Midterm Review

We've covered a lot of material in the past six classes, so I wanted to take some time to go over the key concepts from each class that I want you to remember.

Lecture 1 – Introduction to Machine Learning

Definitions:

- Artificial Intelligence = The ability of a computer to perform tasks commonly associated with human beings.
- Machine learning = The field of study which gives computers the ability to learn without being explicitly programmed.
- Deep learning = A subfield of machine learning concerned with neural networks, which are algorithms inspired by the brain.

Hopefully this makes more sense now.

Types of problems:

- classification = discrete prediction
- regression = continuous prediction
- generation = output creation

next week

most of what we've done

we did a bit with recommender systems

Types of learning:

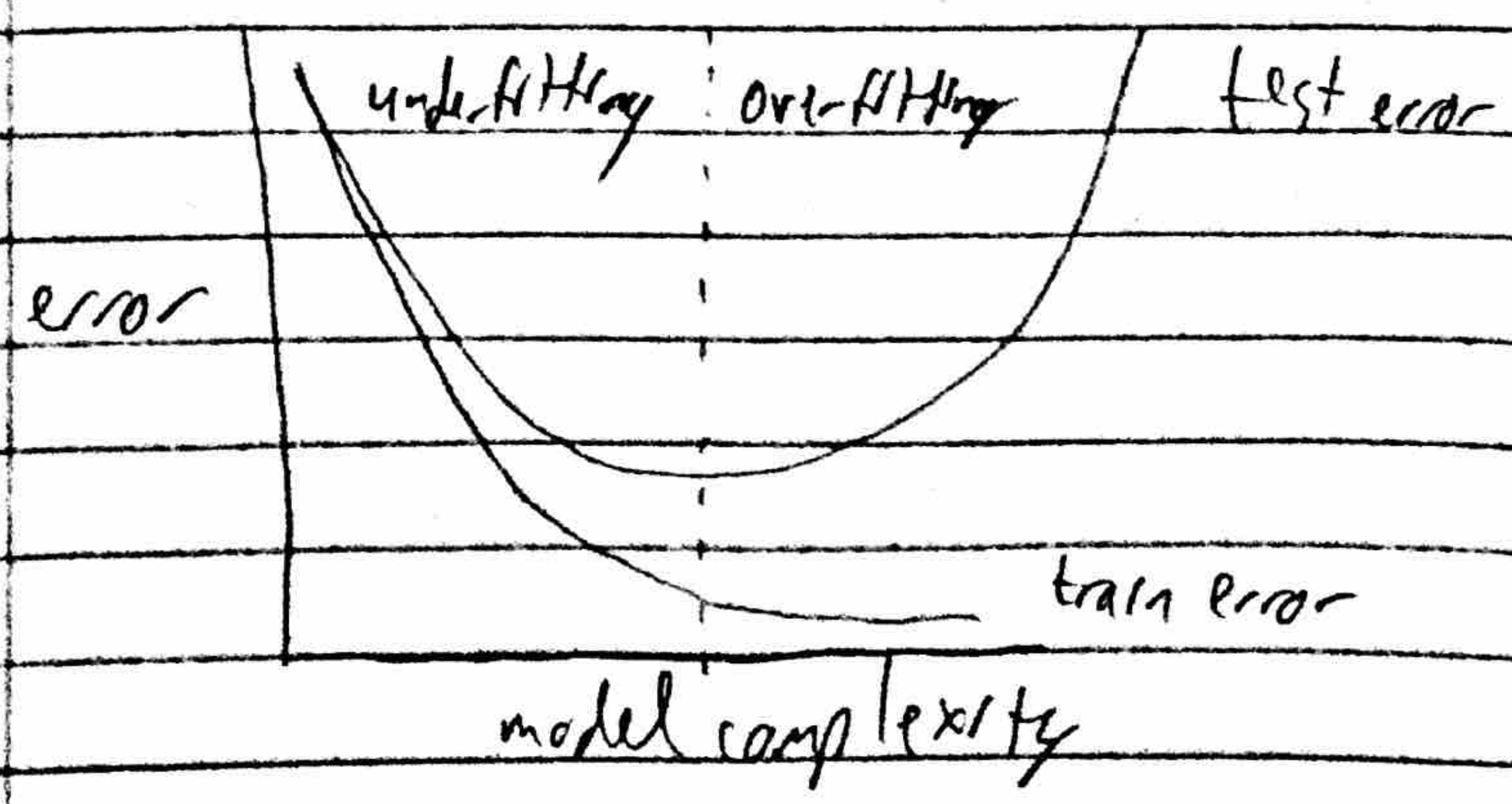
- Supervised learning = Given data and labels, predict the labels
- Reinforcement learning = Given task and reward function, learn to perform the task
- Unsupervised learning = Given data, learn underlying features

