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 7
Objects, Categories and Features

Welcome to the second lecture of the second week of this course on Machine Learning. This lecture will focus on the characteristics of Objects, Categories and Features, essentially the building blocks needed in the area of classification and concept formation. This lecture will discuss the following items: first we will talk about Objects and Features, and then we will also discuss something we call the Object Space and something we call the Object Language, and furthermore we will discuss Categories and Category Structure, and what we call the Category Space and what we call the Hypothesis Language. So the first group of items refers to the way we phrase our data and the second group of items refers to how we phrase our abstractions.

So let's start to talk about Objects, and by Object I refer to the basic data items on which we base our analysis. As you see there is a very complex journal of terminology, people use very many words. “Object” I like personally as a term, but I think in many cases Data or maybe Data Item is a good word because it is understood by many people. Some people talk about Record, some people about Tuple, some people talk about Row, some people talk about Vector. People from statistics talk about Instance or Example, or Training Instance or Training Example. There are also some more exotic terms like Thing or Entity. But essentially all these terms are supposed to refer to the very concrete Data Items or Data Objects on which we base our Analysis. So then the question is how do we characterize an object. And then we come to the use of the word Feature. So in the same fashion that there are many synonym words for Object, there are many synonym words for Feature. So I highlighted Feature as I highlighted Object, but there are of course alternatives - you can talk about Property of an Object, you can talk about Attribute of an Object, Characteristic of an Object. You can talk about a Field, but a Field in a way is compatible with Record. You can talk about Column, but then that also is somewhat compatible with whether you talk about Vector or Row and so on. And then we have words which are primarily used in statistics – Variable. And for Variable there are various variants of that. We can talk about Output Variable, we could talk about Independent Variable, we could talk about a Predictor Variable. We could talk about variants of Feature, we could talk about Target Feature, we could have a Category Feature. And of course we have special semantics here - I mean a Predictor

Variable and an Independent Variable more or less is the same thing. And it is separate from an Output Variable because normally when we talk about Output Variable we mean the kind of class label that relates to the abstraction we want to create. The same is for Category Feature, so Output Variable and Category Feature could be more or less synonymous. Unfortunately I don't think anybody can assume that people will be totally consistent. I must admit that I myself switch between words. Sometimes it feels right to talk about Data Item, sometimes I feel right to talk about Object. But as long as one has the basic understanding that there are two kinds of phenomena here - on one hand there are the concrete objects and there are synonyms for that, and then are ways of characterizing those objects and then there are synonyms for that. So finally we can say that you need a formalism to express these things. It's not something very advanced. It's just a simple conventional formalism in which you express your Features, and of course some formalism in how to group Features so that they can form Object. But when we refer to this formalism of this kind of simple language in which Objects and Features can be expressed, we normally talk about the Object Language or the Data Language or the Observation Language.

So, just a few words about the types of Features. Now of course there can be many types, but the common types or normal types that we will see within this course are the following: so first of all are the Ordinal. You can call them Binary or Boolean. And then we have the Numerical which can either be discrete or continuous. And finally we have the Symbolic Features which are simple symbols, or we can have simple Structures - the most common there is of course Lists, but it could also be more complex like Graphs and so on. Here comes one example from the Zoo Dataset: one data item picked out happens to be a Buffalo and as you can see the Object Language here is a very simple one. It's simply commenting for listing the features in a row. And you can see then below already that the features here are mostly Ordinal, only one is Discrete. Two features are special because they are not predictor features, in the sense that they predict the class, that they are classification features that pre-classify each object. There are two here because on one hand the task is to classify these animals on a higher level, so the class type with values from 1 to 7 issues for that. But then there is also an animal name which is actually the name of that specific kind of animal, and because there are no duplicates here, it's over Buffalo or of some other animal. It's just the identity of that item.

