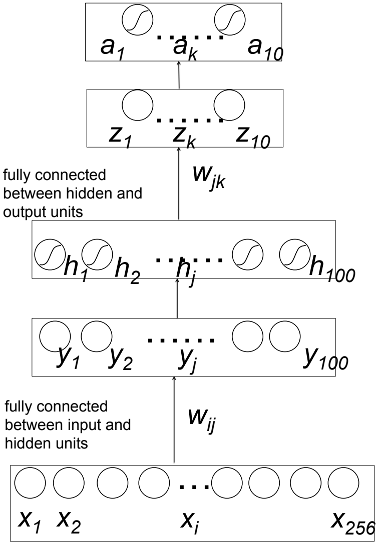
CSC321 Tutorial 7: A2 and Midterm Review

Yue Li
Email: yueli@cs.toronto.edu

Wed 11-12 March 5
Fri 10-11 March 7

Neural Network Schematics	Neural Network information	Variable in classbp2.m
	softmax output	output
	10 output units	outsum
	100 X 10 hidden-to-output w_{jk}	hidout
	100 hidden activities	hidacts
	100 hidden units	hidsum
	256 X 100 input-to-hidden w_{ij}	inhid
	16 X 16 = 256 input units	data

a_1 ...... a_k ...... a_{10}
z_1 ...... z_k ...... z_{10}
w_{jk}
fully connected between hidden and output units
h_1 h_2 ... h_j ... h_{100}
y_1 y_2 ...... y_j ...... y_{100}
w_{ij}
fully connected between input and hidden units
x_1 x_2 ... x_i ... x_{256}

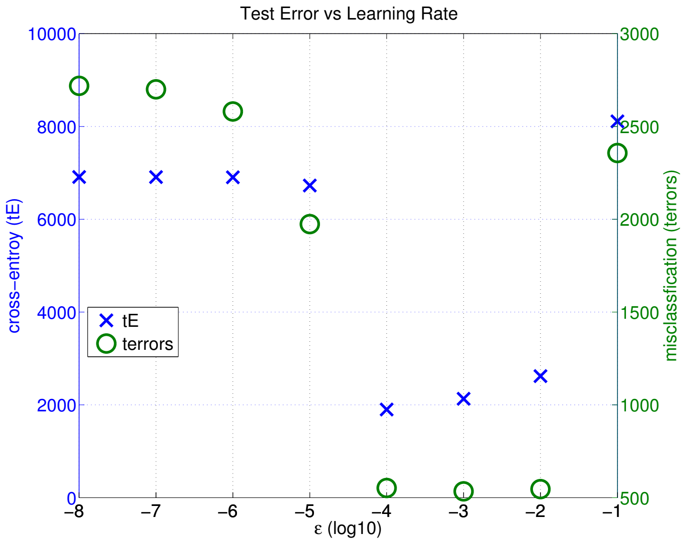
Weight Update (related to part 1&2)

Math	Matlab
$\Delta w_{ij}^{(t)} = \eta \Delta w_{ij}^{(t-1)} - \epsilon(\frac{\partial E}{\partial w_{ij}} + \lambda w_{ij}^{(t-1)})$	`inhidinc = momentum*inhidinc - epsilon*(dEbydinhid + weightcost*inhid);`
$\Delta w_{jk}^{(t)} = \eta \Delta w_{jk}^{(t-1)} - \epsilon(\frac{\partial E}{\partial w_{jk}} + \lambda w_{jk}^{(t-1)})$	`hidoutinc = momentum*hidoutinc - epsilon*(dEbydhidout + weightcost*hidout);`

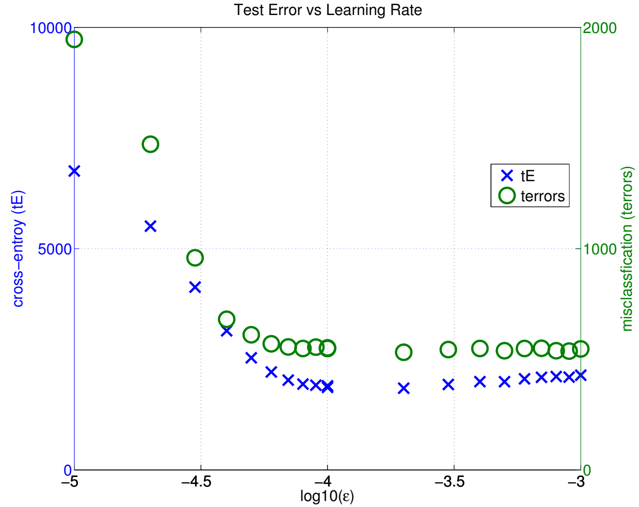
PART 1 (3 points)

