose five sub themes for separate lectures there are other sub topics too that is worth mentioning these subtopics has typically been very very important for working machine learning and indirectly these subtopics is relevant and can come into play in in machine learning it wasn't really possible or relevant to make lecture of each one of these, so what I will do is that I will now shortly comment on these sub themes, I will start with say something about List structures and List processing, I would say something about Arrays, I would say something about Graph theory and Graph search, I will talk a little about Semantic networks and finally I'd say something about Production Rule systems.

So starting with List structures and List processing so which has been for a long time the most established formal representation artificial intelligence, the cornerstone here is to look at expressions in the form of lists where a list can be either as a number of atoms but they can also be the elements can also be lists recursively, so typically what you end up is with a gigantic structure of nested lists, the role and interpretation of the atoms and lists in various position in these structures are typically defined for each domain, this creates a great flexibility to design any kind of system you like, but all the problem is of course that the language itself like some semantics that can be shared across domains, the problem-solving paradigm coupled to lists is functional programming and that's the computational model used for computing List structures in artificial intelligence and computation here is viewed as a recurrent application of mathematical functions so you can see it does that you have recursive structure of functions working on our structure of lessons. The declarative programming paradigm program is done with expressions instead with statements, were typically called statement programming and there is a very long history for this functional program is based on work in the 1930's where something called Lambda Calculus was invented. So the most well-known language in this List is LISP which is the second-oldest program and still used and LISP has been used extensively for all kinds of applications and many times even if you start with I started with another kind of representation semantic Network production rule system and so on and you have in the end mapped those representation on to pure List structures and this base functions. And so in the end you can see an example of that in that chong'er where a very simple semantic network has been expressed in a list format. So now let's look at one form of representation that is very basic for most programming paradigm and still also very extensively used in many machine learning application and this form of representation is Arrays.

An Array is well-known phenomena it's a it's an ordered n dimensional collection of items, where these items or elements can be identified by one index or synonymously by a key, many times the word table is used synonymously for array and if you conceptually have a what you think is a table, it is very straightforward to map that onto some kind of array. Then there are some other concepts of the term vector is easier to refer to one dimensional array, also the tupple would be a better word more mathematically correct because the vector is identity of its own in in mathematics, the same way for the concept of matrix can be represented as a two-dimensional grid on and you could talk actually about two dimensional arrays as main matrices even is also matrix is a separate kind of mathematical entity really related to linear mappings. So and finally there is a certain concept that occurs in the machine learning lecture it is the word Tensor and also tensor is a special kind of mathematical entity but you may decide to talk about the multi-dimensional arrays three four five I've mentioned arrays as tensors. And I would say that even for many of the representations we will talk about in the coming later lectures many times in the end, when we in computation should take place these representations are mapped onto some array structure in the purpose of the basic computation in the form of array processing. So this second slide is just an illustration of the last actually showing the in some graphical form what is typically meant in the literature by a vector matrix and tensor in the sense of being different variants of an array with the various dimensions.

So now let's turn to another very important topic actually in mathematics graph theory is a study of graphs which are mathematical structure used to model pairwise relation between objects. A graph in this context is made up of vertices or nodes which are connected by edges or arcs. Actually graphs is a very powerful tool for modeling and also being a basis for various kinds of implementation both for problems and machine learning. So knowing something about the terminology of graphs and the properties of graphs is a background knowledge that that makes a lot of sense and is virtually invaluable in a work in this field. So apart from having a basic structure there are some distinctions so grass can be undirected or directed, the directed means that that the edge has a direction it can also be so that there are some numerical measures coupled to the vertices that you can typically call them weights, there are also issues like connectivity where there all when there are paths between specific pairs or subsets of nodes, there's also issues of whether there are cycles in a directed graph or not and there are more specific concepts like bipartite graph and certain distinctions so to know something about the issues of graphs and the terminologies related to graphs is very important because it's commonly referred to in all kinds of contexts that we will go into. So coupled to graph theory you find on this line a list of so-called the classical graph theoretical problems ranging from the old problems seven bridges of Königsberg that could be of course he easily mapped onto a simple of a graph, the Traveling Salesman problem various problems where you find to find paths with certain properties for example a path that visits each vertex exactly once or path that uses every edge exactly one onto problems whereas you to find subsets of the graph which spans all the edges. These problems may be known you may not view them as directly relevant for either artificial intelligence or machine learning but indirectly improve many problems of solving situations these variants of these issues will occur so to say in disguise because many times when you model your problem you even if you express herself in graphs of terminology or not essentially you handle your problem as if it was a crafty or a theoretical problem and therefore further using Graph Theory will occur. Closely related to graph theory is Graph Search and actually Graph search has from the start being core methodology in

artificial intelligence and of course machine learning as a sub area inherit the dependence of these techniques so many times the problem-solving will take the form of search, learning will take the form of search and therefore the vast repertoire of search strategies that has been developed is of relevance and on this slide you find a kind of overview of the kind of strategies developed historically, I will not go into great detail because this is not the main topic of this course but obviously when we come into the a later algorithm or algorithmic parts later in this course reference to search techniques, will be imminent. So essentially as you can see there are three categories of search strategies with one category called Brute Force which is essentially that they are uninformed, the second group is here called Informed which means that they are guided by some domain knowledge or heuristic knowledge to proceed and the third category is termed local search. So within each there's a main categories you have the specific types of algorithms so my recommendation is that you may look into some of these detail algorithm yourself because that will be a lot of references coming on if you are not already have a good knowledge about these issues.