So let's talk now about Object Space. So by Object Space I mean essentially the space of all possible objects as can be described by the Object Language. So depending on the language and this can be bigger or larger. But this should be separated from the set of objects that we consider. So this means with this convention the Dataset - the set of objects considered for evaluation - is just a subset normally of the Object Space. And depending on how the features are set up or chosen, this could be a dance or spares space, meaning that the subset is a pretty small subset out of the total Object Space. But there can be other words here - the Sample is a word of statistics

which also means the set of objects you actually look at, Training Sample, Statistical Sample. Then one can also use words related to Vector or Record, we can talk about Tables and Arrays. So if the data item is a row, the feature is a column then the Dataset is the table. And of course an Array is a way of representing a Table. Training Example Set is another word that can be used for a small subset of objects that we consider.

So let's take a few examples from the Zoo Dataset. So we look at the Object Space first of all again. So the Object Space or population is the set of all potential feature vectors with feature values as can be expressed in the zoo Object Language. So Objects Space in the way I phrase it here is syntactically defined by the object language. The sample or dataset in this example is the whole set of Zoo feature vectors as described on one of the earlier slides. The word that is a little tricky to use because it's inherited from philosophy is Extension because normally in philosophy extension of a concept means the set of all entities in the world. So the extension of the Concept would mean the set of all buffaloes in real life it's a tricky to use because not necessarily we have chosen our representations in such a way so that they truly represent all living creatures on a certain map, so therefore I would avoid the word extension.

So now we come to what is called the Hypothesis Space and actually how they are defined from the definition of a category. So actually if we look at the term category there are many synonyms, really a lot. So we talk about categories, we can talk about concepts, about classes, we can talk about hypothesis, we can talk about target function, we can talk about type, about schema, about model, we can look at a classifier, we can even talk, if we are a little more philosophical, about the intention of a concept. All these are words to capture the abstraction of the concept. The hypothesis language, the language for expressing the abstraction, can of course theoretically be any language. It could be different from the Object Language. But in many cases, very typically its syntax wise consistent with the Object Language. Of course there have to be, even in the simplest cases, minor additions. So for example the syntax for feature values has to be extended with suitable generalizations of the normal values. Because the hypotheses or concept in question is always an abstraction from a concrete object, so if we have a complete object expressed in terms of features with concrete values and we want to formulate that abstraction, we need some wild cards. We need some variables that allow us to generalize the feature values. Normally we have a variable background knowledge that also can constrain which kinds of abstractions are meaningful, so this constitutes a-priori knowledge outside the knowledge that we get out from our objects or from our data. So this background knowledge can normally be embedded in the hypothesis languages through simple language constraints and if we embed this knowledge in the language we normally call this a language bias. It actually is a learning bias but because we can only learn what the language allows us to learn. Finally what is

a hypothesis space? So the hypothesis space is actually the space spanned by what can be expressed in the hypothesis language.

So let's talk a little more about terminology here. So we introduced the word object, we introduced the word Category. We have a definition of a Category expressed in a Category hypothesis language. So when you're talking about it, an object is an instance of a category and of course a category is a generalization of an object. But then we also have the relations to the object space. So essentially given a specific category definition one can look at the subset of the object space consistent with that category definition, so this means that all objects that fulfills the constraints of the category definition. This space subset sub Space is normally bigger than the actual subset of the dataset consistent with the category definition because there may be a lot of objects that is not there at this point but it's still expressible in the language we created. So of course there is a subset relation between the subset of the dataset and the total subset of the object base consistent with the category definition, and of course an object is an element of the subset of the dataset. So this is just a little picture to show you these kinds of relations that exist within this terminological framework we have set up.

So you may have found the discussion around the last slide little abstract. So let's take a very simple example from the zoo data set. So actually here you find the subset of all fishes taken out from the data set. Okay. So then the question is what is a meaningful generalization of all those data items? The simplest way of handling this is of course, first of all to choose a hypothesis language which is close to the object language, which I've already earlier stated that is the normal way of going forward here. And apart from following the same instruction what we will have to do is of course to introduce some generalizations of feature valuesthat in this case we only have ordinal values, so we only need some kind of wild card here like question mark. So you can see in red and an abstraction or an abstract record, which is in this case the concept definition, where you find wild card in the positions where the subset of the dataset includes contradictory complementary values. So going back to the last discussion on terminology we have a dataset which is a subset consistent with the concept definition. Then theoretically of course we can think about that object space which is slightly larger because there are typically combinations of the wild card values not covered by the actual subset of the dataset. But it's not explicitly depicted on this slide.