- Using `numhid=100` and `maxepoch=2000` and `weightcost=0`, play around with `epsilon` and `finalmomentum` to find settings that make `tE` low after 2000 epochs.
- Briefly report what you discover. Include the values of `epsilon` and `finalmomentum` that work best and say what values they produce for the **test** errors and the cross-entropy error.
- If you were instead asked to find the `epsilon` that produced the **best minimum** value (not the best final value) for test set cross entropy, would you expect to find a larger or smaller `epsilon` (assuming the minimum value occurs *before* the last epoch)? In a sentence, justify your answer.

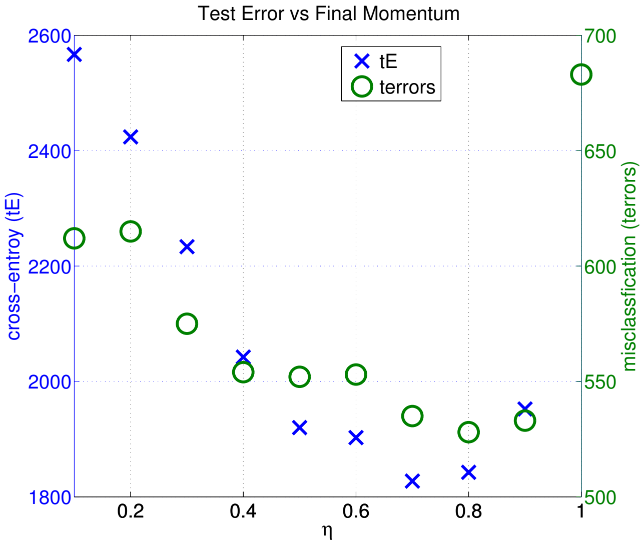
1. The best $\epsilon = 10^x$ was first searched within a wide range, where $x \in [-8, -1]$. The minimum test error (tE) was observed between $\epsilon = 10^{-5}$ and 10^{-3}.



2. A finer search within the smaller ranges of $[10^{-5} : 10^{-5} : 10^{-4}, 10^{-4} : 10^{-4} : 10^{-3}]$ found the best $\epsilon = 2.0e - 04$ with tE $= 1831.45$.



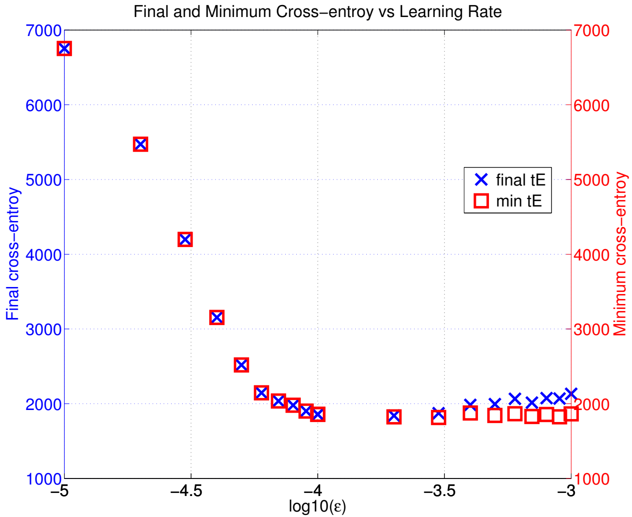
3. Fixing $\epsilon = 2.0e-04$, numhid=100, maxepoch=2000, weightcost=0, the best final momentum was searched in the range [0.1 : 01 : 1] and found to be **0.7** with $\texttt{tE} = \mathbf{1827.25}$.



