Name:

| Problem | Points | Score |
|---------|--------|-------|
| 1 | 50 | |
| 2 | 50 | |
| Total | 100 | |

Notes:

(1) The exam is closed books and notes except for one double-sided sheet of notes.

(2) Please indicate clearly your answer to the problem.

(3) If I can't read or follow your solution, it is wrong and no partial credit will be awarded.

**(50 pts) Problem No. 1**: A machine produces a sequence "ABABAB".

(a) Design a two-state fully ergodic observable discrete Markov model that is capable of generating this sequence. Estimate the parameters of the model using a maximum likelihood approach.

(b) Compute the probability that model produced the sequence "ABB".

(c) Design a two-state fully ergodic hidden discrete Markov model, again using a maximum likelihood approach for parameter estimation.

(d) Compute the probability that the model produced the sequence "ABB" and compare this result to (b). Explain whether your result makes sense.

**(50 pts) Problem No. 2:** You are given a training set that consists of random samples from two uniform distributions. The first distribution, associated with Class 1, is a two-dimensional uniform distribution with a mean of (0,0) and extends from $-1 \leq x \leq 1$ and $-1 \leq y \leq 1$. The second distribution has a mean of (0.5,0.5) and extends from from $0 \leq x \leq 1$ and $0 \leq y \leq 1$.

(a) Compute the probability of error that will be achieved on the training set (closed-set testing) by a classifier designed using a nearest-neighbor algorithm (e.g., kNN) based on a majority voting scheme (as we discussed in class).

(b) Suppose Class 1 was actually perfectly represented by a Gaussian distribution of mean (0,0) and an identity covariance matrix. Supposed Class 2 was perfectly represented by a Gaussian distribution of mean (0.5,0.5) and an identity covariance matrix. Write an equation for the probability of error if you applied the classifier trained in (a).

(c) Explain how your answers to (a) and (b) would change if you used a support vector machine rather than a kNN algorithm. Would the error rate increase or decrease if the system were trained properly?