CS.542 Machine Learning, Fall 2019, Prof. Saenko

Additional Exam Practice Problems

Note: Some of these sample problems had been used in past exams and are provided for practice in addition to the midterm practice and homework problems, which you should also review. A typical exam would have around 5 questions. The exam is closed book, no electronics (no phones, calculators, laptops, tablets, smart watches, etc.) A single 8"x11" sheet of paper with typed or handwritten notes on both sides is allowed.

Table of Contents

I. Math and Probability Basics

Q1: Review of Definitions

(a) For a scalar random variable x, give the definition of its mean and variance.

Answer: see Bishop Ch. 1

(b) For a vector random variable $x \in \mathbb{R}^n$, give the definition of its mean and covariance.

Answer: see Bishop Ch. 1

Q2: Short questions

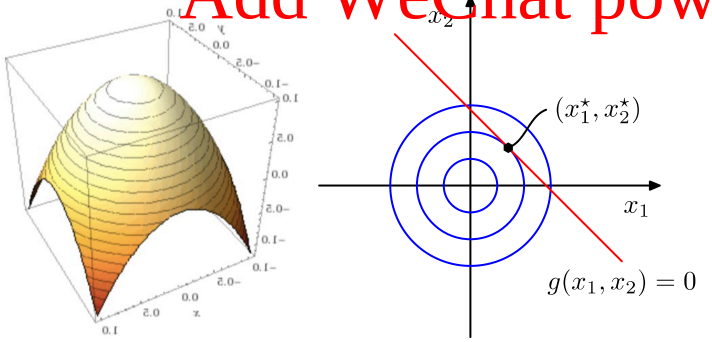
(a) Given a vector $v \in \mathbb{R}^2$ and a line (through the origin) orthogonal to the vector $u \in \mathbb{R}^2$, draw the orthogonal distance from v to the line, and write down the equation for the distance.

Answer: $v^T u/\|u\|$, or dot product of u and v divided by the length of u

(b) A *hyperplane* is a subspace whose dimension is one less than that of its ambient space. Suppose the line in (c) was instead an n-1 dimensional hyperplane in the ambient space $\mathbb{R}^n$. What is the orthogonal distance from a point $v \in \mathbb{R}^n$ to the hyperplane?

Answer: $v^T u/\|u\|$, same as above but in higher dimensions

(c) Find the maximum of the function $f(x_1, x_2) = 1 - x_1^2 - x_2^2$ subject to the constraint $g(x_1, x_2) = x_1 + x_2 - 1 = 0$, using Lagrange Multipliers. The 3-D plot shows f and the 2-D plot shows its contours (blue circles) and the line corresponding to $g = 0$ (red line).



Answer: solve the following system of equations:

$$\nabla f(x_1, x_2) = \lambda \nabla g(x_1, x_2)$$
$$g(x_1, x_2) = 0$$

In this case the first equation yields $-2x_1 = \lambda$; $-2x_2 = \lambda$. Solving for x_1, x_2 and substituting into the second equation, we get $\lambda = -1$, so $\mathrm{x} = (0.5, 0.5)$ is the solution. For more info: http://www.math.harvard.edu/archive/21a_spring_09/PDF/11-08-Lagrange-Multipliers.pdf

(d) Show that for a matrix A and vector x, $\frac{\partial}{\partial x}(A^{-1}) = -A^{-1}\left(\frac{\partial}{\partial x}A\right)A^{-1}$. Use the fact that for any two matrices A and B, $\frac{\partial}{\partial x}(AB) = \frac{\partial A}{\partial x}B + A\frac{\partial B}{\partial x}$.

Answer: since $\frac{\partial}{\partial x}(AA^{-1}) = \frac{\partial A}{\partial x}A^{-1} + A\frac{\partial A^{-1}}{\partial x} = 0$, then $A\frac{\partial A^{-1}}{\partial x} = -\frac{\partial A}{\partial x}A^{-1}$ and after left-multiplying by A^{-1} we get the above result.

