Final Exam Review
Reminders

• **Homework 9: Learning Paradigms**
  – Out: Wed, Apr. 22
  – Due: Wed, Apr. 29 at 11:59pm
  – Can only be submitted up to 3 days late, so we can return grades before final exam

• **Final Exam Practice Problems**
  – Out: Wed, Apr. 29

• **Final Exam**
  – Mon, May 04 (1pm – 4pm)

• **Today’s In-Class Poll**
  – [http://poll.mlcourse.org](http://poll.mlcourse.org)
EXAM LOGISTICS
Final Exam

• **Time / Location**
  – **Time:** Registrar-scheduled Exam
    **Mon, May 4th at 1:00pm – 4:00pm**
  – **Online Exam:** Same format as Midterm Exam 2
  – Please watch Piazza carefully for announcements logistics

• **Logistics**
  – Distribution of Topics: Lectures 19 – 28 (95%), Lectures 1 – 18 (5%)
  – Format of questions:
    • Multiple choice
    • True / False (with justification)
    • Derivations
    • Short answers
    • Interpreting figures
    • Implementing algorithms on paper
  – You are encouraged to **bring** one 8½ x 11 sheet of notes (front and back)
  – Open book according to my definition on Piazza:
    [https://piazza.com/class/k4wzus8w2c11u6?cid=1673](https://piazza.com/class/k4wzus8w2c11u6?cid=1673)
Final Exam

• **How to Prepare**
  – Attend (or watch) this final exam review session
  – Review Practice Problems: Exam 3
    • **Disclaimer:** the practice problems are somewhere between homework-style problems and exam-style problems
  – Review this year’s *homework problems*
  – Review the *poll questions* from each lecture
  – Consider whether you have achieved the *learning objectives* for each lecture / section
Final Exam

• **Advice (for during the exam)**
  – Solve the easy problems first (e.g. multiple choice before derivations)
    • if a problem seems extremely complicated you’re likely missing something
  – Don’t leave any answer blank!
  – If you make an assumption, write it down
  – If you look at a question and don’t know the answer:
    • we probably haven’t told you the answer
    • but we’ve told you enough to work it out
    • imagine arguing for some answer and see if you like it
Topics for Midterm 1

• Foundations
  – Probability, Linear Algebra, Geometry, Calculus
  – Optimization

• Important Concepts
  – Overfitting
  – Experimental Design

• Classification
  – Decision Tree
  – KNN
  – Perceptron

• Regression
  – Linear Regression
Topics for Midterm 2

• **Classification**
  – Binary Logistic Regression
  – Multinomial Logistic Regression

• **Important Concepts**
  – Stochastic Gradient Descent
  – Regularization
  – Feature Engineering

• **Feature Learning**
  – Neural Networks
  – Basic NN Architectures
  – Backpropagation

• **Learning Theory**
  – PAC Learning

• **Generative Models**
  – Generative vs. Discriminative
  – MLE / MAP
  – Naïve Bayes
Topics for Final Exam

• Graphical Models
  – HMMs
  – Learning and Inference
  – Bayesian Networks

• Reinforcement Learning
  – Value Iteration
  – Policy Iteration
  – Q-Learning
  – Deep Q-Learning

• Other Learning Paradigms
  – K-Means
  – PCA
  – SVM (large-margin)
  – Kernels
  – Ensemble Methods
  – Recommender Systems
Classification & Regression

Graphical Models

Learning Paradigms

Reinforcement Learning
Great Race: route and street closing schedule

Street closings
The following city streets will be closed Sunday morning to accommodate the Richard S. Caliguiri City of Pittsburgh Great Race:

Learning as Memorization
Learning as Optimization
Learning from Rewards
Learning and Structure
A new **combined** course...

...with the best (uphill climbs) from both
Material Covered Before Midterm Exam 2

SAMPLE QUESTIONS
Matching Game

**Goal:** Match the Algorithm to its Update Rule

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Update Rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. SGD for Logistic Regression</td>
<td>$\theta_k \leftarrow \theta_k + (h_\theta(x^{(i)}) - y^{(i)})$</td>
</tr>
<tr>
<td>2. Least Mean Squares</td>
<td>$\theta_k \leftarrow \theta_k + \frac{1}{1 + \exp \lambda(h_\theta(x^{(i)}) - y^{(i)})}$</td>
</tr>
<tr>
<td>3. Perceptron (next lecture)</td>
<td>$\theta_k \leftarrow \theta_k + \lambda(h_\theta(x^{(i)}) - y^{(i)})x_k^{(i)}$</td>
</tr>
</tbody>
</table>

A. 1=5, 2=4, 3=6
B. 1=5, 2=6, 3=4
C. 1=6, 2=4, 3=4
D. 1=5, 2=6, 3=6
E. 1=6, 2=6, 3=6
Oh, the Places You’ll Use Probability!

By Dr. Seuss
Sample Questions

1.4 Probability

Assume we have a sample space \( \Omega \). Answer each question with T or F.

(a) [1 pts.] T or F: If events \( A, B, \) and \( C \) are disjoint then they are independent.

(b) [1 pts.] T or F: \( P(A|B) \propto \frac{P(A)P(B|A)}{P(A|B)} \). (The sign ‘\( \propto \)’ means ‘is proportional to’).
Medical Diagnosis

Interview Transcript
Date: Jan. 15, 2020.
Parties: Matt Gormley and Doctor E.
Topic: Medical decision making

Matt: Welcome. Thanks for interviewing with me today.
Dr. E: Interviewing...?
Matt: Yes. For the record, what type of doctor are you?
Dr. E: Who said I'm a doctor?
Matt: I thought when we set up this interview you said—
Dr. E: I'm a preschooler.
Matt: Good enough. Today, I'd like to learn how you would determine whether or not your little brother is sick given his symptoms.
Dr. E: He's not sick.
Matt: We haven't started yet. Now, suppose he is sneezing. Is he sick?
Dr. E: No, that's just the sniffles.
Matt: What if he is coughing; Is he sick?
Dr. E: No, he just has a cough.
Matt: What if he's both sneezing and coughing?
Dr. E: Then he's sick.
Matt: Got it. What if your little brother is sneezing and coughing, plus he's a doctor.
Dr. E: Then he's not sick.
Matt: How do you know?
Dr. E: Doctors don't get sick.
Matt: What if he is not sneezing, but is coughing, and he is a fox....
Dr. E: Then he is must be a tweetle beetle noodle poodle bottled paddled muddled duddled fuddled wuddled fox in socks, sir. That means he's definitely sick.
Matt: Got it. Can I use this conversation in my lecture?
Dr. E: Yes
<table>
<thead>
<tr>
<th>Species</th>
<th>Sepal Length</th>
<th>Sepal Width</th>
<th>Petal Length</th>
<th>Petal Width</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>4.3</td>
<td>3.0</td>
<td>1.1</td>
<td>0.1</td>
</tr>
<tr>
<td>0</td>
<td>4.9</td>
<td>3.6</td>
<td>1.4</td>
<td>0.1</td>
</tr>
<tr>
<td>0</td>
<td>5.3</td>
<td>3.7</td>
<td>1.5</td>
<td>0.2</td>
</tr>
<tr>
<td>1</td>
<td>4.9</td>
<td>2.4</td>
<td>3.3</td>
<td>1.0</td>
</tr>
<tr>
<td>1</td>
<td>5.7</td>
<td>2.8</td>
<td>4.1</td>
<td>1.3</td>
</tr>
<tr>
<td>1</td>
<td>6.3</td>
<td>3.3</td>
<td>4.7</td>
<td>1.6</td>
</tr>
<tr>
<td>1</td>
<td>6.7</td>
<td>3.0</td>
<td>5.0</td>
<td>1.7</td>
</tr>
</tbody>
</table>
4  K-NN [12 pts]

Now we will apply K-Nearest Neighbors using Euclidean distance to a binary classification task. We assign the class of the test point to be the class of the majority of the \( k \) nearest neighbors. A point can be its own neighbor.

