3.2 Versionspace:

Using the find s algorithm, we were able to just get one hypothesis at the end and it was not clear, if it is consistent with the training data until it finishes executing.

3.2.1 Version space : from grundlagen des machinles lernen

The version space $V^{H,D}$ of an hypothesis space H and an example dataset D contains all the hypothesis that are consistent with the training dataset . On the contrary of using the find s algorithm that retrieves just one hypothesis consistent with the training data, the version space learning process retrieves all the hypothesis that are consistent with the data.

$V_B = \{ h \in Lc \mid h\ is\ correct\ and\ complete\ regarding\ D \}$

Example:

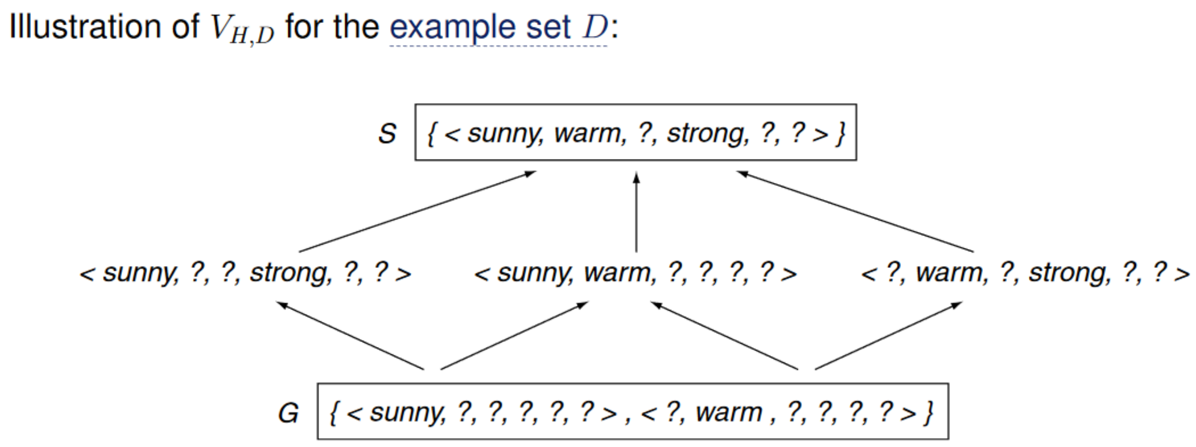

Illustration of $V_{H,D}$ for the example set D:
S
{ < sunny, warm, ?, strong, ?, ? > }
< sunny, ?, ?, strong, ?, ? >
< sunny, warm, ?, ?, ?, ? >
< ?, warm, ?, strong, ?, ? >
G
{ < sunny, ?, ?, ?, ?, ? > , < ?, warm , ?, ?, ?, ? > }

Fig 3.2: The version space of the training data

The above figure shows the version space of the training examples D in the figure 2.4. It can be seen that the boundary S contains the most special generalizations of the version space. On the other hand, the boundary G contains the 2 most general generalizations. The arrows address the relations between the above hypotheses. The above version space contains 6 distinctive hypotheses. Another depiction of the version space can be accomplished through the two border sets S and G, on the grounds it is conceivable to generalize and count all hypothesis that lie between these two boundaries. For this to be done, the “general-as” relation discussed in the chapter 2.7 can be used. [literatur von brewka].

The boundaries S and G are forming the borders of the version space that contains all the hypothesis that fit the training data and the hypothesis they contain are the most **special** and most **general** generalization of the example set. A hypothesis h can be called as the most special generalization of the dataset if and only if it is complete and correct regarding the training data and as well as there should not exist any other hypothesis h’ that is complete and correct regarding the data and h’<h. The same can be applied to a hypothesis in order to be a most general generalization of the training data except that there should not exist any hypothesis that is more general than the hypothesis h’>h.

Regarding the most specific boundary in the figure 3.2, it be claimed that the hypothesis <Sunny, Warm, ?,Strong,?,?> is the most specific generalization , as it is correct and complete and there does not exist any other hypothesis in the version space that is more specific than this. Viewing the most general boundary above, it can be said that the two hypothesis contained in the general boundary are most general generalizations , as they are complete and correct and there does not exist any other hypothesis in the version space that is more general than them.