II. General Machine Learning Concepts

Q1. True or False questions

Circle the correct answer (T or F) and give a one-sentence explanation of each answer; answers without explanation will not be given full points.

a) Supervised classification models can be used to predict if an email attachment contains a computer virus [T/F]

Answer: true, eg binary classification with an SVM

b) Suppose we use polynomial features for linear regression, then the hypothesis is linear in the original features [T/F]

Answer: false, it is linear in the new polynomial features

c) An advantage of normal equations to solve linear regression is that choosing a step size (learning rate) is not necessary [T/F]

Answer: true, it's a closed form solution so no need for gradient descent

d) Maximum likelihood can be used to derive a closed-form solution to logistic regression [T/F]

Answer: false, it can be used to derive cost, but no closed form solution exists

e) The gradient descent update for logistic regression is identical to linear regression [T/F]

Answer: false, they look similar but the hypothesis functions are different

f) Changing the prior in Linear Discriminant Analysis changes the direction of the decision hyperplane [T/F]

Answer: false, changes only the position, not the direction of the decision hyperplane

Q2. Short answer questions

Answer the following questions in brief one to two sentence answers.

a) For a training dataset $\mathrm{D} = \{x_i, y_i\}$ where x_i are the inputs and y_i are the output, explain the difference between discriminative and generative classification models.

Answer: A generative model learns $p(x, y)$ which can *generate* examples $\{x, y\}$ and evaluate $p(y)$, $p(y|x)$ and $p(x|y)$, while a discriminative model only learns $p(y|x)$ or learns the decision boundary directly without modeling the output probability (eg. in the SVM).

b) Give one example of a discriminative classification model.

Answer: logistic regression (others are neural network, SVM)

c) Give one example of a generative model.

Answer: for classification, linear discriminant analysis (LDA) (others we did not cover are Naïve Bayes, Graphical Models such as HMM); for unsupervised learning, GAN, GMM

d) Suppose you want to use training data D to adjust the parameters w of a model where $\mathrm{L(D)} = \mathrm{p(D; w)}$ is the likelihood of the data. You want to prevent overfitting using a squared norm regularizer. What should your objective function look like? Should you minimize or maximize it?

Answer: minimize the following objective (note, this maximizes $L(D)$):

$$J = -L(D) + \lambda\|w\|^2 \quad \text{or,} \quad J = -\lambda L(D) + \|w\|^2$$

where λ is a hyperparameter, or, equivalently, maximize $-J$.

e) What is cross-validation?

Answer: When we split training data into training and extra validation set; learn model parameters on the training set, test and tune hyper-parameters on the validation set. We can do this multiple times, taking a different portion of original training data for the validation set each time (known as N-fold cross-validation).

f) How can we use cross-validation to prevent overfitting? Explain the procedure using the setup of (d).

Answer: Train several models using different values of λ on the training set, test them on the validation set, and pick best model. This will reduce overfitting compared to tuning λ on training data.

Q3. Error metrics

a) Give one example each of error metrics that can be used to evaluate: classification, regression, clustering, and reinforcement learning.

Answer: accuracy, squared error (mean squared error), average distance to cluster centers, total reward.

b) Which are the correct definitions of precision and recall? Here 'actual positives' are examples labeled positive (by humans), and 'predicted positives' are examples for which the algorithm predicts a positive label.

1. $precision = \frac{true\ positives}{predicted\ positives}$ 2. $precision = \frac{true\ positives}{actual\ positives}$

3. $recall = \frac{predicted\ positives}{actual\ positives}$ 4. $recall = \frac{true\ positives}{actual\ positives}$

Answer: 1 and 4

c) Suppose your data has binary labels, where the negative class (y=0) occurs 99% of the time and the positive class (y=1) occurs 1% of the time. Which of the following results obtained by your machine learning algorithm *by itself* confirms that it is performing well on this data (circle one)? *Hint: think about what happens if your algorithm always outputs '0', or always outputs '1'.*

1. High Recall 2. High Accuracy 3. High F1 score

Answer: 3, because it combines recall and precision. A model can have high recall but low precision, e.g. always predicting '1' will have 100% recall but 0% precision (F1=0). A model can have high accuracy but low F1 score, e.g. always predicting '0' will be 99% accurate but F1=0.