3. [2 pts] What value of \( k \) minimizes leave-one-out cross-validation error for the dataset shown in Figure 5? What is the resulting error?
k-NN: Choosing k

Fisher Iris Data: varying the value of k
Q: Why do we need an intercept term?
A: It shifts the decision boundary off the origin.

Q: Why do we add / subtract 1.0 to the intercept term during Perceptron training?
A: Two cases
1. Increasing b shifts the decision boundary towards the negative side.
2. Decreasing b shifts the decision boundary towards the positive side.
**k-NN Regression**

**Example:** Dataset with only one feature $x$ and one scalar output $y$

---

**k=1 Nearest Neighbor Regression**
- **Train:** store all $(x, y)$ pairs
- **Predict:** pick the nearest $x$ in training data and return its $y$

---

**k=2 Nearest Neighbor Distance Weighted Regression**
- **Train:** store all $(x, y)$ pairs
- **Predict:** pick the nearest two instances $x^{(n_1)}$ and $x^{(n_2)}$ in training data and return the weighted average of their $y$ values
Linear Regression by Rand. Guessing

Optimization Method #0: Random Guessing
1. Pick a random $\theta$
2. Evaluate $J(\theta)$
3. Repeat steps 1 and 2 many times
4. Return $\theta$ that gives smallest $J(\theta)$

$h(x; \theta^{(i)})$

$y = h^*(x)$ (unknown)

$J(\theta) = J(\theta_1, \theta_2) = \frac{1}{N} \sum_{i=1}^{N} (y^{(i)} - \theta^T x^{(i)})^2$

<table>
<thead>
<tr>
<th>t</th>
<th>$\theta_1$</th>
<th>$\theta_2$</th>
<th>$J(\theta_1, \theta_2)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.2</td>
<td>0.2</td>
<td>10.4</td>
</tr>
<tr>
<td>2</td>
<td>0.3</td>
<td>0.7</td>
<td>7.2</td>
</tr>
<tr>
<td>3</td>
<td>0.6</td>
<td>0.4</td>
<td>1.0</td>
</tr>
<tr>
<td>4</td>
<td>0.9</td>
<td>0.7</td>
<td>19.2</td>
</tr>
</tbody>
</table>
Sample Questions

3.1 Linear regression

Consider the dataset $S$ plotted in Fig. 1 along with its associated regression line. For each of the altered data sets $S^{\text{new}}$ plotted in Fig. 3, indicate which regression line (relative to the original one) in Fig. 2 corresponds to the regression line for the new data set. Write your answers in the table below.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>(a)</th>
<th>(b)</th>
<th>(c)</th>
<th>(d)</th>
<th>(e)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regression line</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 1: An observed data set and its associated regression line.

(a) Adding one outlier to the original data set.

Figure 2: New regression lines for altered data sets $S^{\text{new}}$.

(b) Adding two outliers to the original data set.

(c) Adding three outliers to the original data set.

(d) Duplicating the original data set.

(e) Duplicating the original data set and adding four points that lie on the trajectory of the original regression line.
Topographical Maps
Linear Regression by Gradient Desc.

\[ J(\theta) = J(\theta_1, \theta_2) = \frac{1}{N} \sum_{i=1}^{N} (y^{(i)} - \theta^T x^{(i)})^2 \]

\[ y = h^*(x) \] (unknown)

<table>
<thead>
<tr>
<th>iteration, ( t )</th>
<th>( \theta_1 )</th>
<th>( \theta_2 )</th>
<th>( J(\theta_1, \theta_2) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.01</td>
<td>0.02</td>
<td>25.2</td>
</tr>
<tr>
<td>2</td>
<td>0.30</td>
<td>0.12</td>
<td>8.7</td>
</tr>
<tr>
<td>3</td>
<td>0.51</td>
<td>0.30</td>
<td>1.5</td>
</tr>
<tr>
<td>4</td>
<td>0.59</td>
<td>0.43</td>
<td>0.2</td>
</tr>
</tbody>
</table>

# tourists (thousands)

\( h(x; \theta^{(1)}) \)
\( h(x; \theta^{(2)}) \)
\( h(x; \theta^{(3)}) \)
\( h(x; \theta^{(4)}) \)
Sample Questions

3.1 Linear regression

Consider the dataset $S$ plotted in Fig. 1 along with its associated regression line. For each of the altered data sets $S^{\text{new}}$ plotted in Fig. 3, indicate which regression line (relative to the original one) in Fig. 2 corresponds to the regression line for the new data set. Write your answers in the table below.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>(a)</th>
<th>(b)</th>
<th>(c)</th>
<th>(d)</th>
<th>(e)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regression line</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 1: An observed data set and its associated regression line.

Figure 2: New regression lines for altered data sets $S^{\text{new}}$.

Dataset

(c) Adding three outliers to the original data set. Two on one side and one on the other side.
3.1 Linear regression

Consider the dataset $S$ plotted in Fig. 1 along with its associated regression line. For each of the altered data sets $S_{\text{new}}$ plotted in Fig. 3, indicate which regression line (relative to the original one) in Fig. 2 corresponds to the regression line for the new data set. Write your answers in the table below.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>(a)</th>
<th>(b)</th>
<th>(c)</th>
<th>(d)</th>
<th>(e)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regression line</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 1: An observed data set and its associated regression line.

Figure 2: New regression lines for altered data sets $S_{\text{new}}$.

(d) Duplicating the original data set.
Sample Questions

3.1 Linear regression

Consider the dataset $S$ plotted in Fig. 1 along with its associated regression line. For each of the altered data sets $S^{\text{new}}$ plotted in Fig. 3, indicate which regression line (relative to the original one) in Fig. 2 corresponds to the regression line for the new data set. Write your answers in the table below.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>(a)</th>
<th>(b)</th>
<th>(c)</th>
<th>(d)</th>
<th>(e)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regression line</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 1: An observed data set and its associated regression line.

Figure 2: New regression lines for altered data sets $S^{\text{new}}$.

Figure 3: New data set.

(a) Old and new regression lines.
(b) Old and new regression lines.
(c) Old and new regression lines.
(d) Adding three outliers to the original data set.
(e) Duplicating the original data set and adding four points that lie on the trajectory of the original regression line.
$J_1(\theta_1, \theta_2)$

$J(\theta_1, \theta_2)$

$J_2(\theta_1, \theta_2)$

$f(\theta_1, \theta_2)$
Robotic Farming

<table>
<thead>
<tr>
<th></th>
<th>Deterministic</th>
<th>Probabilistic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classification</td>
<td>Is this a picture of a wheat kernel?</td>
<td>Is this plant drought resistant?</td>
</tr>
<tr>
<td>(binary output)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Regression</td>
<td>How many wheat kernels are in this picture?</td>
<td>What will the yield of this plant be?</td>
</tr>
<tr>
<td>(continuous output)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Multinomial Logistic Regression

polar bears
sea lions
sharks
Sample Questions

3.2 Logistic regression

Given a training set \( \{ (x_i, y_i), i = 1, \ldots, n \} \) where \( x_i \in \mathbb{R}^d \) is a feature vector and \( y_i \in \{0, 1\} \) is a binary label, we want to find the parameters \( \hat{w} \) that maximize the likelihood for the training set, assuming a parametric model of the form

\[
p(y = 1|x; w) = \frac{1}{1 + \exp(-w^T x)}.
\]

The conditional log likelihood of the training set is

\[
\ell(w) = \sum_{i=1}^{n} y_i \log p(y_i; |x_i; w) + (1 - y_i) \log(1 - p(y_i; |x_i; w)),
\]

and the gradient is

\[
\nabla \ell(w) = \sum_{i=1}^{n} (y_i - p(y_i|x_i; w)) x_i.
\]

(b) [5 pts.] What is the form of the classifier output by logistic regression?

