Machine Learning Lab Class II

Ankit Satpute, 120825
Hsueh Wei, 120820
Sagar Nagaraj Simha, 120797
(M.Sc. CS4DM)

25/11/2019

Exercise 2.a

Find-S algorithm for the example sequence 1,2,3,4:
First we start with maximally specific hypothesis

$$h_0 = < \varnothing,\varnothing,\varnothing,\varnothing >$$

Then for each pair in our training data we check whether it is positive example or not and if it is positive i.e. c(x)=1 then relax our hypothesis

Iteration 1: In this case our x_1 is $<$ Monday, no, easygoing, evening $>$ and c(x)=1. Hence, we generalize our hypothesis. Comparing both we observe that for our hypothesis to accept current example we need to replace all features in hypothesis with corresponding feature in our example, respectively.

$$h_1 = < \text{Monday, no, easygoing, evening} >$$

Iteration 2: For second example x_2 is $<$ Monday, no, annoyed, evening $>$ and c(x)=0. Hence, we don't do anything as this is negative example. Our hypothesis will remain same.

$$h_2 = < \text{Monday, no, easygoing, evening} >$$

Iteration 3: In this case our $x_3 = <$ Saturday, yes, easygoing, lunchtime $>$ and c(x)=0. Hence, we don't do anything as this is negative example. Our hypothesis will remain same.

$$h_3 = < \text{Monday, no, easygoing, evening} >$$

Iteration 4: For final x_4 is $<$ Monday, no, easygoing, morning $>$ and c(x)=1. Hence, we generalize our hypothesis. Comparing both we observe that for our hypothesis to accept current example we need to replace 4th feature in hypothesis with accept all '?' feature.

$$h_4 = < \text{Monday, no, easygoing, ?} >$$

After exhausting all our examples we come to a hypothesis that will accept our positive examples which is < Monday, no, easygoing, ? >

Exercise 2.b

Candidate elimination algorithm for the example sequence 1,2,3,4:
First we start with maximally specific hypothesis and maximally general hypothesis

$$S_0 = \{< \varnothing,\varnothing,\varnothing,\varnothing >\}$$
$$G_0 = \{< ?,?,?,? > \}$$

Then for each pair in our training data we check whether it is positive example or negative one and then proceed according to the steps given in algorithm on lecture slides.

Iteration 1: In this case our x_1 is < Monday, no, easygoing, evening > and c(x)=1. Hence, this is a positive example and as given on slides we check whether satisfies each hypothesis in G, which in this case we only have 1. Therefore g(x)=1, as our G contains maximally general hypothesis.
Now we check if for each s in S whether s(x) = 0 then we generalize s with respect to x. In our as well case s(x)=0. After that we check two conditions with S and S plus and return our S and G.

$$S_1 = < \text{Monday, no, easygoing, evening} >$$
$$G_1 = < ?, ?, ?, ? >$$

As one can see that we only generalized S in order to allow positive examples by S.

Iteration 2: For second example x_2 is < Monday, no, annoyed, evening > and c(x)=0. Hence, our example is negative and we follow the steps which are written algorithm
for each s in S, s(x)=0
for each g in G, g(x)=1, hence we remove g from G and specialize g with respect to s

$$G^- = \{< \text{Saturday}, ?, ?, ? >, < ?, \text{yes}, ?, ? >, < ?, ?, \text{easygoing}, ? >, < ?, ?, ?, \text{lunchtime} >, < ?, ?, ?, \text{morning} >,\}$$

Now we check condition whether to include elements in G^- in our G or not and return G. Also we keep elements in G which are more general than other

$$G = \{ < ?, ?, \text{easygoing}, ? >,\}$$

Finally our G and S after 2nd iterations are

$$S_2 = <\text{Monday, no, easygoing, evening}>$$
$$G_2 = <\text{?, ?, easygoing, ?}>$$

Iteration 3: In this case our $x_3 = <$ Saturday, yes, easygoing, lunchtime $>$ and c(x)=0.
s(x)=0
for each g in G, g(x)=1. Hence G={}
Now we again specialize g with respect to x and add results to G^-

$$G^- = \{<\text{Monday, ?, easygoing, ?}>, <\text{?, no, easygoing, ?}>, <\text{?, ?, }\varnothing\text{, ?}>, <\text{?, ?, easygoing, morning}>, <\text{?, ?, easygoing, evening}>\}$$

Again we check conditions fro including elements from G^- to main G and return our G. As we included only few elements in our G because those were more generalized than at least one existing element in S.

$$G = \{<\text{Monday, ?, easygoing, ?}>, <\text{?, no, easygoing, ?}>, <\text{?, ?, easygoing, evening}>\}$$

Hence, our final S and G after 3rd iteration are

$$S_3 = <\text{Monday, no, easygoing, evening}>$$
$$G_3 = \{<\text{Monday, ?, easygoing, ?}>, <\text{?, no, easygoing, ?}>, <\text{?, ?, easygoing, evening}>\}$$

Iteration 4: For final iteration, x_4 is $<$ Monday, no, easygoing, morning $>$ and c(x)=1. Again we go through the desired loop and perform the actions
c(x)=1
for each g in g(x)
g(x_1)=1
g(x_2)=1
g(x_3)=0, hence we will remove this from our G.
for each s in S, s(x)=0
As we can see, we need to generalize our s in order to accept this positive example, so we just check with every feature and relax those who are making contradiction with the feature in x

$$S^+ = \{<\text{Monday, no, easygoing, ?}>\}$$

After exhausting all our examples we come to a hypotheses that are consistent with all the examples in training set which are as follows

$$S = \{<\text{Monday, no, easygoing, ?}>\}$$
$$G = \{<\text{Monday, ?, easygoing, ?}>, <\text{?, no, easygoing, ?}>\}$$

Exercise 2.c

In this case the boundaries of the version space S and G according to 2.b are the only members of the version space

$$V_{H,D} = \{< \text{Monday, no, easygoing, ?} >, < \text{Monday, ?, easygoing, ?} >, < \text{?, no, easygoing, ?} >\}$$

Yes, version space can contain hypotheses that are neither in S nor in G. S and G are boundaries and can enclose multiple hypotheses that are consistent with training examples. Through the training examples we infer the boundaries for S and G which are most specific and most general respectively, as compared to the once we start, which are maximally specific and maximally general. **The hypotheses in between S and G are, more specific than G and more generic than S as according to "General to specific ordering of hypotheses"** Hence, it is possible that there are hypothesis which are still in between S and G which are consistent with our training examples. We have encountered such cases in examples covered in slides and reference book where one can see the hypotheses in between S and G that are consistent with encountered examples.[1]

References

1 Tom Mitchell. Machine Learning. 1st edition, McGraw-Hill, 1997.

2 https://github.com/profthyagu/Python–Candidate-Elimination-Algorithm