4. Thus, the results suggest the best epsilon = 2.0e-04 and finalmomentum = 0.7, which gives the minimum tE = 1827.25 with other setting fixed to numhid=100, maxepoch=2000, weightcost=0.
5. Remark: The small learning rate and a fairly large final momentum suggest that the network is prone to overfitting at the beginning of the training phase.

   But once the network enters a stead progress toward a global/local optimal state, the learning can be accelerated with a large momentum to obtain the best setting within a reasonable number of iterations (maxepoch = 2000).

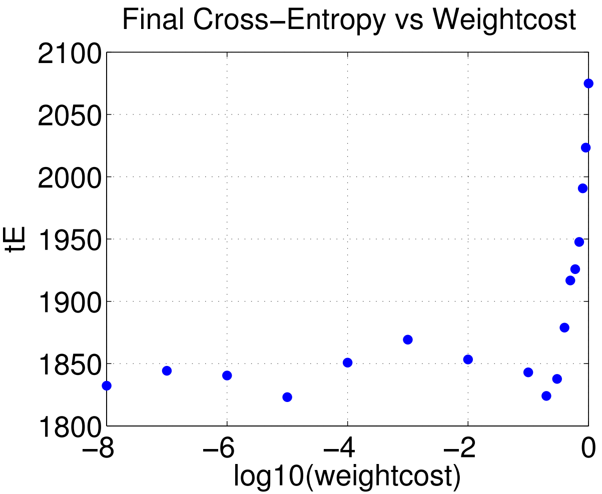
6. An epsilon **slightly larger** than the epsilon (2e-4) found above would be expected to produce a smaller minimum cross-entropy that occurs before the last epoch.



At epsilon = 2e-4, the minimum and final cross-entropy coincide whereas the minimum cross-entropy is slightly better at 1e-3.5 despite the worse final cross-entropy.

This trend shows that the network can still be improved with slightly larger epsilon within the 2000 epochs. However, a even larger epsilon (e.g. 0.1) will "skip" the minimum cross-entropy and is worse off for both values.

PART 2 (2 points): Using the best epsilon = 2e-4 found in Part 1 together with numhid=100, maxepoch=2000, finalmomentum=0.7, it was found that tE generally increases with increasing weightcost.



Common issues in part 1 or 2:

- If fixing the best tE, the epsilon and final momentum are inversely correlated in a non-linear way. That is, if epsilon is large (e.g., 1e-3), final momentum has to be small (e.g., 1e-2) and vice versa.
- What's the relation b/w epsilon and weightcost: The epsilon and weightcost are positively correlated. That is, for small epsilon (e.g., 1e-4), weightcost is almost not needed, which can be set to zero. But if you find a large epsilon in step 1 (e.g., 1e-3), then with finalmomentum now fixed at 0.7, you will need a large weightcost to avoid overfitting.

PART 3: (5 points)

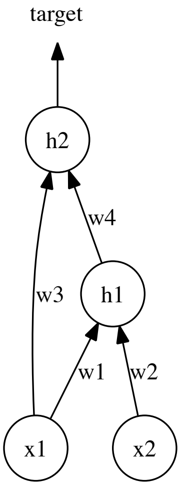
$$\hat{t} = h_2$$

$$h_2 = \frac{1}{1 + \exp(-z_2)}$$

$$z_2 = w_3 x_1 + w_4 h_1$$

$$h_1 = \frac{1}{1 + \exp(-z_1)}$$

$$z_1 = w_1 x_1 + w_2 x_2$$



$$p(\mathbf{W}|\mathbf{D}) = \frac{p(\mathbf{W}) \cdot p(\mathbf{D}|\mathbf{W})}{p(\mathbf{D})} \qquad \text{(Bayes' Rule)}$$

where

- the weight vector is $\mathbf{W} = (w_1, w_2, w_3, w_4)$;
- the input data matrix $\mathbf{D}$ contains N training cases of (x_1, x_2) (i.e. an $N \times 2$ matrix);
- $p(\mathbf{W})$ is the prior (belief) for the weight $\mathbf{W}$ (assumed to be **uniformly** distributed);
- $p(\mathbf{D}|\mathbf{W}) = \exp(-cost)$ is the likelihood under the weights $\mathbf{W}$, where $cost = -\sum_{i=1}[t_i \log \hat{t}_i + (1 - t_i) \log(1 - \hat{t}_i)]$
- $p(\mathbf{D}) = \sum_{\mathbf{W}} p(\mathbf{W}) p(\mathbf{D}|\mathbf{W})$ probability of the data, considered as a normalization factor;
- $p(\mathbf{W}|\mathbf{D})$ is the *posterior* probability of the weights.