3.2.2 Candidate elimination algorithm:

After the concept of the version space has been defined and explained in the preceding chapter, it is very fundamental to explore the candidate elimination algorithm that is being used in order to construct the version space. Given the set of the training examples E and the hypothesis space V, the candidate elimination algorithm builds incrementally the version space. The training examples will be iterated over one by one and each training example may shrink the hypothesis space by getting rid of the hypothesis that are inconsistent with the considered example. After considering each training example, the general and specific boundaries will be updated.

S and G, which are the boundary hypothesis, represent the version space V. For each new example S and G must be checked and if necessary be adjusted. For a considered example e, if there is a hypothesis h belonging to S or G and it is consistent with the example i.e. it applies that h(e) = 1, if e is a positive example and h(e) = 0, if e is a negative example, in this case , the boundaries should not be altered. If it is the case that h is inconsistent with the considered example, then two cases can be taken into consideration. Either e is a positive example and although e is mapped wrongly to zero by the hypothesis h(e) = 0 or e is a negative example and it is mapped wrongly to 1 by the hypothesis

<lerre,lerre,lerre,lerre,lerre,lerre> is the most specific boundary comprising of the nulls. Considering the first example, the first example is a positive example. The hypothesis in the most generic boundary contains all question marks and all question marks match the values of all attributes of the first example and thusly the hypothesis in the most generic boundary stays unaltered and it must be looked at that the hypothesis in the most specific boundary is consistent with the example . The hypothesis in the most specific boundary contains all nulls and is not consistent with the considered example and the hypothesis maps wrongly the example to zero though the example is a positive one. Since the hypothesis in the most specific boundary is not consistent with the example, it should be generalized to S1: <Sunny, Warm,Normal,Strong,Same> . Considering the second example, the second example is also a positive example. At first the hypothesis in the most generic boundary must be consistent with the example and in light of the fact that it is consistent with the example, it stays unaltered G2=G1. Checking now the hypothesis at the most specific boundary, it can be said that S1 wrongly maps the positive example to 0 and thusly it does not consistent with the example as the value of the Humidity attribute at the S1 hypothesis does not match the value in the example and therefore the S1 hypothesis must be generalized, in order to match the example . The value of the humidity attribute at S1 must be replaced with the question mark. The third example is a negative example. At first, it should be ensured that S2 is consistent with the example and considering the third example it can be said that S2 is not consistent with the third example and in this manner it remains unchanged S3=S2. The hypothesis in the most generic boundary contains all question marks and accordingly maps wrongly the example to zero. Since the hypothesis is not consistent with the training example, all hypothesis that are consistent with all the training examples seen till now will be written. In another words, all the minimal spezializations of the hypothesis G2 should be written such that they are consistent with all the positive previous examples and not consistent with the negative one. For writing those particular hypothesis , one question mark at a time should be considered. The first question mark at G2 must be replaced with the opposite value of the first value at the negative example that is Sunny and all other question marks at the hypothesis remains unchanged so the outcome is <Rainy,?,?,?,?,?>. The next hypothesis should contain at its second value the opposite of the value at the second index in the negative example that is Warm and all other values are question marks <?,Warm,?,?,?,?> . The third hypothesis are constructed the same way through replacing the third question mark by the opposite of High that is Normal <?,?,Normal,?,?,?>. For the 4-th attribute at the negative example , there should not be changed , because there is no opposite of the value Strong in the tabe. The last two hypothesis are constructed the same way <?,?,?,?,Cool,?> and <?,?,?,?,?,Same>. After writing all the hypothesis , it should be made sure that these hypothesis are consistent with the previous examples. There are five hypothesis that have been written and now the question whether all of them are consistent or not. If they are consistent with all the seen examples, they can be kept and if they are not, then they should be removed. The first and second hypothesis are consistent with the first two examples and not with the third one , so they should not be deleted. The third hypothesis is not consistent with the second example, because the third value in the hypothesis(Normal) does not match the third value in the second example and therefore it should be removed. The fourth hypothesis is not consistent with the first example as the 4-th value does not match the value in the first example. Th lest hypothesis can be kept as it is consistent with all the first three examples.