III. Bayesian Methods

Q1. Bias-variance in Bayesian models

Alice has a dataset of m points with n-dimensional inputs and scalar outputs. She has trained several regularized linear regression models using regularization parameters $\lambda = e^0, e^{-1}, e^{-2}, e^{-3}$.

a) Which parameter will lead to highest bias? To highest variance?

Answer: since λ weights the regularizer, the smallest λ will lead to highest variance and largest λ will lead to highest bias.

b) Alice then decides to use a Bayesian approach to control the complexity of the model. What is the Bayesian equivalent to changing λ?

Answer: changing the prior on the parameters of the model

c) Which Bayesian model should she use? Explain what makes the model Bayesian.

Answer: Bayesian linear regression, it puts a prior distribution over the linear parameters

d) Alice fit the parameters of her model and wants to use the predictive distribution on new inputs. Explain what the predictive distribution is.

Answer: it models the conditional probability of the output given an input and training data.

e) What is the difference between the predictive distribution of the ML-based and the Bayesian linear regression models?

Answer: they are both Normal distributions, but their means and covariances are computed differently; ML finds a point estimate of the parameters while the Bayesian solution integrates over parameter values using the posterior distribution. See Bishop for more details.

Q2. Bayesian Maximum A Posteriori (MAP)

Suppose we are estimating the probability of seeing 'heads' ($x = 1$) or 'tails' ($x = 0$) after tossing a coin, with μ being the probability of seeing 'heads'. The probability distribution of a single binary variable $x \in \{0,1\}$ that takes value 1 with probability μ is given by the *Bernoulli* distribution

$$Bern(x|\mu) = \mu^{x}(1-\mu)^{1-x}$$

Suppose we have a dataset of independent coin flips $D = \{x^{(1)}, x^{(2)}, \dots, x^{(m)}\}$ and we would like to estimate μ using the Bayesian MAP solution. Recall that we can write down the data likelihood as

$$p(x^{(i)}|\mu) = \mu^{x^{(i)}}(1-\mu)^{1-x^{(i)}}$$

Consider the following Bayesian prior on μ, which believes that the coin is either fair, or slightly biased towards 'tails':

$$p(\mu) = \begin{cases} 0.5\ if\ \mu = 0.5 \\ 0.5\ if\ \mu = 0.4 \\ 0\ otherwise \end{cases}$$

Write down the MAP estimate for μ under this prior as a function of the likelihood and the prior. *(Hint: use the argmax function).*

Answer: a MAP estimate is obtained by maximizing the posterior, using Bayes rules and dropping the p(D) denominator,

$$\mu_{MAP} = \underset{\mu}{\operatorname{argmax}}\, p(\mu|D) = \underset{\mu}{\operatorname{argmax}}\, p(D|\mu)\, p(\mu)\,,$$

where

$$p(D|\mu)\, p(\mu) = \begin{cases} p(D|\mu) * 0.5\ if\ \mu = 0.5 \\ p(D|\mu) * 0.5\ if\ \mu = 0.4 \\ 0\ otherwise \end{cases}$$

and

$$p(D|\mu) = \prod_{i=1}^{m} p(\,x^{(i)}|\mu) = \prod_{i=1}^{m} \mu^{x^{(i)}}(1-\mu)^{1-x^{(i)}}$$

IV. Unsupervised Learning

Q1. Short questions

a) For which problems below would anomaly detection be a suitable algorithm? Explain.