(c) [2 pts.] Extra Credit: Consider the case with binary features, i.e., \( x \in \{0, 1\}^d \subset \mathbb{R}^d \), where feature \( x_1 \) is rare and happens to appear in the training set with only label 1. What is \( \hat{w}_1 \)? Is the gradient ever zero for any finite \( w \)? Why is it important to include a regularization term to control the norm of \( \hat{w} \)?
Handcrafted Features

\[ p(y|x) \propto \exp(\Theta_y \cdot f) \]
**Example: Linear Regression**

**Goal:** Learn $y = w^T f(x) + b$ where $f(.)$ is a polynomial basis function.

The true “unknown” target function is linear with negative slope and Gaussian noise.
Question:
Suppose we are minimizing $J'(\theta)$ where

$$J'(\theta) = J(\theta) + \lambda r(\theta)$$

As $\lambda$ increases, the minimum of $J'(\theta)$ will...

A. ...move towards the midpoint between $J'(\theta)$ and $r(\theta)$
B. ...move towards the minimum of $J(\theta)$
C. ...move towards the minimum of $r(\theta)$
D. ...move towards a theta vector of positive infinities
E. ...move towards a theta vector of negative infinities
F. ...stay the same
2.1 Train and test errors

In this problem, we will see how you can debug a classifier by looking at its train and test errors. Consider a classifier trained till convergence on some training data $D^{\text{train}}$, and tested on a separate test set $D^{\text{test}}$. You look at the test error, and find that it is very high. You then compute the training error and find that it is close to 0.

1. [4 pts] Which of the following is expected to help? Select all that apply.

   (a) Increase the training data size.
   (b) Decrease the training data size.
   (c) Increase model complexity (For example, if your classifier is an SVM, use a more complex kernel. Or if it is a decision tree, increase the depth).
   (d) Decrease model complexity.
   (e) Train on a combination of $D^{\text{train}}$ and $D^{\text{test}}$ and test on $D^{\text{test}}$
   (f) Conclude that Machine Learning does not work.

2. [5 pts] Explain your choices.

3. [2 pts] What is this scenario called?

4. [1 pts] Say you plot the train and test errors as a function of the model complexity. Which of the following two plots is your plot expected to look like?
2.1 Train and test errors

In this problem, we will see how you can debug a classifier by looking at its train and test errors. Consider a classifier trained till convergence on some training data $D_{\text{train}}$, and tested on a separate test set $D_{\text{test}}$. You look at the test error, and find that it is very high. You then compute the training error and find that it is close to 0.

4. [1 pts] Say you plot the train and test errors as a function of the model complexity. Which of the following two plots is your plot expected to look like?
Sample Questions

4.1 True or False

Answer each of the following questions with T or F and provide a one line justification.

(a) [2 pts.] Consider two datasets $D^{(1)}$ and $D^{(2)}$ where $D^{(1)} = \{(x_1^{(1)}, y_1^{(1)}), \ldots, (x_n^{(1)}, y_n^{(1)})\}$ and $D^{(2)} = \{(x_1^{(2)}, y_1^{(2)}), \ldots, (x_m^{(2)}, y_m^{(2)})\}$ such that $x_i^{(1)} \in \mathbb{R}^{d_1}$, $x_i^{(2)} \in \mathbb{R}^{d_2}$. Suppose $d_1 > d_2$ and $n > m$. Then the maximum number of mistakes a perceptron algorithm will make is higher on dataset $D^{(1)}$ than on dataset $D^{(2)}$. 

(b) [2 pts.] Suppose $(x)$ is an arbitrary feature mapping from input $x \in \mathcal{X}$ to $(x) \in \mathbb{R}^N$ and let $K(x, z) = (x) \cdot (z)$. Then $K(x, z)$ will always be a kernel function.
Logistic Regression

\[ y = h_\theta(x) = \sigma(\theta^T x) \]

where \( \sigma \) is the sigmoid function.

Input

\[ \begin{align*}
\theta_1 & \quad x_1 \\
\theta_2 & \quad x_2 \\
\theta_3 & \quad x_3
\end{align*} \]

Output

Decision Functions

In-Class Example

\[ \begin{pmatrix}
1 \\
1 \\
0
\end{pmatrix} \]
Extra Credit: Neural Networks [6 pts.]

In this problem we will use a neural network to classify the crosses (×) from the circles (○) in the simple dataset shown in Figure 5a. Even though the crosses and circles are not linearly separable, we can break the examples into three groups, \( S_1, S_2, \) and \( S_3 \) (shown in Figure 5a) so that \( S_1 \) is linearly separable from \( S_2 \) and \( S_2 \) is linearly separable from \( S_3 \). We will exploit this fact to design weights for the neural network shown in Figure 5b in order to correctly classify this training set. For all nodes, we will use the threshold activation function \( \varphi(z) = \begin{cases} 1 & z > 0 \\ 0 & z \leq 0 \end{cases} \).

(a) The dataset with groups \( S_1, S_2, \) and \( S_3 \).

(b) The neural network architecture

Figure 5: NN classification.

Can the neural network in Figure (b) correctly classify the dataset given in Figure (a)?
Multi-Class Output

Softmax:

\[ y_k = \frac{\exp(b_k)}{\sum_{l=1}^{K} \exp(b_l)} \]

\( (A) \) Input
Given \( x_i, \forall i \)

\( (B) \) Hidden (linear)
\[ a_j = \sum_{i=0}^{M} \alpha_{ji} x_i, \forall j \]

\( (C) \) Hidden (nonlinear)
\[ z_j = o(a_j), \forall j \]

\( (D) \) Output (linear)
\[ b_k = \sum_{j=0}^{D} \beta_{kj} z_j, \forall k \]

\( (E) \) Output (softmax)
\[ y_k = \frac{\exp(b_k)}{\sum_{l=1}^{K} \exp(b_l)} \]

\( (F) \) Loss
\[ J = \sum_{k=1}^{K} y_k^* \log(y_k) \]
Error Back-Propagation

$p(y|x^{(i)})$

$y^{(i)}$

$z$

$\theta$

Slide from (Stoyanov & Eisner, 2012)
Neural Networks

Apply the backpropagation algorithm to obtain the partial derivative of the mean-squared error of $y$ with the true value $y^*$ with respect to the weight $w_{22}$ assuming a sigmoid nonlinear activation function for the hidden layer.
Architecture #2: AlexNet

**CNN for Image Classification**
(Krizhevsky, Sutskever & Hinton, 2012)
15.3% error on ImageNet LSVRC-2012 contest

- Input image (pixels)
- 1000-way softmax
- Five convolutional layers (w/max-pooling)
- Three fully connected layers

**Figure 2:** An illustration of the architecture of our CNN, explicitly showing the delineation of responsibilities between the two GPUs. One GPU runs the layer-parts at the top of the figure while the other runs the layer-parts at the bottom. The GPUs communicate only at certain layers. The network's input is 150,528-dimensional, and the number of neurons in the network's remaining layers is given by 253,440–186,624–64,896–64,896–43,264–4096–4096–1000.

