Adaboost

a) Let $E = \{i : h_t(x^{(i)}) \neq t^{(i)}\}$ *and* $E^c = \{i : h_t(x^{(i)}) = t^{(i)}\}$. First notice that $err_t = \sum_{i \in E} w_i / N$ and consequently $1 - err_t = \sum_{i \in E^c} w_i / N$. We have that $err_t' = (\sum_{i \in E} w_i'(1) + \sum_{i \in E^C} w_i'(0)) / \sum_{i=1}^{N} w_i'$, which by the definition of w_i' is equivalent to $\sum_{i \in E} w_i exp(-\alpha_t(-1)) / \sum_{i=1}^{N} w_i'$ because $t^{(i)} h_t(x^{(i)}) = -1$ when they are unequal. Now we split the denominator to get $err_t' = \sum_{i \in E} w_i exp(\alpha_t) / (\sum_{i \in E} w_i' + \sum_{i \in E^C} w_i') = \sum_{i \in E} w_i exp(\alpha_t) / (\sum_{i \in E} w_i exp(\alpha_t) + \sum_{i \in E^C} w_i')$ by the same steps outlined above. Then, $err_t' = \sum_{i \in E} w_i exp(\alpha_t) / (\sum_{i \in E} w_i exp(\alpha_t) + \sum_{i \in E^C} w_i exp(-\alpha_t)(1))$ by the definition of w_i' and because $t^{(i)} h_t(x^{(i)}) = 1$ when they are equal. Thus, by the definition of α_t, $err_t' = \sum_{i \in E} w_i exp(\alpha_t) / (\sum_{i \in E} w_i exp(\alpha_t) + \sum_{i \in E^C} w_i (\frac{1 - err_t}{err_t})^{-0.5}) =$

$\sum_{i \in E} w_i (\frac{\sum_{i \in E^C} w_i}{\sum_{i \in E} w_i})^{0.5} / (\sum_{i \in E} w_i (\frac{\sum_{i \in E^C} w_i}{\sum_{i \in E} w_i})^{0.5} + \sum_{i \in E^C} w_i (\frac{\sum_{i \in E^C} w_i}{\sum_{i \in E} w_i})^{-0.5}) =$ by definition of err_t

$\sum_{i \in E} w_i / (\sum_{i \in E} w_i + \sum_{i \in E^C} w_i (\frac{\sum_{i \in E} w_i}{\sum_{i \in E^C} w_i})^{0.5} / (\frac{\sum_{i \in E^C} w_i}{\sum_{i \in E} w_i})^{0.5}) = \sum_{i \in E} w_i / (\sum_{i \in E} w_i + \sum_{i \in E^C} w_i (\frac{(\sum_{i \in E} w_i)^2}{(\sum_{i \in E^C} w_i)^2})^{0.5}) =$

$\sum_{i \in E} w_i / (\sum_{i \in E} w_i + \sum_{i \in E^C} w_i \frac{\sum_{i \in E} w_i}{\sum_{i \in E^C} w_i}) = \sum_{i \in E} w_i / (\sum_{i \in E} w_i + \sum_{i \in E} w_i) = 1/2$

The interpretation is that Adaboost will adjust the weights such that the previous learner is no better than chance to maximize the power of the next learner such that it addresses the shortcomings of the previous one to as great an extent as possible.

b) We have that $w_i exp(2\alpha_t I(h_t(x^{(i)}) \neq t^{(i)}) = w_i exp(2\alpha_t 0.5(1 - h(x^{(i)}) t^{(n)}))$ The ratio is thus
$w_i exp(-\alpha_t t^{(i)} h_t(x^{(i)})) / w_i exp(\alpha_t(1 - h(x^{(i)}) t^{(n)})) =$
$exp(-\alpha_t t^{(i)} h_t(x^{(i)})) / exp(\alpha_t(1 - h(x^{(i)}) t^{(n)})) = exp(-\alpha_t t^{(i)} h_t(x^{(i)}) - \alpha_t(1 - h(x^{(i)}) t^{(n)})) =$
$exp(-\alpha_t t^{(i)} h_t(x^{(i)}) - \alpha_t t^{(n)} + \alpha_t h(x^{(i)}) t^{(n)}) = exp(-\alpha_t)$

