CISC859 Pattern Recognition, Winter 2019

Assignment 4, due February 27

Clustering course reader pp 49-55

1) Clustering using a criterion function: what are strengths and weaknesses of the sum-of-squared-error criterion function?

[The answer to this question is stated directly in the course reader and textbook. I want to draw your attention to this because clustering is widely used. It is indeed easy to choose a clustering method that “imposes structure on the data” rather than “discovering structure inherent in the data”, and because this happens in high-dimensional space it can be difficult for the researcher to become aware of the problem. One strategy is to apply several different clustering methods and compare results. If the different methods produce similar results then you can be pretty confident that the clustering algorithms are describing structure inherent in the data. If the different methods produce dissimilar clusterings then try to figure out why; doing this will probably give you insight about the structure of the data.

Problems of dimensionality course reader pp 56-60

2a) Section 3.7.1 of DHS says after equation (73) “The most useful features are the ones for which the difference between the means is large relative to the standard deviations”. Briefly explain this sentence, providing informal justification.

b) The next sentence is “However, no feature is useless if its means for the two classes differ.” Consider the converse: if a feature has the *same* mean for two classes, then does this imply that the feature is useless (i.e. the feature provides no information that is helpful for classification)? Provide an example to illustrate your answer.

c) The last paragraph of section 3.7.1 says: “Unfortunately, it has frequently been observed in practice that, beyond a certain point, the inclusion of additional features leads to worse rather than better classifier performance.” Briefly describe a reason why this might occur.

Read about the following topics in the course reader. No assignment question on these topics.

Defect model course reader pp 61-62, used in c/e example pp 33-34

Introduction to structural pattern recognition course reader pp 63-65

Graph matching in structural pattern recognition course reader pp 66-67

Syntactic pattern recognition; review of string grammars course reader pp 68-74

3) Consider this context-free grammar:

<start> → <letter> <start>
<start> → <letter>
<letter> → <vowel> | <cons> ; The “|” symbol stands for “or”
<vowel> → a | e | i | o | u | y
<cons> → b | c | d | f | g | h | j | k | l | m | n | p | q | r | s | t | v | w | x | y | z

a) Show a parse tree for the string “eat”.

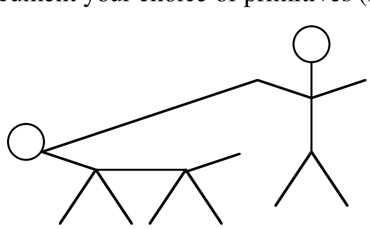
b) This grammar is ambiguous. Find an example of a string that can be parsed in two ways.

c) Modify this grammar so that a letter “q” must be followed by a letter “u”.

d) Modify this grammar so that it can parse a string if and only if that string contains at least one vowel. Your grammar should not add restrictions such as “must start with a vowel” or “must end with a vowel” — as long as the stering contains a vowel somewhere, it should be accepted by the grammar.

Picture Description Language (PDL) course reader pp 75-77

4) Write a PDL expression to describe the following figure. Use primitives consisting of circles and straight line segments. Document your choice of primitives (including the “head” and “tail” location of each primitive).



The line segment for the dog-tail is the same as the line segment for the person-arm at right, and the dog-neck is the same as the person-arm at left. The “leash” consists of four person-arm line segments concatenated together. Clearly show how you define one or two circle primitives to get the correct attachment points for the person-head and dog-head.

Read about:
Transformational grammars course reader pp 78-80
Attribtued grammars course reader p 81

Anderson Math Grammar course reader pp 82-106
5) Show a parse tree for Anderson's coordinate grammar applied to the input expression on the right. Anderson’s grammar is discussed in the course reader pages 82-90, with a complete list of the grammar rules on pages 91-98. Show your answer as a parse tree, with the root of the tree labeled EXPRESSION. At each node in the tree, write down the number of the production that you use to expand the parent node into child nodes.

$$5\,x^2 \sum_{n=1}^{50} n\sqrt{n}$$

Anderson's parser uses a lot of search and backtracking to build the parse tree in a top-down manner. You do not need to go through these backtracking steps; just use whatever method allows you to most quickly identify the correct set of production rules to use. For example, you can see that the input has implied multiplication: there is no explicit symbol “*” to indicate multiplication. So you need to use Anderson’s rules for implied multiplication; avoid rules such as A7 or A9 that only deal with explicit multiplication.

You can draw the tree by hand or use formatting software, whatever you find easiest. As is done on page 88 of the course reader, you can use a short-hand notation for the following derivation sequence:

EXPRESSION (A5) TERM (A10) ADJTERM (A22) DIVTERM (A25) FACTOR (A32) VARIABLE (A55) ALPHA (A56) LETTER

Once you have drawn the whole parse tree, then assign m values by starting at the leaves and working toward the root. **Write down the “m” attribute value at each place in the tree where it changes**; the value of “m” at the root of the tree is the output produced by this computation.

Question about the scope of summation: Does the Anderson grammar interpret this input as a summation over the expression "(n * root n)" or does it interpret this expression as a summation over "n", with the result multiplied by root n?

Read about:
Graph grammars and tree grammars course reader pp 107-113

Stochastic and fuzzy grammars course reader pp 114-116
6) Here is a stochastic grammar.

<S> ::= <HELLO> <ASK> <ENDING> Prob. = 1.0

<HELLO> ::= Hi Prob. = 0.3
<HELLO> ::= Hello Prob. = 0.6
<HELLO> ::= Howdy Prob. = 0.1

<ASK> ::= How are you? Prob. = 0.7
<ASK> ::= What’s up? Prob. = 0.1
<ASK> ::= What’s happenin’? Prob. = 0.1
<ASK> ::= <ASK> <ASK> Prob. = 0.1

<ENDING> ::= Eh? Prob. = 0.3
<ENDING> ::= --silence--- Prob. = 0.7

Give two different sentences that belong to the language generated by this grammar. Show the sentence and also the probability that the sentence belongs to the language.

Read about:
Summary course reader pp 117-119