1. Given an image of a face, determine whether or not it is the face of a particular famous individual.
2. From a large set of hospital patient records, predict which patients have a particular disease (eg. flu)
3. Given a dataset of credit card transactions, identify transactions to flag them as possibly fraudulent.
4. From a large set of primary care patient records, identify individuals who might have unusual health conditions.

Answer: 3 and 4, because in both cases the non-anomaly class is very large and the anomaly class cannot be easily defined but rather is "everything else".

b) Suppose you have trained an anomaly detection system for intruder detection in a security camera, and your system flags anomalies when p(x) is less than ε. You find on the cross-validation set that it is missing many intruder events. What should you do?

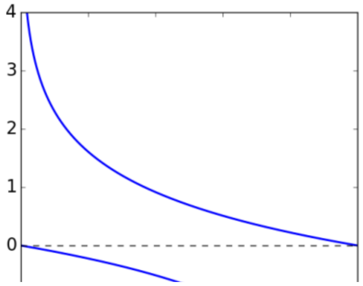
Answer: Try increasing ε to flag more examples as anomalies; can also collect more training data to get a better model of p(x).

c) Which of the following approaches use density estimation? Explain.

1. Linear discriminant analysis
2. Logistic Regression
3. Anomaly detection
4. Generative Adversarial Networks

Answer: 1,3 and 4. LDA and anomaly detection use explicit density estimation (Gaussian density) but GANs use implicit density estimation i.e. they can sample from the density but cannot estimate its value.

Q2. Generative adversarial network (GAN) mini-max loss



A GAN consists of a generator and a discriminator: the generator network $G(z)$ has parameters θ_G, takes a random noise vector z as input, and outputs a sample of data x (e.g., an image); the discriminator network $D(x)$ has parameters θ_D, takes in x, and outputs a binary label $y \in \{0,1\}$ where 1 indicates that the input is real and 0 that it is fake. The discriminator's cost function is the cross-entropy loss for the binary classification task over the real and the generated inputs:

$$J_D(\theta_D) = \mathbb{E}_x[-\log D(x)] + \mathbb{E}_z[-\log(1 - D(G(z)))]$$

Suppose the generator's loss is the opposite (negative) of the discriminator's cross entropy loss.

a) The generator's loss is shown in the plot above. Write down the equation for it.

Answer: $J_G(\theta_G) = \mathbb{E}_z[\log(1 - D(G(z)))]$. Note that the term that doesn't depend on G is constant so we omit it here.

b) Explain why the generator's loss J_G suffers from slow learning.

Answer: if the discriminator is good at predicting that the generator's samples are fake (the input to the loss is close to zero), gradients are flat, which slows down learning.

c) Why doesn't this happen in the regular cross entropy loss?

Answer: in regular cross-entropy, the term involving samples with label 0 corresponds to the negative of the generator's loss, but in that case, if the classifier is incorrect its outputs are close to 1 and the loss is not flat there.

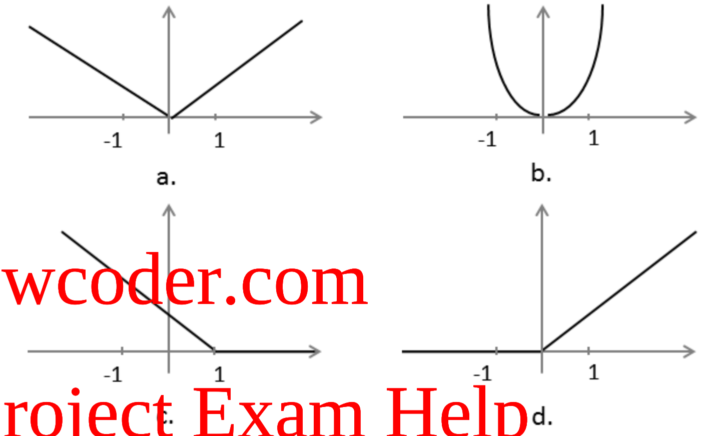
V. Loss Functions, SVM, Kernels

Q1. Hinge Loss

Given the SVM decision function $f(x) = w^T x + b$ and labels $y \in \{-1, +1\}$, the unconstrained primal SVM problem is to minimize the following w.r.t. w over the training points i:

$$\|w\|^2 + C \sum \max\left(0, 1 - y_i f(x_i)\right)$$

The loss function $\max\left(0,1 - y_i f(x_i)\right)$ is called *Hinge Loss*. Which of these four loss functions implements the SVM hinge loss? Explain in words what the hinge loss does.