Fitting a Naive Bayes Model

a) We have that the likelihood of obtaining a specific dataset of size n is $L(\theta, \pi) = \prod_{i=1}^{n} p(\vec{x}^{(i)}, c^{(i)} \mid \theta, \pi)$. Then the log-likelihood function is
$l(\theta) = \sum_{i=1}^{n} log(p(c^{(i)} \mid \pi) \prod_{j=1}^{784} p(x_j \mid c^{(i)}, \theta_{jc})) = \sum_{i=1}^{n} log(p(c^{(i)} \mid \pi) + \sum_{j=1}^{784} \sum_{i=1}^{n} log(p(x_j^{(i)} \mid c^{(i)}, \theta_{jc}))$.
Because each of these two terms depend on different parameters, we can optimize them separately. First, we fix j and consider maximizing $l_1 = \sum_{i=1}^{n} log(p(x_j^{(i)} \mid c^{(i)}, \theta_{jc}))$. This is

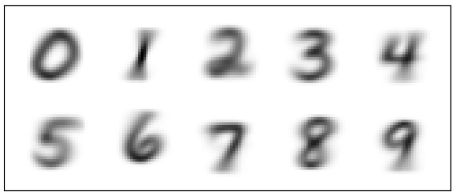
equivalent to $\sum_{i=1}^{n}\sum_{c=0}^{9} t_c^{(i)}(x_j^{(i)} log(\theta_{jc}) + (1-x_j^{(i)}) log(1-\theta_{jc}))$ by (0.1) and the given definition of θ_{jc}. We have that for a particular θ_{jc}, $\frac{\partial l_1}{\partial \theta_{jc}} = \sum_{i=1}^{n} t_c^{(i)}(\frac{x_j^{(i)}}{\theta_{jc}} - \frac{1-x_j^{(i)}}{1-\theta_{jc}})$ because the other terms in the summation don't contain the particular θ_{jc} so they go to 0 when the derivative is taken. Setting this derivative to 0, we obtain $0 = \sum_{i=1}^{n} t_c^{(i)}(\frac{x_j^{(i)}}{\theta_{jc}} - \frac{1-x_j^{(i)}}{1-\theta_{jc}}) = \frac{\sum_{i=1}^{n} t_c^{(i)} x_j^{(i)}}{\theta_{jc}} - \frac{\sum_{i=1}^{n} t_c^{(i)}(1-x_j^{(i)})}{1-\theta_{jc}}$. Solving for θ_{jc} yields our MLE estimator $\hat{\theta}_{jc} = \sum_{i=1}^{n} x_j^{(i)} t_c^{(i)} / \sum_{i=1}^{n} t_c^{(i)}$. $\hat{\theta}_{MLE}$ is then just the matrix such that the (j,c) entry is given by $\hat{\theta}_{jc}$. Now we consider maximizing $l_2 = \sum_{i=1}^{n} log(p(c^{(i)} \mid \pi))$. This is equivalent to $\sum_{i=1}^{n} log(\prod_{c=0}^{9} \pi_j^{t_j^{(i)}}) = \sum_{i=1}^{n}\sum_{c=0}^{9} t_c^{(i)} log(\pi_c)$ by the definition of π_c *and* $t_c^{(i)}$. For a particular π_c, $\frac{\partial l_2}{\partial \pi_c} = \sum_{i=1}^{n} \frac{t_c^{(i)}}{\pi_c}$. Setting this derivative to 0, we obtain $0 = \sum_{i=1}^{n} \frac{t_c^{(i)}}{\pi_c} = \frac{\sum_{i=1}^{n} t_c^{(i)}}{\pi_c}$, in which we cannot solve for π_c. This is because without the bound $\sum_{c=0}^{9} \pi_c = 1$, we could maximize l_2 by choosing π_c to be arbitrarily large. With this constraint, we pick $\pi_c = \sum_{i=1}^{n} t_c^{(i)} / n$.