this is all we've done

last class

2nd to last class

Overfitting vs. generalization

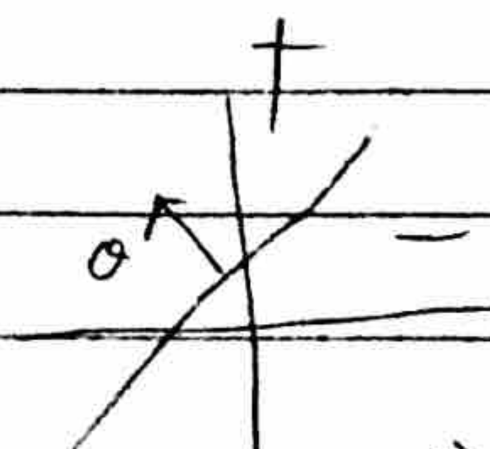


M-2

Lecture 2 - Linear Classifiers and the Perceptron Algorithm

A linear classifier is a line specified by two parameters, $\theta \in \mathbb{R}^d$ and $\theta_0 \in \mathbb{R}$.

The line is called a decision boundary. One one side we predict +1, on the other -1.



$\theta \cdot x + \theta_0 = 0$ decision boundary

$$h(x;\theta,\theta_0) = \text{sign}(\theta \cdot x + \theta_0) = \begin{cases} +1, & \theta \cdot x + \theta_0 > 0 \\ -1, & \theta \cdot x + \theta_0 \leq 0 \end{cases}$$

Perceptron algorithm = a mistake-driven algorithm for learning a linear classifier.

$\theta = \vec{0}$

$\theta_0 = 0$

repeat T times:

  for $i = 1, 2, \ldots n$:

    if $y^{(i)}(\theta \cdot x^{(i)} + \theta_0) \leq 0$:

      $\theta = \theta + y^{(i)} \cdot x^{(i)}$

      $\theta_0 = \theta_0 + y^{(i)}$

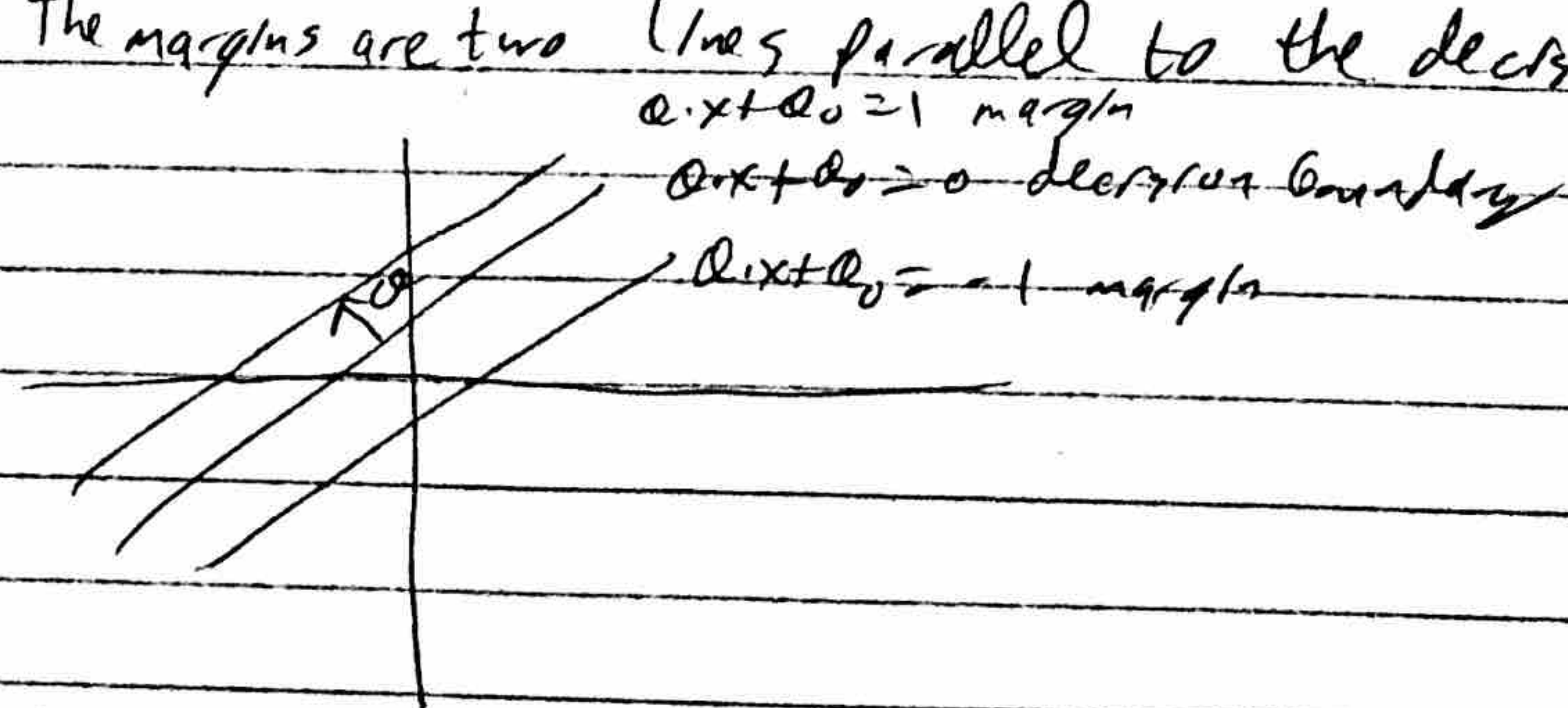
return θ, θ_0

1 of 2

Lecture 3 - Maximum Margin Classifiers and Support Vector Machines

We define the margin of a linear classifier to be the set of all points where $\theta \cdot x + \theta_0 = \pm 1$

The margins are two lines parallel to the decision boundary.



A larger margin helps our classifier generalize better.

We maximize the margin by minimizing θ $\left(D = \frac{1}{\|\theta\|}\right)$.

A support vector machine is a maximum margin linear classifier.

An SVM tries to minimize error while maximizing the margin.

Two ways to solve SVM:

1) offline - optimize across all data points (hard but optimal solution)
2) online - optimize across data points one by one (easy but non-optimal solution)

lab 3 → The pegasos algorithm is an online SVM algorithm.

Lecture 4 – Non-Linear Classifiers and Kernels

Most real-world datasets cannot be separated by a linear classifier.

Non-linear classifiers are needed to classify these datasets.

k-nearest neighbors (KNN) – find the k nearest training examples and predict the majority label

lab 4 first day → Non-linear transformations – find a function ϕ which can transform the data so that $\phi(x)$ is linearly separable

Building a linear classifier on the transformed data $\phi(x)$ is equivalent to building a non-linear classifier on the original data x.

Kernels – functions which allow us to learn a non-linear classifier without explicitly computing $\phi(x)$

lab 4 second day → Kernel perceptron – a modified version of the perceptron algorithm which uses kernels to learn a non-linear classifier

Lecture 5 – Ensembles and the Random Forest Algorithm

An ensemble is a collection of classifiers which makes predictions based on the majority vote of its classifiers.

Ensembles generalize better than individual classifiers, which can be biased.

Decision tree algorithm – a method for building a flowchart of rules to make a prediction. Rules are selected based on which rules best split the data.

Random forest algorithm – a method which builds an ensemble of decision trees using random subsets of the data and of the rules.

Lecture 6 – Recommender Systems

Linear regression – a method for learning parameters $\theta \in \mathbb{R}^d$ and $\theta_0 \in \mathbb{R}$ in order to predict real numbers with the function $f(x; \theta, \theta_0) = \theta \cdot x + \theta_0$

Content-based recommendation – a linear regression-based method for recommending content for a user based on the features of the content

Collaborative filtering – a method for recommending content by using both content features and other users' ratings

Low-rank matrix factorization – a collaborative filtering algorithm which makes predictions by learning a product of low-rank matrices $X = UV^T$