Answer: (c). Input to the loss is distance to decision boundary times the label: $yf(x) = y(w^T x + b)$. The loss assigns no loss to examples with distance more than +1 from decision boundary, otherwise penalty increases linearly with decreasing distance from decision boundary. The y factor controls direction, i.e. loss is reflected around 0 for negative y. (d) is correct if we instead use $-y((w^T x + b) - 1)$ as input. By default, (c) is correct.

Q2. Loss functions for car sales

Alice works for a used car dealership. Her boss wants her to estimate the price y to charge for a car (in dollars) based on features such as: x_1= car manufacturer, x_2= model, x_3= distance driven in miles, x_4= age in years, etc. She collects data points from previous car sales, $(x^{(i)}, y^{(i)})$, $i = 1, \dots, m$, where $y^{(i)}$ is the price the car was sold for, and decides to use a linear regression model, $y = \sum_{j=0}^{n} \theta_j x_j$.

a) Alice decides to minimize a sum of squares loss function. Does it make sense in this case? What effect will it have?

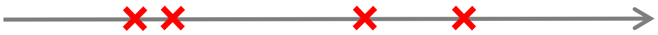
Answer: Yes; it will encourage the sale price to be the same as previous years.

b) Her boss tells Alice she strongly prefers that the dealership not lose money on a sale. Given that the ith car cost the dealer $z^{(i)}$, suggest a way to pre-process the training data to encourage this.

Answer: Set y(i) to be the maximum of y(i) and z(i), then apply the SSE loss function as before.

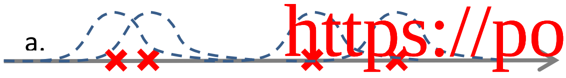
Q3. Kernels and PCA

Consider the following dataset of 1-dimensional datapoints:

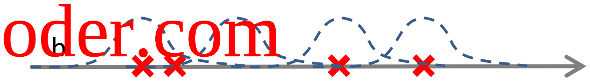


a) Which placement of Gaussian basis functions corresponds to a kernel feature representation for this dataset?

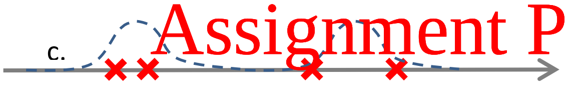
a.



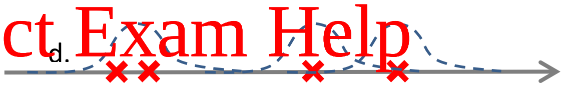
b.



c.



d.



Answer: (a); we place a Gaussian distribution centered at each data point.

For $k(x_1, x_2)$ to be a valid kernel, there must be a feature basis function $\varphi(.)$ such that we can write $k(x_1, x_2) = \varphi(x_1)^T\varphi(x_2)$.

b) Prove that $k = {x_1}^T C x_2$ is a valid kernel, where $C = XX^T$ is the data covariance obtained from the design matrix X. *Hint: use* $(AB)^T = B^T A^T$.