Prediction on test data $\mathbf{X}_j$:
Bayesian estimate:

$$\hat{t}_j^{(\text{Bayes})} = p(t_j|\mathbf{X}_j) = \sum_{\mathbf{W}} p(\mathbf{W}|\mathbf{D}) p(t|\mathbf{X}_j, \mathbf{W})$$

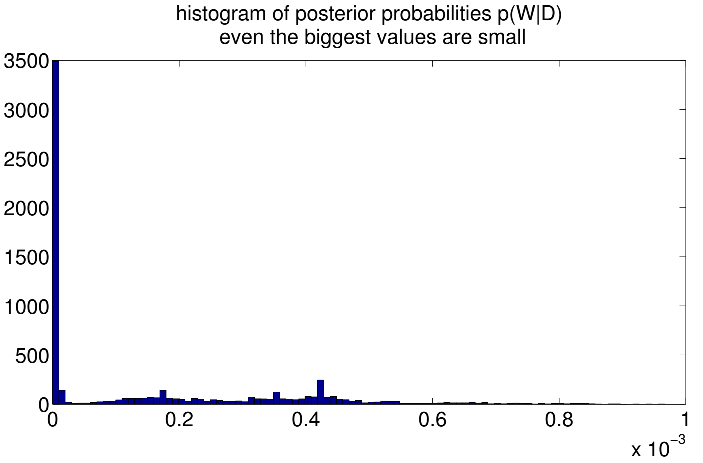
MAP:

$$\begin{aligned}
\mathbf{W}_{MAP} &= \arg\max_{\mathbf{W}} p(\mathbf{W}|\mathbf{D}) \\
&= \arg\max_{\mathbf{W}} \frac{p(\mathbf{W})p(\mathbf{D}|\mathbf{W})}{p(\mathbf{D})} \\
&= \arg\max_{\mathbf{W}} p(\mathbf{W})p(\mathbf{D}|\mathbf{W}) \\
&= \arg\max_{\mathbf{W}} p(\mathbf{D}|\mathbf{W}) \qquad \text{(for uniform prior)} \\
&= \mathbf{W}_{ML}
\end{aligned}$$

$$\hat{t}_j^{(\text{MAP/ML})} = p(t_j|\mathbf{X}_j) = p(t_j|\mathbf{X}_j, \mathbf{W}_{MAP})$$

Bayesian estimate (bayespredictions):

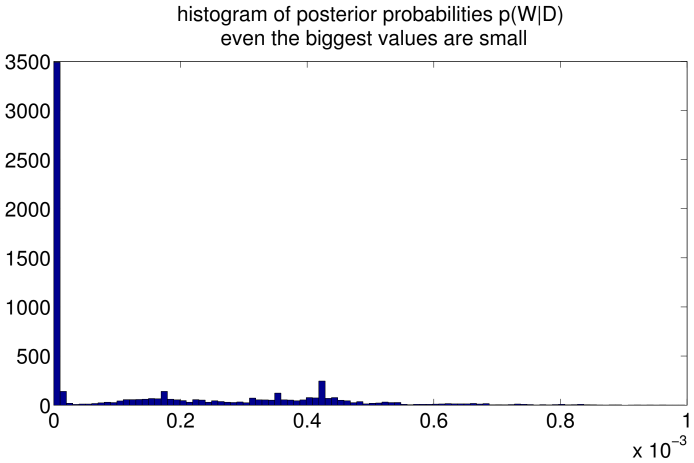
$$p(t|D) = \sum_{\forall \mathbf{W}} p(\mathbf{W}|\mathbf{D})p(t|\mathbf{D}, \mathbf{W})$$