Reducing Overfitting

Our neural network architecture has 60 million parameters. Although the 1000 classes of ILSVRC make each training example impose 10 bits of constraint on the mapping from image to label, this turns out to be insufficient to learn so many parameters without considerable overfitting. Below, we describe the two primary ways in which we combat overfitting.

**4.1 Data Augmentation**

The easiest and most common method to reduce overfitting on image data is to artificially enlarge the dataset using label-preserving transformations (e.g., [25, 4, 5]). We employ two distinct forms of data augmentation, both of which allow transformed images to be produced from the original images with very little computation, so the transformed images do not need to be stored on disk. In our implementation, the transformed images are generated in Python code on the CPU while the GPU is training on the previous batch of images. So these data augmentation schemes are, in effect, computationally free.

The first form of data augmentation consists of generating image translations and horizontal reflections. We do this by extracting random $224 \times 224$ patches (and their horizontal reflections) from the $256 \times 256$ images and training our network on these extracted patches. This increases the size of our training set by a factor of 2048, though the resulting training examples are, of course, highly interdependent. Without this scheme, our network suffers from substantial overfitting, which would have forced us to use much smaller networks. At test time, the network makes a prediction by extracting five $224 \times 224$ patches (the four corner patches and the center patch) as well as their horizontal reflections (hence ten patches in all), and averaging the predictions made by the network's softmax layer on the ten patches.

The second form of data augmentation consists of altering the intensities of the RGB channels in training images. Specifically, we perform PCA on the set of RGB pixel values throughout the ImageNet training set. To each training image, we add multiples of the found principal components.
Bidirectional RNN

inputs: \( x = (x_1, x_2, \ldots, x_T), x_i \in \mathcal{R}^I \)

hidden units: \( \mathbf{h} \) and \( \mathbf{\overline{h}} \)

outputs: \( y = (y_1, y_2, \ldots, y_T), y_i \in \mathcal{R}^K \)

nonlinearity: \( \mathcal{H} \)

Recursive Definition:
\[
\begin{align*}
\mathbf{\overrightarrow{h}}_t &= \mathcal{H} \left( W_{x \mathbf{h}} x_t + W_{\mathbf{\overrightarrow{h}} \mathbf{h}} \mathbf{\overrightarrow{h}}_{t-1} + b_{\mathbf{h}} \right) \\
\mathbf{\overleftarrow{h}}_t &= \mathcal{H} \left( W_{x \mathbf{\overleftarrow{h}}} x_t + W_{\mathbf{\overleftarrow{h}} \mathbf{h}} \mathbf{\overleftarrow{h}}_{t+1} + b_{\mathbf{h}} \right) \\
y_t &= W_{\mathbf{h} y} \mathbf{\overrightarrow{h}}_t + W_{\mathbf{\overleftarrow{h}} y} \mathbf{\overleftarrow{h}}_t + b_y 
\end{align*}
\]
Question 2:
What is the expected number of PAC-MAN levels Matt will complete before a Game-Over?

A. 1-10
B. 11-20
C. 21-30
2.1 True Errors

(b) [4 pts.] T or F: Learning theory allows us to determine with 100% certainty the true error of a hypothesis to within any $\epsilon > 0$ error.
2.2 Training Sample Size

In this problem, we will consider the effect of training sample size $n$ on a logistic regression classifier with $d$ features. The classifier is trained by optimizing the conditional log-likelihood. The optimization procedure stops if the estimated parameters perfectly classify the training data or they converge. The following plot shows the general trend for how the training and testing error change as we increase the sample size $n = |S|$. Your task in this question is to analyze this plot and identify which curve corresponds to the training and test error. Specifically:

(a) [8 pts.] Which curve represents the training error? Please provide 1–2 sentences of justification.

(b) [4 pt.] In one word, what does the gap between the two curves represent?
Sample Questions

5 Learning Theory [20 pts.]

(a) [3 pts.] T or F: It is possible to label 4 points in $\mathbb{R}^2$ in all possible $2^4$ ways via linear separators in $\mathbb{R}^2$.

(d) [3 pts.] T or F: The VC dimension of a concept class with infinite size is also infinite.

(f) [3 pts.] T or F: Given a realizable concept class and a set of training instances, a consistent learner will output a concept that achieves 0 error on the training instances.
PAC Learning & Regularization

Model Selection

Q. Is Corr. 4 useful? A: Yes!

Key Idea: tradeoff between low training error and keeping $H$ simple (low VCDim).

$\hat{R}(h) + \sqrt{\frac{1}{2N}[VC(H) + \ln(1/\delta)]}$

$R(h)$ true error

$\left(\sqrt{\frac{1}{2N}[VC(H) + \ln(1/\delta)]}\right)$

Ex: Lin. Sep. in $\mathbb{R}^m$

$VC(H) = M + 1$

How to tradeoff?

(use a regularizer)

$r(\theta) = \frac{M}{N} |\theta_m |$

$\theta = \arg \min_{\theta} J(\theta) + r(\theta)$
MLE vs. MAP

Suppose we have data $\mathcal{D} = \{x^{(i)}\}_{i=1}^N$

**Principle of Maximum Likelihood Estimation:**
Choose the parameters that maximize the likelihood of the data.

$$\theta^{\text{MLE}} = \arg\max_{\theta} \prod_{i=1}^N p(x^{(i)}|\theta)$$

**Maximum Likelihood Estimate (MLE)**

**Principle of Maximum a posteriori (MAP) Estimation:**
Choose the parameters that maximize the posterior of the parameters given the data.

$$\theta^{\text{MAP}} = \arg\max_{\theta} \prod_{i=1}^N p(x^{(i)}|\theta)p(\theta)$$

**Maximum a posteriori (MAP) estimate**
1.2 Maximum Likelihood Estimation (MLE)

Assume we have a random sample that is Bernoulli distributed $X_1, \ldots, X_n \sim \text{Bernoulli}(\theta)$. We are going to derive the MLE for $\theta$. Recall that a Bernoulli random variable $X$ takes values in $\{0, 1\}$ and has probability mass function given by

$$P(X; \theta) = \theta^X (1 - \theta)^{1-X}.$$ 

(a) [2 pts.] Derive the likelihood, $L(\theta; X_1, \ldots, X_n)$.

(c) Extra Credit: [2 pts.] Derive the following formula for the MLE: $\hat{\theta} = \frac{1}{n} (\sum_{i=1}^n X_i)$. 
Sample Questions

1.3 MAP vs MLE

Answer each question with T or F and provide a one sentence explanation of your answer:

(a) [2 pts.] T or F: In the limit, as $n$ (the number of samples) increases, the MAP and MLE estimates become the same.
Fake News Detector

Today’s Goal: To define a generative model of emails of two different classes (e.g. real vs. fake news)

The Economist

The Onion
Model 1: Bernoulli Naïve Bayes

If HEADS, flip each red coin

Flip weighted coin

If TAILS, flip each blue coin

We can generate data in this fashion. Though in practice we never would since our data is given.

Instead, this provides an explanation of how the data was generated (albeit a terrible one).

<table>
<thead>
<tr>
<th>y</th>
<th>x_1</th>
<th>x_2</th>
<th>x_3</th>
<th>…</th>
<th>x_M</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>…</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>…</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>…</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>…</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>…</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>…</td>
<td>0</td>
</tr>
</tbody>
</table>

Each red coin corresponds to an x_m
1.1 Naive Bayes

You are given a data set of 10,000 students with their sex, height, and hair color. You are trying to build a classifier to predict the sex of a student, so you randomly split the data into a training set and a testing set. Here are the specifications of the data set:

- sex ∈ {male,female}
- height ∈ [0,300] centimeters
- hair ∈ {brown, black, blond, red, green}
- 3240 men in the data set
- 6760 women in the data set

Under the assumptions necessary for Naive Bayes (not the distributional assumptions you might naturally or intuitively make about the dataset) answer each question with T or F and provide a one sentence explanation of your answer:

(a) [2 pts.] T or F: As height is a continuous valued variable, Naive Bayes is not appropriate since it cannot handle continuous valued variables.

