

## Machine Learning, spring 2005.

Extended hints and tentative solutions to exercises.

Send flames and corrections to [jtindgr@cs.helsinki.fi](mailto:jtindgr@cs.helsinki.fi).

On many cases, I have given only a general outline of the solution. It should be enough to let you fill in the details. **For a student, these kinds of answers may not be acceptable in a test** – you should give as detailed account as you can.

On some exercises, I have included a paragraph tagged **rant**. This is a piece of assistant's subjective speculation and handwaving that usually tries to relate the exercise to the universe as the empirist sees it. If you bother to read them, do so with a large grain of salt. They can be skipped safely.

### Exercise 1.1

Mostly, use what you know from elementary logic.

b) The formula can be multiplied from left to right, resulting in DNF

$$f(x) = x_1x_2 + x_1x_3x_4 + x_1x_3x_5x_6.$$

c) Can be expanded e.g. deepest first, using the distributive law ( $x + yz = (x + y)(x + z)$ ), towards CNF

$$f(x) = x_1(x_2 + x_3(x_4 + x_5)(x_4 + x_6)) \quad (1)$$

$$= x_1(x_2 + x_3)(x_2 + x_4 + x_5)(x_2 + x_4 + x_6) \quad (2)$$

a) Can be reasoned from the CNF, giving

$$((\overline{x_1}, -), (x_2, +), (\overline{x_3}, -), (x_4, +), (\overline{x_5}, -), (x_6, +), -)$$

**rant:** Algorithms exist for conversions like these. Less formal attempts with small amount of literals can be verified for correctness using a truth table (basically brute force checking of all variable/truth\_assignment possibilities). The moral of the story could be that "the truth" might exist in different forms and they can occasionally represent the same thing, although it might not be apparent at all. Around the eighties, it was a great pastime to prove the relations between different hypotheses (or -classes). Current status: on real data, algorithms working explicitly on logic tend to produce less accurate classifiers than their more graded competitors. Often a linear classifier or a nearest neighbour does reasonably. With DNF-style rule learners, your mileage might vary. Attempts to enhance (e.g. logic-based) algorithms by heuristical kludging easily hit a Murphy-variant of the *the law of diminishing returns* (gains grow logarithmically w.r.t. the amount of workhours).

### Exercise 1.2

Informally, start by drawing a 2D illustration of the situation and generalize to larger dimensions. The idea is to see that we are in effect separating one corner of a hypercube, we don't have to care about the rest of the examples as they are not close to the margin.

a) As the situation is symmetrical, set  $\overline{w_{1:k}} = 1$  and  $\overline{w_{k+1:n}} = 0$ . By flipping one bit of the positive example that has just bits  $1 : k$  true, reason that the threshold  $b = k - 1$ . Hence, by simple arithmetic the margin will turn out to be  $1/\sqrt{k}$ .

b) Divide the  $(\overline{w}, b)$  of part a) by  $1/\sqrt{k}$ . You'll see that the unnormalized margin ends up the same as the normalized margin was in part a).

c) The same as in a).

d) Using the method from the lectures, add coefficient  $\overline{w_{n+1}} = -(k - 1) = 1 - k$  and set  $b = 0$ . As the bias is now a part of the vector, its calculated along in the margin, resulting in  $1/\sqrt{k + (1 - k)^2} \leq 1/\sqrt{k}$ .

note: you could show, using linear inequalities, that any linear classifier having a larger margin than those given leads to a contradiction. Or, you might apply geometrical ideas of convexity with optimization theory. Not required here.

**rant:** Why we are so interested in the margin might get clearer further down the course. For example, the perceptron convergence theorem (Novikoff) gives a mistake-bound on the perceptron algorithm that is inversely related to the square of the margin. Also, according to Structural Risk Minimization theory (SRM), *getting the classes well-separated with a simple model is often a good thing*. But. Real-life data might be continuous and the labeling might be *just an opinion of some person*. There is not necessarily a margin in the input space, if the data comes from smooth, partially overlapping distributions (imagine e.g. a "candy space". Co-ordinate axis represent percentages of candy ingredients, i.e. sugar. Sample candies. Taste. Label each as good or bad. Is there a margin?). We might be more interested about keeping the most of the data away from the decision surface than attempting to get a "hard margin" by some complexity (or norm) increasing tricks.

### Exercise 1.3

i) Note that a parity function is TRUE iff an odd number of literals are TRUE. Hence, create a binary decision tree with three levels that essentially counts the number of TRUE literals in each root-to-leaf -path. The tree is not unique, as you can do sums in any order.

ii) Write out parity of  $x_i, x_j$  as  $x_i\overline{x_j} + \overline{x_i}x_j$ . Expand the formula by basic logic. Up to ordering, should give something like

$$x_1\overline{x_2}\overline{x_3} + \overline{x_1}x_2\overline{x_3} + \overline{x_1}\overline{x_2}x_3 + x_2x_1x_3.$$

b) Hard way: we have four examples,  $(-1,-1,\text{FALSE}), (-1,1,\text{TRUE}), (1,-1,\text{TRUE}), (1,1,\text{FALSE})$ . Set up a system of linear inequalities to say that these should be classified correctly by some  $(\overline{w}, b)$ , i.e.  $\overline{w} \cdot \overline{x} \geq b$  for positive examples etc. Trying to solve this linear system should result in a contradiction. Less formally,

looking at a 2D illustration of the case and trying to put a hyperplane there should demonstrate the case rather convincingly to everyone except the most unreformed theorists. A third possibility is to note the monotonicity of linear models: each weight can only increase or decrease the relation of the example to a class. In the XOR problem, whether value of  $x_1$  should have a positive or negative effect depends on the value of  $x_2$ .

#### Exercise 1.4

M. Anthony 2002, "Decision Lists and Threshold Decision Lists", CDAM Research Report LSE-CDAM-2002-11, has an induction proof of the required mapping at pages 15-16 (theorem 5.1) <sup>1</sup>.

The idea shortly is to start from the first non-redundant (positive-label) node of the decision list tail and make a linear classifier for that single node. Then, moving from the declist tail to front, start to add weights to the linear classifier (while editing the bias) so that the new weight is always large enough to override all the weights to the right (those nearer the tail) as they are less important. The hinted sequence  $\overline{w}_i = 2 * \overline{w}_{i+1}$  should also manage this. This should give a linear classifier where the weights grow exponentially w.r.t. the decision list length, so the margin will get small really fast.

**rant** From this you can see e.g. that with a linear classifier, you could simulate the fragility of a 1-decision list: try adding a little errors to the data to the right place.

#### Exercise 1.5

Note that the data sample conforms to some multiliteral, monotone conjunction  $h$  and it has no errors. The sample does not necessarily suffice to specify the conjunction exactly, but here we are interested just in conforming to just the sample. Put the positive examples to a matrix, each row an example. Apply logical AND on the matrix columnwise. The result is the hypothesis. It predicts TRUE on all positive examples, as it necessarily has atleast the bits on that  $h$  has (because all positive examples must have them on). It predicts FALSE on all negative examples, because they cannot have those all the bits in  $h$  on (that our hypothesis atleast requires). If we have  $m$  examples and dimension  $n$ , the complexity is clearly  $O(mn)$ , which is polynomial in  $n$ .

**rant:** It should be remembered that any algorithm that makes irrevocable choices based on just one example is almost always unsuitable for practical use. This is due to errors/uncertainty present in natural data. Also, often the hypothesis class used by the method can only approximate the underlying labeling phenomena, not equal it (resulting in errors).

---

<sup>1</sup><http://www.maths.lse.ac.uk/Personal/martin/cdambfdls.pdf>