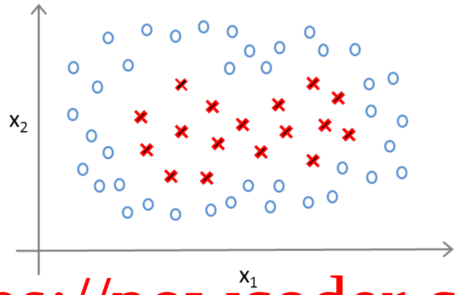
Answer: ${x_1}^T C x_2 = {x_1}^T X X^T x_2 = (X^T x_1)^T X^T x_2; \varphi(x) = X^T x$

c) Typically we use kernels to project data into a higher dimensional space. Principal Component Analysis projects data into a lower dimensional space. Suggest a kernel that uses PCA to project data into a lower, k-dimensional space, and prove that it is a valid kernel. You can assume the data already has a mean of zero. *Hint: use the result of (b).*

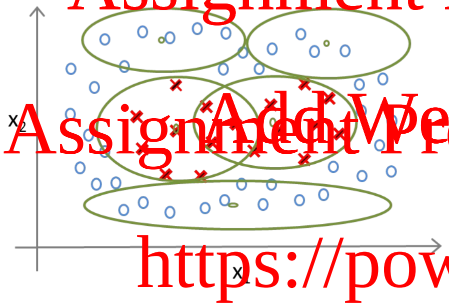
Answer: $k = {x_1}^T U U^T x_2$, use U equal to k top eigenvectors of the covariance matrix. Then use (b) replacing X with U to prove the kernel is valid. The feature function corresponds to $\varphi(x\ \ ) = U^T x$ or the projection only the eigenvectors. Note that since data is assumed to have zero mean, we do not need to subtract the mean before computing $U^T x$.

Q4. SVMs and Kernels

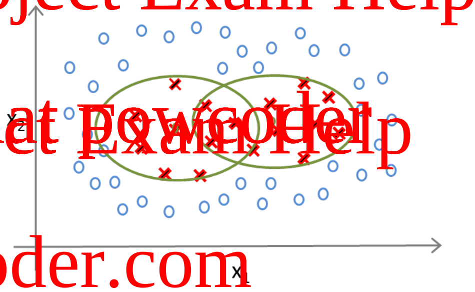
Consider the following dataset of 2-dimensional datapoints:



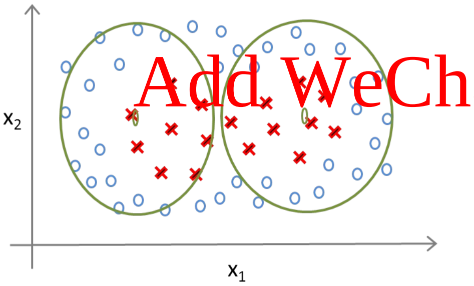
a) Which placement of Gaussian basis functions corresponds to a kernel feature representation for this dataset? Explain your answer in one sentence below.



(A)



(B)



(C)

(D) none of the above

Answer: (D); to compute a kernel representation, we place a separate Gaussian distribution centered at **each** data point.

b) How does increasing the variance of the Gaussian σ^2 affect the bias and variance of the resulting Gaussian Kernel SVM classifier? Explain.

Answer: it makes the boundary smoother, so the classifier has higher bias and lower variance

c) For $k(x_1, x_2)$ to be a valid kernel, there must be a feature basis function $\varphi(.)$ such that we can write $k(x_1, x_2) = \varphi(x_1)^T \varphi(x_2)$. Suppose $k_1(x_1, x_2)$ and $k_2(x_1, x_2)$ are valid kernels. Prove that the following is also a valid kernel:

$$k(x_1, x_2) = k_1(x_1, x_2) + k_2(x_1, x_2)$$

Answer:
$k(x_1, x_2) = k_1(x_1, x_2) + k_2(x_1, x_2) = \phi_1(x_1)^T \phi_1(x_2) + \phi_2(x_1)^T \phi_2(x_2) = \phi(x_1)^T \phi(x_2)$
where $\phi(x) = [\phi_1(x), \phi_2(x)]$

d) Both SVMs with Gaussian kernels and Neural Networks with at least one hidden layer can be used to learn non-linear decision boundaries, such as the boundary between positive and negative examples in the dataset above. Describe the main similarity and the main difference in how these two approaches achieve this.