MAP (bestpredictions):

$$\mathbf{W}_{MAP} = \arg\max_{\mathbf{W}} p(\mathbf{W}|\mathbf{D})$$

$$p(t|\mathbf{D}) \equiv p(t|\mathbf{D}, W_{MAP})$$



Evaluation (Contrastive Divergence):

$$\begin{aligned} ErrorSum = & -\sum_{i=1}^{N'} \left[t_n \log(\hat{t}_i) + (1 - t_i)\log(1 - \hat{t}_i)\right] \\ & + \sum_{i} [t \log(t_i) + (1 - t_i)\log(1 - t_i)] \end{aligned}$$

$$\textbf{ErrorPerBit} = \frac{ErrorSum}{N' \log(2)}$$

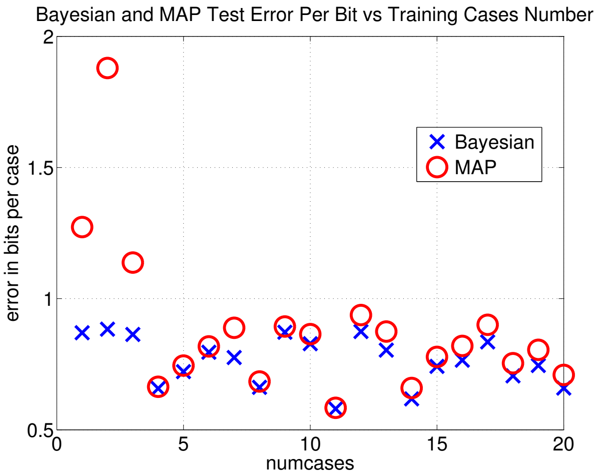
where N' is the number of test cases (i.e., `testnumcases`).

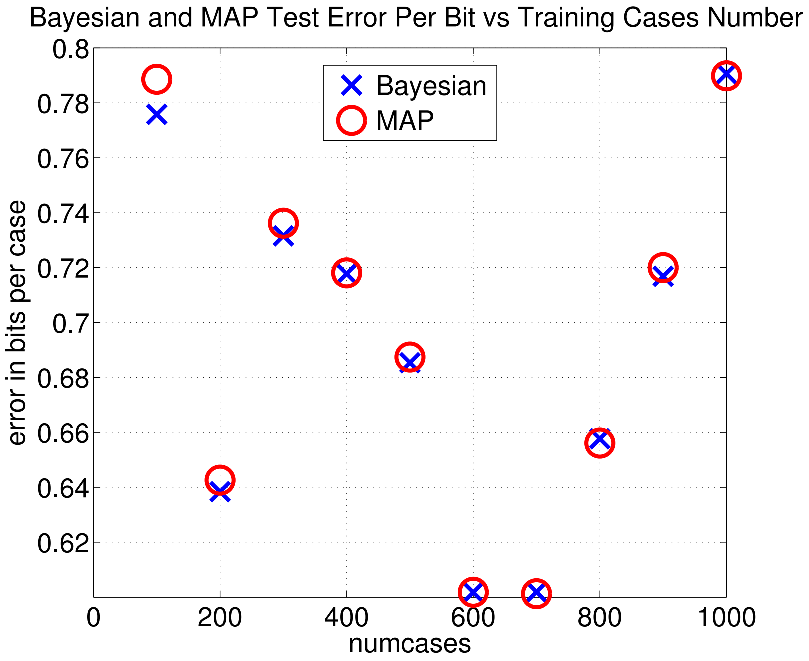
Note: Use **ErrorPerBit** to evaluate Bayesian and MAP estimate rather than ErrorSum.

Describe the effects of changing the number of training cases.

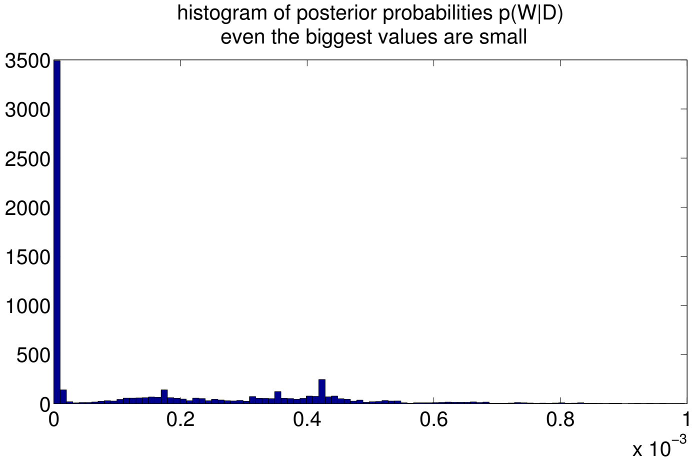
1. Experimental approach:
   - To have a stable estimate of the model performance, the experiment was repeated 10 times for the same number of training cases.
   - In each time the four weights from teacher net were randomly sampled from the uniform distribution (-1,1).
   - The averaged bayestesterr per bit and besttesterr per bit were then taken over the 10 experiments (for the same number of training cases).

