Machine Learning 2
Exercise Sheet 7

Prof. Dr. Dr. Lars Schmidt-Thieme, Brad Baker
Information Systems and Machine Learning Lab
University of Hildesheim

June 9th, 2017

Submission until June 20th, 12:00 PM by learnweb.
Please put your name in all filenames.
Non-pdf submissions for non-programming exercises will not be graded.

Exercise 12: Boosting (10 points)

Consider the following dataset consisting of five training examples followed by three test examples:

x_1	x_2	x_3	y
	training		
-	+	+	-
+	+	+	+
-	+	-	+
+	+	-	+
	test		
+	-	-	?
-	-	-	?
+	-	+	?

Perform three rounds of *AdaBoost* learning on this data in order to predict the test labels. Treat the labels $+$ and $-$ as real numbers and use Decision-Stumps (one-level decision trees) as the underlying "weak" predictive model, assuming that each stump minimizes error as much as possible on the training set.

After running the three iterations, provide your final predictions and comment on the boosted model as compared to one of the decision trees.

Exercise 13: Mixture of Experts

Product of Experts work similar to mixtures of experts. They split the data into n subsets as

$$X = (X^{(1)}, ..., X^{(n)}), \;\; y = (y^{(1)}, ..., y^{(n)}),$$

and then model the joint likelihood of y given univariate data X and model parameters θ as a product of individual experts:

$$p(y|X, \theta) = \prod_{i=1}^{n} p_i(y^{(i)}|X^{(i)}, \theta^{(i)}).$$

Show that for p_i being the predictive density of a Gaussian process, the product of two GP experts for a new instance x_* is **proportional** to a gaussian distribution with mean μ_* and variance ρ_*^2.

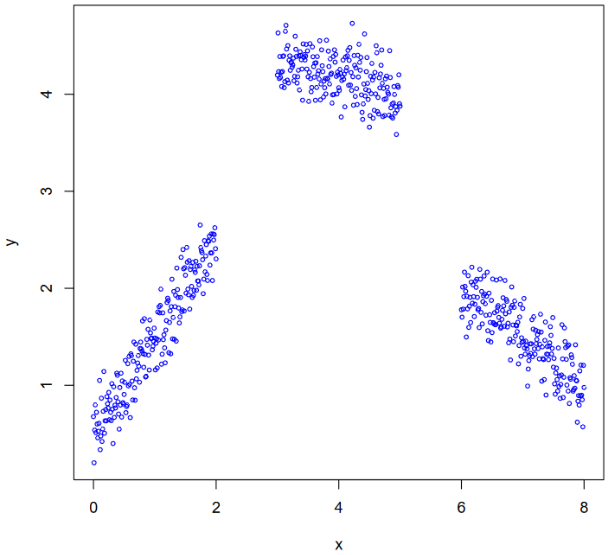
$$\mu_* = \rho_*^2(\rho_1^{-2}\mu_1 + \rho_2^{-2}\mu_2)$$

$$\rho_*^2 = (\rho_1^{-2} + \rho_2^{-2}))^{-1}$$

Note: μ_i and ρ_i are dependent on $x*$.

Bonus (10 points)

n R, create a univariate data set that looks approximately like this:



- Think of three linear functions γ_i^T for $i = 1, 2, 3$ such that their softmax is a gating function for the respective data clouds
- Learn three (unregularized) linear regressions (one per cloud) and use the gating function from the previous problem to build a mixture of experts. Use a mixture of experts on all data points and plot the result.