So now we turn in to something different. So apart from just looking at objects and categories, we can also look at category structures which is very typical in any domain analysis, which means that we not just have a simple level of one category but that we characterize objects in a domain in some structure. And such conceptual structures or category structures also have many

names like class structure, class hierarchy, class lattice, concept structure, concept hierarchy, type structure or taxonomy. And of course it's possible in machine learning not only to learn one level of categories, but also to learn multiple levels of abstraction, which means it's possible to learn category structures. So apart from persistent and domain relevant category structures that really make sense in the domain, it is of course also possible to work with temporary structures as part of the learning process. We will come back to that, but examples of such things like Version spaces.

So obviously when we add levels here with categories on several levels of course we can generalize. So we can define specialization and generalizations, relations between all these levels.

So finally in this slide and the next slide I'm just going to illustrate for you how it can look in one case. So the most common case is when one looks at concept hierarchies, which is a tree structure. So we abstract from objects in several levels in a hierarchical way. The other variant of that is what we call the concept lattice and that is a very similar thing. The only difference here is that in a lattice it's allowed to have multiple generalizations upwards. This means that you can generalize from category six to category three or category two in parallel. Many times in reality lattice better maps on to the real situation, but technically in most cases hierarchies are more easy to handle.

So to convince you that multiple levels of abstraction, multiple levels of categories or concepts is not only purely a theoretical thing, I show you here an extension of the zoo example where I've actually just followed one path from one kind of item in our dataset - the Buffalo. And actually what you see is all the established zoological super- and sub-categories, and as you can count there are thirteen or fourteen levels from Buffalo up to animal. Of course this is a domain where Zoologists have worked for hundreds of years with the effort to find the kind of optimal taxonomy for the area, so of course it's not likely any new or more artificial domains that you'll find this depth. But I take it to show you one extreme case and illustrating that many levels of abstraction is not just a theoretical thing, but that it occurs in practice. Another observation on this line which is here in parenthesis is that if you remember the features that were used in this dataset and then compare to features that you can infer to be important from this kind of categorization you will see that they are not identical. Not surprisingly so because actually what's shown on this slide is the line from the Buffalo to the top. And of course if we only want to classify animals on that line starting from Buffalo going up to animals then a certain set of features would be optimal. In the general case for our dataset we want not only to classify mammals, we want to classify fishes and insects and so on. So this means that for that general

scenario we need a broader range of features than the dataset that is optimal only for going down the Buffalo lines so to say, so everything here of course depends on the purpose.

So finally I wanted to comment on what happens to features in this kind of more abstract category structure. So of course when a conceptual structure is formed during the learning process features will be attributed to the categories of different generality. In the simplest case we only have general category it's not an issue. But if we consider many levels, the normal way of looking at this is that every level in a way of abstraction contributes to a certain specification of certain features. Also the common way of looking at it is that when you understand what are the features of an object in a certain sub-tree or a conceptual hierarchy, and one looks at it the way that hierarchies further down the line inherit the same features as categories higher up the line. Also normally features are not normally spread evenly across the abstraction levels, most features are grouped in the mid-range of the conceptual structure, and this kind of mid-range is termed the basic level. Of course it all depends on what you want to classify and how you arrange your structures. But anyway it doesn't really matter how you do it because depending on how you do it, it's possible always to say that some level is basic. And a very simple example you can see here - so if we look at various kinds of fruit trees, in the middle we have trees like apple trees, peach trees, grape trees and so on. More abstract is fruit trees. More specific are specific kind of apple trees like Mackintosh trees, Delicius trees and so on. And then it turns out that if you think about what kind of characteristics you can attribute to each level, you will normally find that it is more natural and easier to attribute categoristics to what is here termed the basic level. Of course again with the disclaimer that is the basic level is something relative to the way you set up the category structure.

So, we reached the end of this lecture. Thanks again for your attention. The next lecture will be on the topic “Feature Related Issues. Bye and thank you.