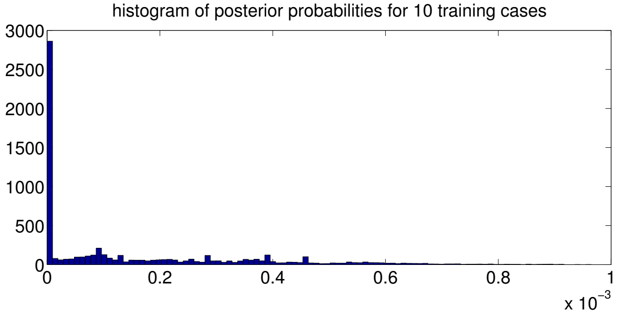
2. With varying number of training cases (1:1:20 and 100:100:1000), the Bayesian estimation has a more robust and generally better performance than MAP/ML method in terms of the discrepancy in cross-entropy when comparing with the ideal model estimate (i.e. the teacher net).

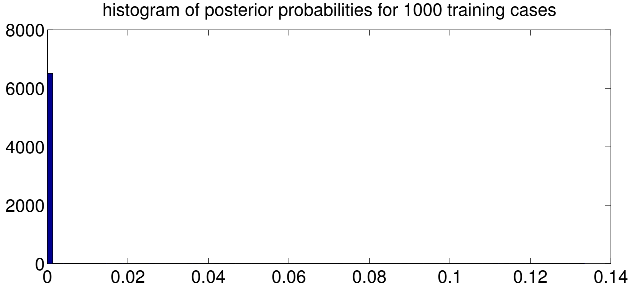



Bayesian and MAP Test Error Per Bit vs Training Cases Number
Bayesian
MAP
error in bits per case
numcases
0.8
0.78
0.76
0.74
0.72
0.7
0.68
0.66
0.64
0.62
0
200
400
600
800
1000

3. Bayesian method averages over all of the model estimates weighted by their posterior probability. Thus, the approach spreads the uncertainty across the entire model spectrum. In contrast, MAP/ML method only employs the model with the highest posterior probability to predict test data.




histogram of posterior probabilities for 10 training cases
3000
2500
2000
1500
1000
500
0
0
0.2
0.4
0.6
0.8
1
x 10^-3


histogram of posterior probabilities for 1000 training cases
8000
6000
4000
2000
0
0
0.02
0.04
0.06
0.08
0.1
0.12
0.14

4. For smaller number training cases (numcases < 100), the histogram of the posterior probability distribution suggests that even the highest posterior probability is only about 0.001. Thus, considerable uncertainty exists even in "best" model.

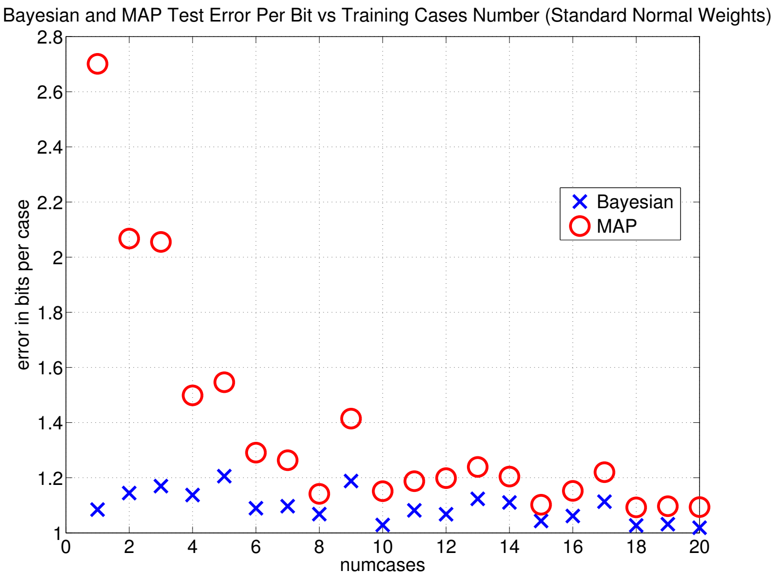
   When the number of training cases increases to 1000, the highest posterior probability increases by more than 2 orders to 0.14.