(c) [2 pts.] T or F: \( P(\text{height}|\text{sex, hair}) = P(\text{height}|\text{sex}) \).
Material Covered After Midterm Exam 2

SAMPLE QUESTIONS
Totoro’s Tunnel
Great Ideas in ML: Message Passing

Count the soldiers

2 before you

there's 1 of me

only see my incoming messages

3 behind you

Belief: Must be 2 + 1 + 3 = 6 of us

adapted from MacKay (2003) textbook
Forward-Backward Algorithm: Finds Marginals

\[ \alpha_2(n) = \text{total weight of these path prefixes (} a + b + c \text{)} \]
\[ \beta_2(n) = \text{total weight of these path suffixes (} x + y + z \text{)} \]

Product gives \( ax + ay + az + bx + by + bz + cx + cy + cz \) = total weight of paths
Sample Questions

4 Hidden Markov Models

1. Given the POS tagging data shown, what are the parameter values learned by an HMM?

\[
A = \begin{pmatrix}
\text{Verb} & \text{Noun} & \text{Adj} \\
\text{see} & 1/4 & 0/4 & 3/4 \\
\text{spot} & 0/4 & 0/4 \\
\text{funny} & & & \\
\end{pmatrix}
\]

\[p(x_t | y_t)\]

\[
B = \begin{pmatrix}
\text{Verb} & \text{Noun} & \text{Adj} \\
\text{see} & \text{spot} & \text{run} \\
\text{run} & \text{spot} & \text{run} \\
\text{funny} & \text{funny} & \text{spot} \\
\end{pmatrix}
\]
Sample Questions

4 Hidden Markov Models

1. Given the POS tagging data shown, what are the parameter values learned by an HMM?

2. Suppose you are learning an HMM POS Tagger, how many POS tag sequences of length 23 are there?

3. How does an HMM efficiently search for the most probable tag sequence given a 23 word sentence?
Example: Ryan Reynolds’ Voicemail

From https://www.adweek.com/brand-marketing/ryan-reynolds-left-voicemails-for-all-mint-mobile-subscribers/
Example: Tornado Alarms

1. Imagine that you work at the 911 call center in Dallas
2. You receive six calls informing you that the Emergency Weather Sirens are going off
3. What do you conclude?

Figure from https://www.nytimes.com/2017/04/08/us/dallas-emergency-sirens-hacking.html
Sample Questions

(a) [2 pts.] Write the expression for the joint distribution.

\[ p(S, R, E, A) = p(A | E) p(E | S, R) p(S) p(R) \]

5 Graphical Models [16 pts.]

We use the following Bayesian network to model the relationship between studying (S), being well-rested (R), doing well on the exam (E), and getting an A grade (A). All nodes are binary, i.e., \( R, S, E, A \in \{0, 1\} \).

![Directed graphical model for problem 5.](image)

Figure 5: Directed graphical model for problem 5.
(b) [2 pts.] How many parameters, i.e., entries in the CPT tables, are necessary to describe the joint distribution?

5 Graphical Models [16 pts.]

We use the following Bayesian network to model the relationship between studying (S), being well-rested (R), doing well on the exam (E), and getting an A grade (A). All nodes are binary, i.e., $R, S, E, A \in \{0, 1\}$.

![Figure 5: Directed graphical model for problem 5.](image-url)
(d) [2 pts.] Is $S$ marginally independent of $R$? Is $S$ conditionally independent of $R$ given $E$? Answer yes or no to each question and provide a brief explanation why.

5 Graphical Models [16 pts.]

We use the following Bayesian network to model the relationship between studying ($S$), being well-rested ($R$), doing well on the exam ($E$), and getting an A grade ($A$). All nodes are binary, i.e., $R, S, E, A \in \{0, 1\}$.

Figure 5: Directed graphical model for problem 5.
Sample Questions

5 Graphical Models

(f) [3 pts.] Give two reasons why the graphical models formalism is convenient when compared to learning a full joint distribution.
Gibbs Sampling

\[ p(x) \]

\[ p(x_2 | x_1^{(t+1)}) \]

\[ x^{(t)} \]

\[ x^{(t+1)} \]

\[ x^{(t+2)} \]
Example: Path Planning
Today’s lecture is brought you by the letter....
Playing Atari with Deep RL

- Setup: RL system observes the pixels on the screen.
- It receives rewards as the game score.
- Actions decide how to move the joystick / buttons.

Figures from David Silver (Intro RL lecture)
not-so-Deep Q-Learning

Approx w/NN:

\[ Q(s,a;\theta) \]

\[ \begin{array}{c}
\uparrow \\
\scriptstyle{s} \\
\uparrow \\
\scriptstyle{a} \\
\end{array} \]

\[ Q(s,a;\theta) \]

\[ \begin{array}{c}
\uparrow \\
\scriptstyle{s} \\
\uparrow \\
\end{array} \]

\[ IA| = K = \# \text{ actions} \]

Approx w/Linear Regression:

Represent state as a feature vector:

\[ \tilde{s}_t = [0,1,0,0,\ldots,1]^T \]

\[ |\tilde{s}_t| \in \mathbb{R}^M \]

\[ M = \# \text{ features} \]

\[ Q(\tilde{s},a;\theta) = \tilde{\theta}_a^T \tilde{s} \]

\[ \Theta = \begin{bmatrix}
\tilde{\theta}_1^T \\
\tilde{\theta}_2^T \\
\vdots \\
\tilde{\theta}_K^T
\end{bmatrix} \theta \in \mathbb{R}^{K \times M} \]

\[ \nabla_\theta Q(\tilde{s},a;\theta) = \begin{cases}
\tilde{s} & \text{if } a = b \\
0 & \text{if } a \neq b
\end{cases} \]

\[ \begin{bmatrix}
0 \\
\vdots \\
\tilde{s}_T \\
\vdots \\
0
\end{bmatrix} \text{ row} \]

\[ \nabla Q(\tilde{s},a;\theta) = \begin{bmatrix}
\tilde{s} & 0 & \ldots & 0 \\
0 & \ddots & \ddots & \vdots \\
\vdots & \ddots & \ddots & \ddots \\
0 & \ldots & 0 & \tilde{s}
\end{bmatrix} \text{ row} \]
7.1 Reinforcement Learning

3. (1 point) **Please select one statement that is true for reinforcement learning and supervised learning.**

- Reinforcement learning is a kind of supervised learning problem because you can treat the reward and next state as the label and each state, action pair as the training data. **B**

- Reinforcement learning differs from supervised learning because it has a temporal structure in the learning process, whereas, in supervised learning, the prediction of a data point does not affect the data you would see in the future. **C ✓**

4. (1 point) **True or False:** Value iteration is better at balancing exploration and exploitation compared with policy iteration.

- True
- **False ✓**
7.1 Reinforcement Learning

1. For the R(s,a) values shown on the arrows below, what is the corresponding optimal policy? Assume the discount factor is 0.1

2. For the R(s,a) values shown on the arrows below, which are the corresponding V*(s) values? Assume the discount factor is 0.1

3. For the R(s,a) values shown on the arrows below, which are the corresponding Q*(s,a) values? Assume the discount factor is 0.1
Example: Robot Localization

![Figure from Tom Mitchell](image_url)
K-Means Example:
A Real-World Dataset
Example: K-Means
Example: K-Means
2. K-Means Clustering

(a) [3 pts] We are given $n$ data points, $x_1, ..., x_n$ and asked to cluster them using K-means. If we choose the value for $k$ to optimize the objective function how many clusters will be used (i.e. what value of $k$ will we choose)? No justification required.