b) We have by Bayes' Rule, $p(t \mid x, \theta, \pi) = \frac{p(t \mid \theta,\pi) p(x \mid t,\theta,\pi)}{\sum_c p(c \mid \theta,\pi) p(x \mid c,\theta,\pi)} = \frac{\pi_t \prod_{j=1}^{784} p(x_j \mid t,\theta_{jt})}{\sum_c \pi_c p(x_j \mid c,\theta_{jc})} = \frac{\pi_t \prod_{j=1}^{784} \theta_{jt}^{x_j}(1-\theta_{jt})^{(1-x_j)}}{\sum_c \pi_c \prod_{j=1}^{784} \theta_{jc}^{x_j}(1-\theta_{jc})^{(1-x_j)}}$.

Then, $log(p(t \mid x, \theta, \pi)) = log(\pi_t) + \sum_{j=1}^{784} x_j log(\theta_{jc}) + \sum_{j=1}^{784} (1-x_j) log(1-\theta_{jc}) -$

$log(\sum_c exp(log(\pi_c) + \sum_{j=1}^{784} x_j log(\theta_{jc}) + \sum_{j=1}^{784} (1-x_j) log(1-\theta_{jc})))$

c) It's possible that we have log(0) here, which the program cannot compute. A Runtime Warning was raised indicating that the program tried to divide by 0.



d)

e) We have that $p(\theta_{jc}, D) = p(\theta_{jc})p(D \mid \theta_{jc})$, where D is our dataset. Taking the log-likelihood gives $2log(\theta_{jc}) + 2log(1-\theta_{jc}) + \sum_{i=1}^{n} t_c^{(i)}(x_j^{(i)} log(\theta_{jc}) + (1-x_j^{(i)})log(1-\theta_{jc})) + c$ for some constant c from properties of the Beta distribution and l_1 that was derived in (a) after fixing c. Distributing the summation yields:

$$2log(\theta_{jc}) + 2log(1-\theta_{jc}) + \sum_{i=1}^{n} t_c^{(i)} x_j^{(i)} log(\theta_{jc}) + \sum_{i=1}^{n} t_c^{(i)}(1-x_j^{(i)})log(1-\theta_{jc}) + c =$$

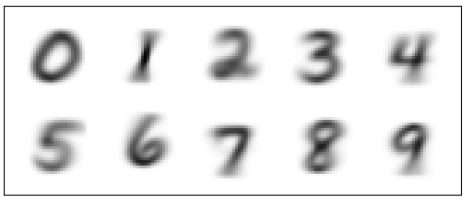
$$(2 + \sum_{i=1}^{n} t_c^{(i)} x_j^{(i)})log(\theta_{jc}) + (2 + \sum_{i=1}^{n} t_c^{(i)}(1-x_j^{(i)}))log(1-\theta_{jc})) + c$$

Taking the derivative with respect to θ_{jc} and setting to 0:

$$(2 + \sum_{i=1}^{n} t_c^{(i)} x_j^{(i)})/\theta_{jc} - (2 + \sum_{i=1}^{n} t_c^{(i)}(1-x_j^{(i)}))/(1-\theta_{jc}) = 0$$

After solving for θ_{jc}, we get our MAP estimate $\hat{\theta}_{jc} = (\sum_{i=1}^{n} t_c^{(i)} x_j^{(i)} + 2)/(\sum_{i=1}^{n} t_c^{(i)} + 4)$

f) Average log-likelihood for MLE is nan because of (c)
Average log-likelihood for MAP is -3.357063137860283
Training accuracy for MAP is 0.8352166666666667
Test accuracy for MAP is 0.816



Generating from a Naive Bayes Model

a) The naive assumption of a Naive Bayes model assumes that features are conditionally independent given class c. Thus, this is true.
b) Conditional independence does not imply independence, so this is false.