5. The results explain why MAP/ML performs more poorly than Bayesian method when the number of training cases is small but become more comparable to the latter method for larger number of training cases.

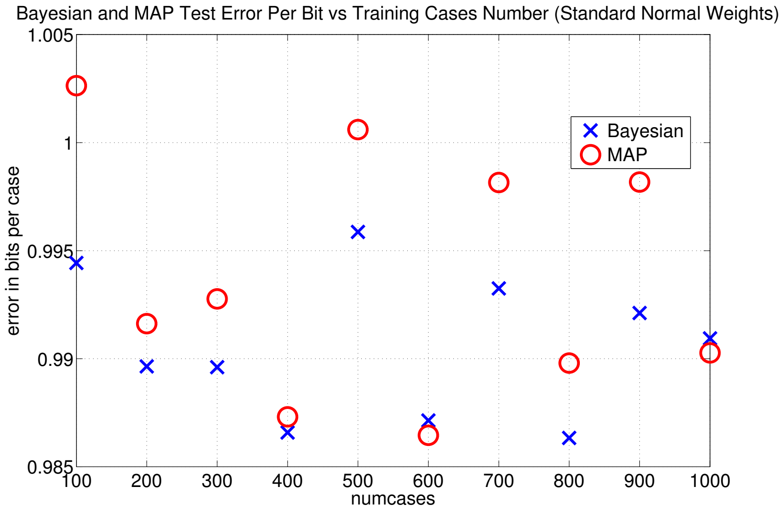
- **Part3 b**: Modifying maketeacher by changing `wteacher = 2*rand(1,4) - 1;` to
  - `wteacher = randn(4, 1);`

    or

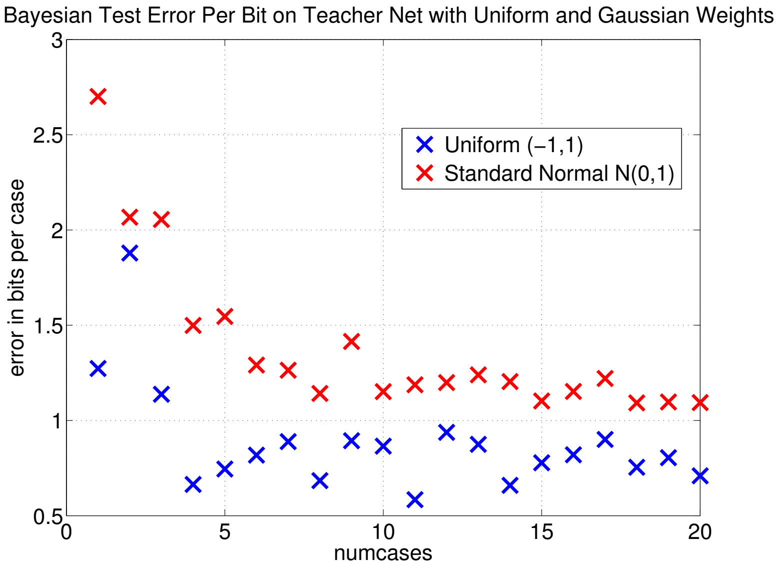
  - `wteacher = mean(randn(5000,4), 1);`



