

Artificial Intelligence

CSE413

Project: Version Space Search

This project is to implement the Version Space Search inductive learning algorithm as supervised learning technique in Prolog. Given a labeled dataset of binary classes your program will determine the concept hypothesis present in the dataset. It will then classify a test sample without label as belonging to either positive or negative class.

You must write a menu driven program to train your model by showing positive class instances and negative class instances. You can represent the instances and concepts as a list of features like for a small, red, ball use [small, red, ball]. Your program can check the concept built in the model so far. Also, your program can test an unlabeled sample to be positive or of negative class.

You can use the following dataset to test your program.

Train/Test	Attribute-Size	Attribute-Color	Attribute-Shape	Class
Train	small	red	Ball	positive
Train	small	white	Ball	positive
Train	large	blue	Ball	positive
Train	small	red	Brick	negative
Train	large	blue	Cube	negative
Test	medium	blue	Ball	?
Test	large	red	Cube	?

Your menu driven program should interact with the user as follows:

This program will build a concept space model from training examples and then classify an unseen sample.

Enter a selection followed by a period.

- 1. Enter a positive instance**
- 2. Enter a negative instance**
- 3. Show the concept hypothesis so far**
- 4. Enter a test sample**
- 5. Exit**

For this project you have to submit your prolog program as filename.ipynb and a PDF file showing screen shots of execution results of all menu items.

Hint: You can implement the following predicates:

With the meaning of the variables as SH= specific hypothesis. GH=General hypothesis, US=updated specific, UG=updated generic, P= Positive samples, N= negative samples, L1 =LIST 1, L2=list2, you can implement the following predicates and others as needed.

vss(SH, GH, P, N)
covers(L1, L2)
more_general(L1, L2)
generalize(SH, P, US)
generic_to_specific(SH, US, N, GH, UG)
specific_to_generic(SH, US, GH, P, _)
generalized_set(SH, N, US)