So now we leave the more formal matters and look into two kinds of representations that has been used a lot for practical problem-solving historically and one of them one of these is Sematic or called semantic networks or synonymously conceptual graphs. So semantic network is a graphical formalism that represent semantic relations between concepts beside directed or undirected graph consists of vertices which have said concepts and edges which represent semantic relations between concepts. So the edges can be labelled in a bigger ways which you will see in a coming slide but there is a small set of standard age type that occurs frequently in many of the other cases developed. So IS-A was a kind of relation which corresponds to what was described on an earlier lecture corresponds to a generalization relation in a taxonomy. HAS it's a standard attribute feature relation so for example an object has a property. PART OF is what you call a component relation which means that one kind of entity is actually physically part of some other entity. Finally you have a certain kind of Causal Relations related to actions or events. A drawback with semantic Network is that there is no agreed upon notion on what given representation structure means or some formal semantics as there is no logic. So here you can draw a parallel to the list representations in in lists for example where they're also real everything is decided from case to case so if you go from one domain to another you'll find syntactically it looks the same but the intention of what is in the representation is the domain-specific , so therefore it's much more difficult to compare what is done in in different cases. They're all the ways of expressing the basic semantic networks so this means that an edge not only relates to concept but an edge can be a network of itself, which means that you can express things like a people and beliefs something and something is not just a simple fact it's a more complex situation of that complex situation can in turn be represented by a network.

So on this slide you see a few examples of semantic networks you can just have a look and the intention is of course to give you a flavour how things are expressed in this kind of formulas. And a few general messages could be that as you see as was told on the earlier slide on one hand there are a few standard attributes that occurs in many cases like generalization, relations attribute realizations, part of relation and so on, while there is also a bunch of relations or edge labels that are totally domain specific and tailored to the current situation and it should also be obvious to you that if one want to work systematically and professional with this kind of representation, it's very clear that graph theoretical issues come into play.

So finally let's turn to something termed Rule-based system. So actually rule-based systems is a kind of systems used a lot a long time in artificial intelligence to implement what was also earlier called expert systems, and there are very synonymous you can call a Rule system, IF-THEN Rule systems or Production systems and so on. The idea with Rule based system is to try to separate out declarative knowledge from procedural knowledge and the rules specify each possible inference type in a declarative way and normally there is a very self-simple some simple syntax so there is a syntax saying there is an IF part with a number of conditions and then there's N THEN part with something with a list of actions to perform if all the whole list of conclusions they satisfy. The problem with relaxed system as a sematic networks and also less representation, so on these there is no well-defined semantics for what actually is intended because you kind of obvious that's the syntax for Rule it's very simple, so actually it's up to you when you design your own based system to do define what you mean by rule, the intention of a rule so a rule could be that the IF part looks at the situation and have conditions that evaluates that situation the THEN part is a set of actions acting on that situation, but it could also be that through the if-then construct models something more close to logic where the IF part the premise part and the then part is a conclusion part and so on. All of the old rule-based system shares some core parts so a little longer is a collection of facts and the collection facts you can say represents the problem statement and then there is a collection root called the knowledge base which is a which can act upon the facts, so there is a cycle. So there is the computation is a cycle where each in each step of the cycle the collection of rules potentially work of the facts some rules becomes relevant because the conditions are satisfied so then these rules fire and there is an inference engine that handles this matching process of which rules that apply in a certain step and carrying out the act relevant actions and there are many different kinds of systems and actually you can use rules both in a forward manner and backward chaining manner I mean forward means that you go start with evidence and try to infer consequences while in backward chaining you look at consequences and try to infer what reasons or evidence is most likely to have led to these consequences, and of course always have to be some criteria for when we have rushed reached a satisfactory solution.

So on this slide you will find a graphical depiction of what can go on in in a rule-based system actually this kind of picture is not only relevant for production system but for most of the systems we look at because what is included in this picture is both as you see at the top the relation to an external situation where the system interacts with that environment with that environment but it's also a relation at the bottom where you can get again a graphical model how the development of the system can take place, so of course none of the systems we consider here it will be useful if there is not a user interface where the behaviour of the systems can be made explainable, and of course we also need kind of the developer interface where users or humans involved can add new knowledge not everything can be learned you also have to even if you have a powerful learning mechanism you have to incrementally and new knowledge to the systems in order to be useful. So the intention of this slide is only to give you a flavor for how a rule-based system can look like actually this is an example of a rule-based system that is diagnosed based is for a diagnosis system, so please have a look and hopefully you get a good picture of how things are typically done in this kind of representation this was the end of this lecture thank you very much for your attention the next lecture will be on the topic of decision trees thank you good bye