(i) 1   (ii) 2   (iii) $n$   (iv) $\log(n)$

A  B  C  D

60%  40%  

E =灿常	
2.2 Lloyd’s algorithm

Circle the image which depicts the cluster center positions after 1 iteration of Lloyd’s algorithm.

Figure 2: Initial data and cluster centers
2.2 Lloyd’s algorithm

Circle the image which depicts the cluster center positions after 1 iteration of Lloyd’s algorithm.

Figure 2: Initial data and cluster centers
High Dimension Data

Examples of high dimensional data:
– Brain Imaging Data (100s of MBs per scan)

Image from (Wehbe et al., 2014)
Shortcut Example

https://www.youtube.com/watch?v=MlJN9pEfPfE
Projecting MNIST digits

Task Setting:
1. Take 25x25 images of digits and project them down to 2 components
2. Plot the 2 dimensional points
Sample Questions

4 Principal Component Analysis [16 pts.]

(a) In the following plots, a train set of data points $X$ belonging to two classes on $\mathbb{R}^2$ are given, where the original features are the coordinates $(x, y)$. For each, answer the following questions:

(i) [3 pt.] Draw all the principal components.

(ii) [6 pts.] Can we correctly classify this dataset by using a threshold function after projecting onto one of the principal components? If so, which principal component should we project onto? If not, explain in 1–2 sentences why it is not possible.

Dataset 1:

Dataset 2:
4 Principal Component Analysis

(c) [2 pts.] Assume we apply PCA to a matrix $X \in \mathbb{R}^{n \times m}$ and obtain a set of PCA features, $Z \in \mathbb{R}^{m \times n}$. We divide this set into two, $Z_1$ and $Z_2$. The first set, $Z_1$, corresponds to the top principal components. The second set, $Z_2$, corresponds to the remaining principal components. Which is more common in the training data:

A: a point with large feature values in $Z_1$ and small feature values in $Z_2$

B: a point with large feature values in $Z_2$ and small feature values in $Z_1$
4 Principal Component Analysis

(i) **T or F** The goal of PCA is to interpret the underlying structure of the data in terms of the principal components that are best at predicting the output variable.

(ii) **T or F** The output of PCA is a new representation of the data that is always of lower dimensionality than the original feature representation.

(iii) **T or F** Subsequent principal components are always orthogonal to each other.
SVM Example: Building Walls

https://www.facebook.com/Mondobloxx/
SVM QP

SVM Quadratic Program

Classification with SVM ($w=[1.28, 1.60]$)
Soft-Margin SVM

Hard-margin SVM (Primal)
\[
\min_{\mathbf{w}, b} \frac{1}{2} \| \mathbf{w} \|^2_2 \\
\text{s.t. } y^{(i)} (\mathbf{w}^T \mathbf{x}^{(i)} + b) \geq 1, \quad \forall i = 1, \ldots, N
\]

Hard-margin SVM (Lagrangian Dual)
\[
\max_\alpha \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y^{(i)} y^{(j)} \mathbf{x}^{(i)} \cdot \mathbf{x}^{(j)} \\
\text{s.t. } \alpha_i \geq 0, \quad \forall i = 1, \ldots, N \\
\sum_{i=1}^N \alpha_i y^{(i)} = 0
\]

Soft-margin SVM (Primal)
\[
\min_{\mathbf{w}, b} \frac{1}{2} \| \mathbf{w} \|^2_2 + C \left( \sum_{i=1}^N e_i \right) \\
\text{s.t. } y^{(i)} (\mathbf{w}^T \mathbf{x}^{(i)} + b) \geq 1 - e_i, \quad \forall i = 1, \ldots, N \\
e_i \geq 0, \quad \forall i = 1, \ldots, N
\]

Soft-margin SVM (Lagrangian Dual)
\[
\max_\alpha \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y^{(i)} y^{(j)} \mathbf{x}^{(i)} \cdot \mathbf{x}^{(j)} \\
\text{s.t. } 0 \leq \alpha_i \leq C, \quad \forall i = 1, \ldots, N \\
\sum_{i=1}^N \alpha_i y^{(i)} = 0
\]
Sample Questions

(c) [4 pts.] **Extra Credit:** Consider the dataset in Fig. 4. Under the SVM formulation in section 4.2(a),

(1) Draw the decision boundary on the graph.
(2) What is the size of the margin?
(3) Circle all the support vectors on the graph.

Figure 4: SVM toy dataset
Sample Questions

4.2 Multiple Choice

(a) [3 pt.] If the data is linearly separable, SVM minimizes $\|w\|^2$ subject to the constraints $\forall i, y_i w \cdot x_i \geq 1$. In the linearly separable case, which of the following may happen to the decision boundary if one of the training samples is removed? **Circle all that apply.**

- Shifts toward the point removed
- Shifts away from the point removed
- Does not change
3. [Extra Credit: 3 pts.] One formulation of soft-margin SVM optimization problem is:

\[
\begin{align*}
\min_w & \quad \frac{1}{2} \|w\|_2^2 + C \sum_{i=1}^{N} \xi_i \\
\text{s.t.} & \quad y_i (w^\top x_i) \geq 1 - \xi_i \quad \forall i = 1, \ldots, N \\
& \quad \xi_i \geq 0 \quad \forall i = 1, \ldots, N \\
& \quad C \geq 0
\end{align*}
\]

where \((x_i, y_i)\) are training samples and \(w\) defines a linear decision boundary.

Derive a formula for \(\xi_i\) when the objective function achieves its minimum (No steps necessary). Note it is a function of \(y_i w^\top x_i\). Sketch a plot of \(\xi_i\) with \(y_i w^\top x_i\) on the x-axis and value of \(\xi_i\) on the y-axis. What is the name of this function?
RBF Kernel Example

KNN vs. SVM

RBF Kernel: \[ K(x^{(i)}, x^{(j)}) = \exp(-\gamma \|x^{(i)} - x^{(j)}\|^2) \]
Sample Questions

4.3 Analysis

(a) [4 pts.] In one or two sentences, describe the benefit of using the Kernel trick.

(b) [4 pt.] The concept of margin is essential in both SVM and Perceptron. Describe why a large margin separator is desirable for classification.