G3 = <Sunny,?,?,?,?,?>   <?,warm,?,?,?,?>   <?,?,?,?,?,Same>

The last example is a positive example. All the hypothesis that are consistent with the example should be retained and if there is a hypothesis that is not consistent with the example , then it should be removed. The first and second hypothesis at the most generic boundary <Sunny,?,?,?,?,?> <?,Warm,?,?,?,?> are consistent with the examples as their values match the values of the example and therefore they should not be removed. The last hypothesis in the most generic boundary is not consistent with the example as the value at the last index(Same) does not match the value of the last attribute in the example and therefore it should be removed.

G4= <Sunny,?,?,?,?,?> <?,Warm,?,?,?,?>

The last example cannot be covered by the hypothesis at the most specific boundary S3 as the 4-th and the 5-th values (Warm,Same) in the hypothesis does not match the values of the 4-th and 5-th attributes (Cool, Change) in the example and therefore the last two values in the hypothesis should be replaced with the ? , so that the examples can be covered by the hypothesis.

S4 = <Sunny, Warm,?,Strong,?,?>

Features of the version space learning process:

Before the learning process gets started, there are some prerequisites that must be meet in order to convergent against the right hypothesis that defines the target concept.

The training set does not contain any errors:

it should be made sure that the given data set is correct and does not contain any errors. It can be said that the data contains any errors, if a positive example e is introduced to learn process as a negative one. In this case all hypothesis that cover the example e must be deleted as it is a negative example. Furthermore, with that the right target concept from the version space will be likewise eliminated, as this must cover the example e. Similarly, representing a negative example e to the learning process as a positive example can have an impact on finding the right hypothesis. Because the example e is represented wrongly as a positive example, all hypothesis that do not cover it should be deleted, as a result the right hypothesis that define the concept will be eliminated from the version space as well, as it does not cover the example e. If there are enough training examples, such an example will lead to the disclosure of the failure. When there are enough positive and negative training examples, a pattern can be derived from the set of positive examples and another one can be as well derived form the set of negative examples. A well observing of these patterns can be helpful by detecting the wrongly represented training example. A wrongly represented example can lead to disclosure of the error, since the version space at last will fall into the empty set.

The breakdown of the version space can likewise be because the target concept cannot be described in the representation language.

Concepts that cannot be described in the hypothesis space:

Sky	Temp	Humid	Wind	Water	Forecast	Enjoy Sport
Sunny	Warm	Normal	Strong	Warm	Same	Yes
cloudy	Warm	Normal	Strong	Warm	Same	Yes
Rainy	Warm	Normal	Strong	Warm	Change	No

Not always it is conceivable to infer a hypothesis from the hypothesis space that best fits the training data. It relies upon the training data and how they are structured. Concerning the above example , it can be claimed that there does not exist any hypothesis that is consistent(complete and correct in regards to the training set) with the training data and that can be expressed in the hypothesis space L_C of the training data. The most special hypothesis that is consistent with the first two examples is the following hypothesis h = <?,Warm,Normal,Strong,Warm,Same> and it is not consistent with the third one as it is not correct concerning it , on the grounds that it wrongly covers the examples, however, it is a negative example. The problem is that it is not possible to describe such disjunctions Sky= Sunny OR Cloudy in the concept language L_C. In other words, an attribute cannot be set to take specific values. The version space learning process is functioning using the inductive bias and concerning the inductive bias, only conjunctive hypothesis can be taken into consideration.

Concept learning with feature trees:

In the previous section and specifically in the example …,Even though the last example is a negative example, the final hypothesis covers it wrongly, because the inductive hypothesis (inductive bias) , with which the learning system is functioning ,allows only conjunctive hypothesis to be taken into consideration and therefore it is not possible to assign an attribute a specific number of values (Sky = sunny OR sky = cloudy).

Feature trees overcomes this defizit and can be used to structure attribute values hierarchically , so as not to generalize into the most general constraint , when two different values of an attribute coincide.