Answer: The similarity is that both can be thought of as mapping the input features x into a new feature space. The SVM kernel maps x to a new feature vector $\phi(x)$, and the hidden layer of the neural network maps x to the activations of the last hidden layer, a. The difference is in how the mapping is done, where SVM uses training data as landmarks, but neural network learns the feature mapping through layer parameters.

e) Explain what slack variables are used for when training SVMs.

Answer: slack variables are assigned to solve a problem that is not linearly separable by adding ‘slack’ values to the constraints

VI. Reinforcement Learning

Q1. Markov Decision Process

Consider an agent that learns to navigate a simple grid world, shown below. Suppose the Markov Decision Process is given as follows:

States: locations (x,y) on the map
Actions: move right →, left ←, up↑, down↓ or the "do nothing" action ○
Reward: +1 in state (4,4), -0.01 in all other states
Transitions: $P_{state,action}(state')$ is given as

(1,1)	(2,1)	(3,1)	(4,1)
(1,2)	(2,2)	(3,2)	(4,2)
(1,3)	(2,3)	(3,3)	(4,3)
(1,4)	(2,4)	(3,4)	(4,4)

policy π

→	→	→	↓
→	→	→	↓
→	→	→	↓
→	→	→	○

$$P_{(x,y),\rightarrow}(s') = \begin{Bmatrix} 0.9 \; if \; s' = (x+1,y) \\ 0.1 \; if \; s' = (x,y) \\ 0 \; otherwise \end{Bmatrix}$$

$$P_{(x,y),\leftarrow}(s') = \begin{Bmatrix} 0.9 \; if \; s' = (x-1,y) \\ 0.1 \; if \; s' = (x,y) \\ 0 \; otherwise \end{Bmatrix} P_{(x,y),\uparrow}(s') = \begin{Bmatrix} 0.9 \; if \; s' = (x,y-1) \\ 0.1 \; if \; s' = (x,y) \\ 0 \; otherwise \end{Bmatrix} P_{(x,y),\downarrow}(s') = \begin{Bmatrix} 0.9 \; if \; s' = (x,y+1) \\ 0.1 \; if \; s' = (x,y) \\ 0 \; otherwise \end{Bmatrix}$$

(a) Consider the policy π shown above (right). If the agent starts in state (1,1) and takes actions according to this policy, what is the probability that it will end up in state (4,4) after 6 steps? Explain your answer.

Answer: probability of transitioning to the intended state is .9, so the probability that the agent reaches the final state in 6 steps is (0.9)^6

(b) For the same policy π, suppose the agent transitions into state (4,4) for the first time on step 9. What is the total reward collected by the agent? Assume the agent collects reward in initial state and there is no discounting of rewards.

Answer: 1-8*0.01=0.92 if at s1=(1,1), s2= (2,1),… s9=(4,4) and we count the first state as step 1
We don't count the first state as time 1, but collect the reward, then we collect 1-9*.01=.91

(c) Is there a more optimal policy (i.e. with strictly higher reward) for the agent than π? If so, what is it?

Answer: no

(d) What is the main difference between supervised learning and reinforcement learning?

Answer: supervised algorithms learn from input and output pairs, whereas reinforcement learning algorithms learn from observations that depend on the model's actions and from delayed rewards.

Appendix: Useful Formulas

Matrix Derivatives

For vectors x, y and matrix A,

$y = Ax$, then $\frac{\partial y}{\partial x} = A$

If $z = x^T A x$, then $\frac{\partial z}{\partial x} = x^T(A + A^T)$. For the special case of a symmetric matrix A, $\frac{\partial z}{\partial x} = 2x^T A$.

Single Dimension Normal Distribution

$$N(x; \mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

Multivariate Normal Distribution

The p-dimensional multivariate normal distribution is given by

$$N(x; \mu, \Sigma) = \frac{1}{(2\pi)^{\frac{p}{2}}|\Sigma|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right)$$