10-701 Machine Learning (spring 2012)
Solution for homework 2

1. Logistic regression

1.1 Logistic vs linear regression

1.1.1 The simplest option is to set a threshold T s.t. class affiliation is determined by comparing wx with T, e.g. $\mathrm{wx} < \mathrm{T} \rightarrow \mathrm{class1}$. The drawbacks are mainly two 1) susceptibility to outliers 2) logit squashes things to between 0 and 1 and hence at least offers a way to represent the degree of affiliation based on the class conditionals.

1.1.2 The logit models the log odds of class conditionals as a linear combination of the input variable: $\mathrm{wx} = \log\left(\frac{\mathrm{p(Y=1|X)}}{1-\mathrm{p(Y=1|X)}}\right)$. Hence a unit change in x induces an exponential $\exp(\mathrm{w}) = \mathrm{e}^2$ change in the odds.

1.2 Logistic vs Naïve Bayes

See Mitchell chapter (http://www.cs.cmu.edu/~tom/mlbook/NBayesLogReg.pdf) for derivation—highly recommends reading the entire chapter if haven't done so.

1.3 Loss function

The loss function is the negative (log) likelihood

$$\mathrm{L} = \ln \prod_{\mathrm{i}} \left(1 - \sigma(\mathrm{wx_i})\right)^{\mathrm{y_i}} \sigma(\mathrm{wx_i})^{1-\mathrm{y_i}} = \sum_{\mathrm{i}} \mathrm{wx_i y_i} - \ln\left(\mathrm{e}^{\mathrm{wx_i}} + 1\right)$$

Where the derivative is $\frac{\partial \mathrm{L}}{\partial \mathrm{w}} = \sum_{\mathrm{i}} \mathrm{x_i}(\mathrm{y_i} - \sigma(\mathrm{wx_i}))$.

2. Learning theory

2.1 PAC learning theory

2.1.1 2^N, since each ingredient takes up 2 possible states: used or not used. Answers that excludes the empty salad case would also be considered correct.

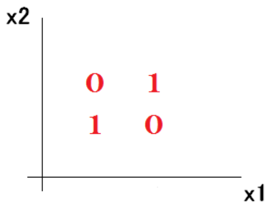
2.1.2 3^N, since slicing adds an additional state to the ingredients. Answers that consider each of two slices as distinct state would also be considered correct.

2.1.3 Using the equation in PAC learning we could work out the bound on the samples:

$$m \geq \frac{1}{\epsilon}\left(\ln|H| + \ln\left(\frac{1}{\delta}\right)\right) = \frac{1}{0.01}\left(\ln|3^6| + \ln\left(\frac{1}{0.01}\right)\right) \cong 1120$$

2.2 VC dimensions

Here is a scenario where such a circle cannot shatter 4 points in 2D.



3. Mistake bounds

3.1 $w_b = (\frac{1}{2})^{M_b}$ since for every mistake the best agent makes its weight is halved.

3.2 $W_{max} \leq N\left(\frac{3}{4}\right)^{M_e}$ since the initial ensemble weight is N and that for each mistake made by the ensemble as a whole, the least number of agents get wrong would be a half of the population. Those agents would get halved in the weights while the weights for the "correct" agents would be intact, amounting to a fraction of three quarters of the weight for the ensemble.

3.3 Here is a trivial bound $w_b \leq W_{max}$.

3.4 Equating answers in 1 and 2 we get the following

$$(\frac{1}{2})^{M_b} \leq N\left(\frac{3}{4}\right)^{M_e} \rightarrow M_e \leq 2.4(M_b + log_2 N)$$

4. Guess the lean animal

As covered in the recitation (see recitation slides on decision trees and learning theory), this question can be tackled with DT too—perhaps in a way more elegant than logit since DT returns a clear and more succinct definition for the two classes, as opposed to weights on all 6 features.

The data set includes 25 samples in each class.

A canonical result without any regularization on the weights should yield around 88% on the lean class and 68% on the chubby class in the LOOCV test. The weight (W) estimated by fitting on all the data is compared with the correlation (C) of each feature with respect to the class labels as follows:

	YouGuess=Smart	Hunter	Agility	Slow	Toughskin	Spots
W	$.91$	1.12	$.72$	-1.82	$-.87$	$.01$
C	$.43$	$.46$	$.55$	$-.65$	$-.36$	$.00$

Logit code and animal ranks provided by *Soonho Kong*.

```
function [L, C, W] = generic_LR(X, y)
   X = addOne(X);     W = logistic_grad_ascent(X, y);
   C = sigmoid(X * W);     L = C >= 0.5;
end


function theta = logistic_grad_ascent(X, y)
   alpha = 0.0001;     delta = 0.0001;     theta = zeros(size(X, 2), 1);
   done = true;     ll = -Inf;
   while done
      hx = sigmoid(X * theta);
      theta = theta + alpha * X' * (y - hx);
      ll_new = sum( y .* log(hx) + (1 - y) .* log(1 - hx) );
      done = ll_new - ll > delta;
      ll = ll_new;
   end
end


function ret = sigmoid(z)     ret = 1.0 ./ (1.0 + exp(-z)); end
function ret = addOne(X)     [m,n] = size(X);     ret = [ones(m, 1) X];  end
```

Animal Name	(P(Y = 1 | X))
'leopard'	[0.9137]
'bobcat'	[0.9137]
'german shepherd'	[0.9129]
'siamese cat'	[0.9129]
'tiger'	[0.9129]
'fox'	[0.9129]
'wolf'	[0.9129]
'rat'	[0.9129]
'weasel'	[0.9129]
'otter'	[0.9129]
'lion'	[0.9129]
'collie'	[0.9129]
'killer whale'	[0.8154]
'bat'	[0.8138]
'dalmatian'	[0.7752]
'deer'	[0.7752]
'raccoon'	[0.7752]
'spider monkey'	[0.7734]
'chihuahua'	[0.6233]
'beaver'	[0.5874]
'horse'	[0.5874]
'gorilla'	[0.5874]
'chimpanzee'	[0.5874]
'zebra'	[0.5874]
'dolphin'	[0.5874]
'hamster'	[0.5776]
'squirrel'	[0.5776]
'rabbit'	[0.5776]
'mouse'	[0.5776]
'seal'	[0.4156]
'antelope'	[0.3632]
'persian cat'	[0.3546]
'grizzly bear'	[0.2544]
'polar bear'	[0.2199]
'giant panda'	[0.2121]
'mole'	[0.1804]
'blue whale'	[0.1009]
'pig'	[0.1009]
'humpback whale'	[0.1000]
'elephant'	[0.1000]
'walrus'	[0.1000]
'giraffe'	[0.0973]
'skunk'	[0.0964]
'sheep'	[0.0964]
'cow'	[0.0430]
'hippopotamus'	[0.0426]
'moose'	[0.0426]
'ox'	[0.0426]
'rhinoceros'	[0.0426]
'buffalo'	[0.0426]