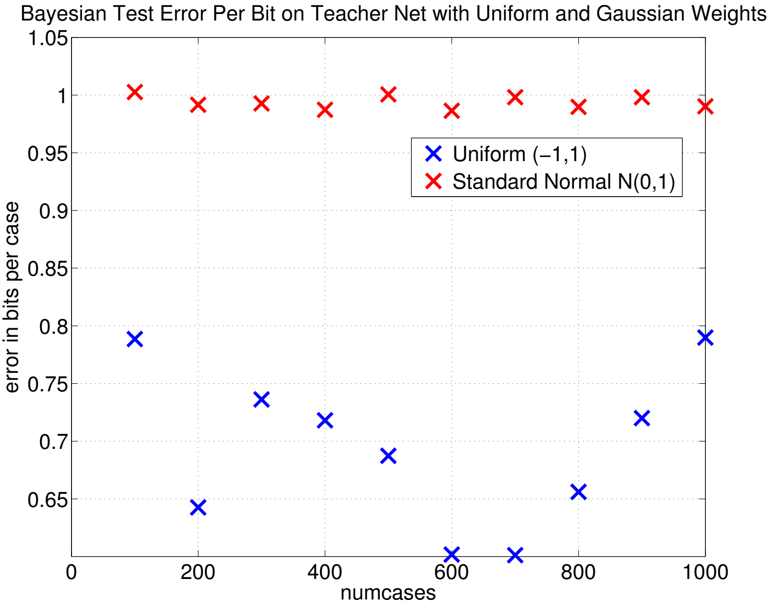
Bayesian and MAP Test Error Per Bit vs Training Cases Number (Standard Normal Weights)
error in bits per case
numcases
Bayesian
MAP
2.8
2.6
2.4
2.2
2
1.8
1.6
1.4
1.2
1
0
2
4
6
8
10
12
14
16
18
20



Bayesian and MAP Test Error Per Bit vs Training Cases Number (Standard Normal Weights)
error in bits per case
numcases
1.005
1
0.995
0.99
0.985
100
200
300
400
500
600
700
800
900
1000
Bayesian
MAP



Bayesian Test Error Per Bit on Teacher Net with Uniform and Gaussian Weights
error in bits per case
numcases
0
5
10
15
20
0.5
1
1.5
2
2.5
3
Uniform (−1,1)
Standard Normal N(0,1)



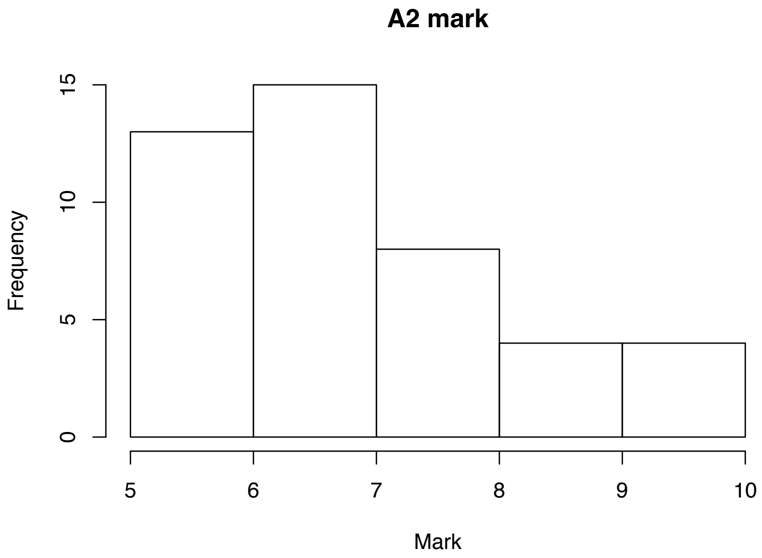
Bayesian Test Error Per Bit on Teacher Net with Uniform and Gaussian Weights
error in bits per case
numcases
1.05
1
0.95
0.9
0.85
0.8
0.75
0.7
0.65
0
200
400
600
800
1000
Uniform (−1,1)
Standard Normal N(0,1)

6. When the true weights of teacher net were sampled from a standard normal distribution (N(0,1)), (or other non-uniform distribution), the assumption of a uniform prior used in both Bayesian and MAP/ML methods is violated. As expected, the performance of both method decreases. However, Bayesian is still more robust than MAP/ML.

Common issues in part 3:

- How do Bayesian and MAP/ML compare and why one is better than the other?
- How does changing wteacher to non-uniform affect both models?
- Due to random fluctuation, the test needs to be repeated multiple times and then average the error per bits to obtain a reliable error estimate.

A2 Marks distribution



Midterm Review

TA office hours now-1:30 at DV1160