(e) [2 pts.] T or F: The function $K(x, z) = -2x^Tz$ is a valid kernel function.
Recommender Systems

![Netflix Prize Leaderboard](image_url)

<table>
<thead>
<tr>
<th>Rank</th>
<th>Team Name</th>
<th>Best Test Score</th>
<th>% Improvement</th>
<th>Best Submit Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>BellKor's Pragmatic Chaos</td>
<td>0.8567</td>
<td>10.06</td>
<td>2009-07-26 18:18:28</td>
</tr>
<tr>
<td>2</td>
<td>The Ensemble</td>
<td>0.8567</td>
<td>10.06</td>
<td>2009-07-26 18:38:22</td>
</tr>
<tr>
<td>3</td>
<td>Grand Prize Team</td>
<td>0.8568</td>
<td>9.90</td>
<td>2009-07-10 21:24:40</td>
</tr>
<tr>
<td>4</td>
<td>Opera Solutions and Vandelay United</td>
<td>0.8594</td>
<td>9.77</td>
<td>2009-06-24 12:06:56</td>
</tr>
<tr>
<td>5</td>
<td>Vandelay Industries I</td>
<td>0.8601</td>
<td>9.70</td>
<td>2009-05-13 08:14:09</td>
</tr>
<tr>
<td>6</td>
<td>PragmaticTheory</td>
<td>0.8622</td>
<td>9.48</td>
<td>2009-07-12 13:11:51</td>
</tr>
<tr>
<td>7</td>
<td>BellKor in BigChaos</td>
<td>0.8623</td>
<td>9.47</td>
<td>2009-04-07 12:33:59</td>
</tr>
<tr>
<td>8</td>
<td>BigChaos</td>
<td>0.8623</td>
<td>9.47</td>
<td>2009-07-24 00:34:07</td>
</tr>
<tr>
<td>9</td>
<td>Opera Solutions</td>
<td>0.8623</td>
<td>9.46</td>
<td>2009-07-26 17:19:11</td>
</tr>
</tbody>
</table>
Weighted Majority Algorithm

(Littlestone & Warmuth, 1994)

- **Given:** pool $A$ of binary classifiers (that you know nothing about)
- **Data:** stream of examples (i.e. online learning setting)
- **Goal:** design a new learner that uses the predictions of the pool to make new predictions
- **Algorithm:**
  - Initially weight all classifiers equally
  - Receive a training example and predict the (weighted) majority vote of the classifiers in the pool
  - Down-weight classifiers that contribute to a mistake by a factor of $\beta$
AdaBoost: Toy Example

\[
H_{\text{final}} = \text{sign} \left( 0.42 + 0.65 + 0.92 \right)
\]
Two Types of Collaborative Filtering

2. Latent Factor Methods

- Assume that both movies and users live in some low-dimensional space describing their properties.
- **Recommend** a movie based on its **proximity** to the user in the latent space.
- **Example Algorithm:** Matrix Factorization.

Figures from Koren et al. (2009)
Crowdsourcing Exam Questions

In-Class Exercise

1. Select one of lecture-level learning objectives
2. Write a question that assesses that objective
3. Adjust to avoid ‘trivia style’ question

Answer Here:
The Big Picture

MACHINE LEARNING
## Learning Paradigms

<table>
<thead>
<tr>
<th>Paradigm</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supervised</td>
<td>$D = {x^{(i)}, y^{(i)}}_{i=1}^{N}$, $x \sim p^<em>(\cdot)$ and $y = c^</em>(\cdot)$</td>
</tr>
<tr>
<td>$\rightarrow$ Regression</td>
<td>$y^{(i)} \in \mathbb{R}$</td>
</tr>
<tr>
<td>$\rightarrow$ Classification</td>
<td>$y^{(i)} \in {1, \ldots, K}$</td>
</tr>
<tr>
<td>$\rightarrow$ Binary classification</td>
<td>$y^{(i)} \in {+1, -1}$</td>
</tr>
<tr>
<td>$\rightarrow$ Structured Prediction</td>
<td>$y^{(i)}$ is a vector</td>
</tr>
</tbody>
</table>
## Learning Paradigms

<table>
<thead>
<tr>
<th>Paradigm</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supervised</td>
<td>$\mathcal{D} = {x^{(i)}, y^{(i)}}_{i=1}^N$  $x \sim p^<em>(\cdot)$ and $y = c^</em>(\cdot)$</td>
</tr>
<tr>
<td>$\leftrightarrow$ Regression</td>
<td>$y^{(i)} \in \mathbb{R}$</td>
</tr>
<tr>
<td>$\leftrightarrow$ Classification</td>
<td>$y^{(i)} \in {1, \ldots, K}$</td>
</tr>
<tr>
<td>$\leftrightarrow$ Binary classification</td>
<td>$y^{(i)} \in {+1, -1}$</td>
</tr>
<tr>
<td>$\leftrightarrow$ Structured Prediction</td>
<td>$y^{(i)}$ is a vector</td>
</tr>
<tr>
<td>Unsupervised</td>
<td>$\mathcal{D} = {x^{(i)}}_{i=1}^N$  $x \sim p^*(\cdot)$</td>
</tr>
</tbody>
</table>
# Learning Paradigms

<table>
<thead>
<tr>
<th>Paradigm</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supervised</td>
<td>$D = {x^{(i)}, y^{(i)}}_{i=1}^N \quad x \sim p^<em>(\cdot) \text{ and } y = c^</em>(\cdot)$</td>
</tr>
<tr>
<td>$\rightarrow$ Regression</td>
<td>$y^{(i)} \in \mathbb{R}$</td>
</tr>
<tr>
<td>$\rightarrow$ Classification</td>
<td>$y^{(i)} \in {1, \ldots, K}$</td>
</tr>
<tr>
<td>$\rightarrow$ Binary classification</td>
<td>$y^{(i)} \in {+1, -1}$</td>
</tr>
<tr>
<td>$\rightarrow$ Structured Prediction</td>
<td>$y^{(i)}$ is a vector</td>
</tr>
<tr>
<td>Unsupervised</td>
<td>$D = {x^{(i)}}_{i=1}^N \quad x \sim p^*(\cdot)$</td>
</tr>
<tr>
<td>Semi-supervised</td>
<td>$D = {x^{(i)}, y^{(i)}}<em>{i=1}^{N_1} \cup {x^{(j)}}</em>{j=1}^{N_2}$</td>
</tr>
</tbody>
</table>

123
## Learning Paradigms

<table>
<thead>
<tr>
<th>Paradigm</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supervised</td>
<td>$\mathcal{D} = {x^{(i)}, y^{(i)}}_{i=1}^N$ $x \sim p^<em>(\cdot)$ and $y = c^</em>(\cdot)$</td>
</tr>
<tr>
<td>Regression</td>
<td>$y^{(i)} \in \mathbb{R}$</td>
</tr>
<tr>
<td>Classification</td>
<td>$y^{(i)} \in {1, \ldots, K}$</td>
</tr>
<tr>
<td>Binary classification</td>
<td>$y^{(i)} \in {+1, -1}$</td>
</tr>
<tr>
<td>Structured Prediction</td>
<td>$y^{(i)}$ is a vector</td>
</tr>
<tr>
<td>Unsupervised</td>
<td>$\mathcal{D} = {x^{(i)}}_{i=1}^N$ $x \sim p^*(\cdot)$</td>
</tr>
<tr>
<td>Semi-supervised</td>
<td>$\mathcal{D} = {x^{(i)}, y^{(i)}}<em>{i=1}^{N_1} \cup {x^{(j)}}</em>{j=1}^{N_2}$</td>
</tr>
<tr>
<td>Online</td>
<td>$\mathcal{D} = {(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), (x^{(3)}, y^{(3)}), \ldots}$</td>
</tr>
</tbody>
</table>
# Learning Paradigms

<table>
<thead>
<tr>
<th>Paradigm</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supervised</td>
<td>$\mathcal{D} = {x^{(i)}, y^{(i)}}_{i=1}^{N}$ \quad $x \sim p^<em>(\cdot)$ and $y = c^</em>(\cdot)$</td>
</tr>
<tr>
<td>→ Regression</td>
<td>$y^{(i)} \in \mathbb{R}$</td>
</tr>
<tr>
<td>→ Classification</td>
<td>$y^{(i)} \in {1, \ldots, K}$</td>
</tr>
<tr>
<td>→ Binary classification</td>
<td>$y^{(i)} \in {+1, -1}$</td>
</tr>
<tr>
<td>→ Structured Prediction</td>
<td>$y^{(i)}$ is a vector</td>
</tr>
<tr>
<td>Unsupervised</td>
<td>$\mathcal{D} = {x^{(i)}}_{i=1}^{N}$ \quad $x \sim p^*(\cdot)$</td>
</tr>
<tr>
<td>Semi-supervised</td>
<td>$\mathcal{D} = {x^{(i)}, y^{(i)}}<em>{i=1}^{N_1} \cup {x^{(j)}}</em>{j=1}^{N_2}$</td>
</tr>
<tr>
<td>Online</td>
<td>$\mathcal{D} = {(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), (x^{(3)}, y^{(3)}), \ldots}$</td>
</tr>
<tr>
<td>Active Learning</td>
<td>$\mathcal{D} = {x^{(i)}}_{i=1}^{N}$ \quad and can query $y^{(i)} = c^*(\cdot)$ at a cost</td>
</tr>
</tbody>
</table>
### Learning Paradigms

<table>
<thead>
<tr>
<th>Paradigm</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supervised</td>
<td>$\mathcal{D} = {x^{(i)}, y^{(i)}}_{i=1}^N$</td>
</tr>
<tr>
<td></td>
<td>$\mathbf{x} \sim p^<em>(\cdot)$ and $y = c^</em>(\cdot)$</td>
</tr>
<tr>
<td>Regression</td>
<td>$y^{(i)} \in \mathbb{R}$</td>
</tr>
<tr>
<td>Classification</td>
<td>$y^{(i)} \in {1, \ldots, K}$</td>
</tr>
<tr>
<td>Binary classification</td>
<td>$y^{(i)} \in {+1, -1}$</td>
</tr>
<tr>
<td>Structured Prediction</td>
<td>$y^{(i)}$ is a vector</td>
</tr>
<tr>
<td>Unsupervised</td>
<td>$\mathcal{D} = {x^{(i)}}_{i=1}^N$</td>
</tr>
<tr>
<td>Semi-supervised</td>
<td>$\mathcal{D} = {x^{(i)}, y^{(i)}}<em>{i=1}^{N_1} \cup {x^{(j)}}</em>{j=1}^{N_2}$</td>
</tr>
<tr>
<td>Online</td>
<td>$\mathcal{D} = {(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), (x^{(3)}, y^{(3)}), \ldots}$</td>
</tr>
<tr>
<td>Active Learning</td>
<td>$\mathcal{D} = {x^{(i)}}_{i=1}^N$ and can query $y^{(i)} = c^*(\cdot)$ at a cost</td>
</tr>
<tr>
<td>Imitation Learning</td>
<td>$\mathcal{D} = {(s^{(1)}, a^{(1)}), (s^{(2)}, a^{(2)}), \ldots}$</td>
</tr>
</tbody>
</table>
## Learning Paradigms

<table>
<thead>
<tr>
<th>Paradigm</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supervised</td>
<td>( \mathcal{D} = { \mathbf{x}^{(i)}, y^{(i)} }_{i=1}^{N} ) \quad \mathbf{x} \sim p^<em>(\cdot) \text{ and } y = c^</em>(\cdot) )</td>
</tr>
<tr>
<td>Regression</td>
<td>( y^{(i)} \in \mathbb{R} )</td>
</tr>
<tr>
<td>Classification</td>
<td>( y^{(i)} \in {1, \ldots, K} )</td>
</tr>
<tr>
<td>Binary classification</td>
<td>( y^{(i)} \in {+1, -1} )</td>
</tr>
<tr>
<td>Structured Prediction</td>
<td>( y^{(i)} ) is a vector</td>
</tr>
<tr>
<td>Unsupervised</td>
<td>( \mathcal{D} = { \mathbf{x}^{(i)} }_{i=1}^{N} ) \quad \mathbf{x} \sim p^*(\cdot) )</td>
</tr>
<tr>
<td>Semi-supervised</td>
<td>( \mathcal{D} = { \mathbf{x}^{(i)}, y^{(i)} }<em>{i=1}^{N_1} \cup { \mathbf{x}^{(j)} }</em>{j=1}^{N_2} )</td>
</tr>
<tr>
<td>Online</td>
<td>( \mathcal{D} = {(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), (\mathbf{x}^{(3)}, y^{(3)}), \ldots} )</td>
</tr>
<tr>
<td>Active Learning</td>
<td>( \mathcal{D} = { \mathbf{x}^{(i)} }_{i=1}^{N} ) and can query ( y^{(i)} = c^*(\cdot) ) at a cost</td>
</tr>
<tr>
<td>Imitation Learning</td>
<td>( \mathcal{D} = {(s^{(1)}, a^{(1)}), (s^{(2)}, a^{(2)}), \ldots} )</td>
</tr>
<tr>
<td>Reinforcement Learning</td>
<td>( \mathcal{D} = {(s^{(1)}, a^{(1)}, r^{(1)}), (s^{(2)}, a^{(2)}, r^{(2)}), \ldots} )</td>
</tr>
</tbody>
</table>
Machine Learning: The Big Picture

Whiteboard

– **Decision Rules / Models** (probabilistic generative, probabilistic discriminative, perceptron, SVM, regression, MDP, graphical models)

– **Objective Functions** (likelihood, conditional likelihood, hinge loss, mean squared error)

– **Regularization** (L1, L2, priors for MAP)

– **Update Rules** (SGD, perceptron)

– **Nonlinear Features** (preprocessing, kernel trick)
ML Big Picture

Learning Paradigms: 
What data is available and when? What form of prediction?
- supervised learning
- unsupervised learning
- semi-supervised learning
- reinforcement learning
- active learning
- imitation learning
- domain adaptation
- online learning
- density estimation
- recommender systems
- feature learning
- manifold learning
- dimensionality reduction
- ensemble learning
- distant supervision
- hyperparameter optimization

Problem Formulation: 
What is the structure of our output prediction?
- boolean: Binary Classification
- categorical: Multiclass Classification
- ordinal: Ordinal Classification
- real: Regression
- ordering: Ranking
- multiple discrete: Structured Prediction
- multiple continuous: (e.g. dynamical systems)
- both discrete & cont.: (e.g. mixed graphical models) cont.

Facets of Building ML Systems:
How to build systems that are robust, efficient, adaptive, effective?
1. Data prep
2. Model selection
3. Training (optimization / search)
4. Hyperparameter tuning on validation data
5. (Blind) Assessment on test data

Theoretical Foundations: 
What principles guide learning?
- probabilistic
- information theoretic
- evolutionary search
- ML as optimization

Big Ideas in ML:
Which are the ideas driving development of the field?
- inductive bias
- generalization / overfitting
- bias-variance decomposition
- generative vs. discriminative
- deep nets, graphical models
- PAC learning
- distant rewards

Application Areas
Key challenges?
NLP, Speech, Computer Vision, Robotics, Medicine, Search
A new **combined** course...

...with the best (uphill climbs) from both
Course Level Objectives

You should be able to...

1. Implement and analyze existing learning algorithms, including well-studied methods for classification, regression, structured prediction, clustering, and representation learning
2. Integrate multiple facets of practical machine learning in a single system: data preprocessing, learning, regularization and model selection
3. Describe the formal properties of models and algorithms for learning and explain the practical implications of those results
4. Compare and contrast different paradigms for learning (supervised, unsupervised, etc.)
5. Design experiments to evaluate and compare different machine learning techniques on real-world problems
6. Employ probability, statistics, calculus, linear algebra, and optimization in order to develop new predictive models or learning methods
7. Given a description of a ML technique, analyze it to identify (1) the expressive power of the formalism; (2) the inductive bias implicit in the algorithm; (3) the size and complexity of the search space; (4) the computational properties of the algorithm; (5) any guarantees (or lack thereof) regarding termination, convergence, correctness, accuracy or generalization power.
Q&A