EECS 553: Machine Learning (ECE) **University of Michigan**

Course Introduction

Winter 2023 **Clayton Scott**

Machine learning is a field of study concerned with making quantitative predictions and inferences based on data. Machine learning theory and methodology emerged historically out of three areas: multivariate statistics, artificial intelligence, and signal processing. By now, the best practices of these areas have spread to the others, and machine learning has an independent identity. With the advent of "big data," scalability of algorithms is now a major theme and as such, machine learning is now also integrally connected to several areas of applied mathematics such as optimization and numerical linear algebra.

1 Terminology and Notation

A *feature* is an individual measurable property or characteristic of a phenomenon being observed [1]. In this course, features will usually be numerical, although many of the methods covered can be extended to accommodate nonnumerical features, such as graphs, strings, or even functions.

A *feature vector* is a vector of different numerical features associated to an observable phenomenon. The set of all possible feature vectors is called the *feature space*. Usually we take the feature space to be $\mathbb{R}^d$. Feature vectors will be denoted $\boldsymbol{x} \in \mathbb{R}^d$ and written

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_d \end{bmatrix} \tag{1}$$

where $x_1, \ldots, x_d$ are the features. Features are also called *attributes, predictors*, or *covariates*, and feature vectors are also called *data points, examples, instances, patterns, signals*, or *inputs*, depending on the context and personal preference. The terms "measurement" and "observation" can also be used for either of these concepts.

We will at times write features and feature vectors as capital letters to indicate that they are random variables.

2 Categories of Machine Learning Problems

There are several categories of machine learning problems. This course focuses on supervised and unsupervised learning in the offline setting.

2.1 Supervised vs Unsupervised Learning

In supervised learning, we observe *training data* $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)$ where y_i is the *label* (or *output* or *response*) associated to feature vector $\boldsymbol{x}_i$. The fundamental goal of supervised learning is *prediction*, namely, to predict the output y associated to a new input $\boldsymbol{x}$.

There are two main kinds of supervised learning problems. In *classification*, y is called a *label* and takes values in a finite set. If there are two labels, the problem is referred to as *binary classification*, otherwise it is referred to as *multiclass classification.* Figure 1 illustrates training data taken from a larger data set called the MNIST handwritten digit data set. The feature vectors are images, and the features are the individual pixel intensities. The goal is to use the training data to learn a function $f : \mathbb{R}^d \to \{0, 1, \ldots, 9\}$, called a *classifier*, and apply that function to accurately assign labels to future unlabeled handwritten digits.

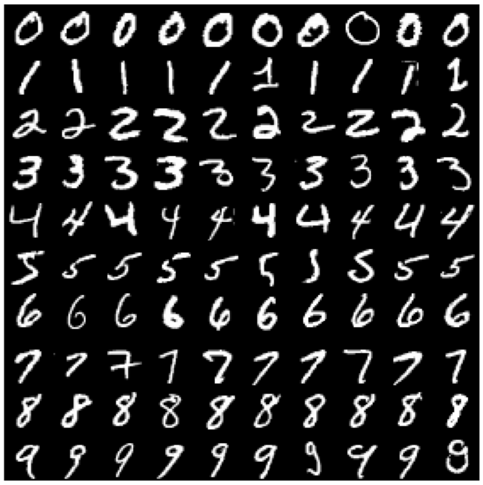

Figure 1: Several feature vectors for the multiclass classification problem of identifying handwritten digits. Each image can be viewed as a column vector by viewing the image as a matrix, and reshaping the matrix into a column vector.

In *regression*, y is called a *response* or *output variable* and takes values in $\mathbb{R}$. An example of a regression data set with $d = 1$ is shown in Figure 2. One can also have multi-dimensional outputs but this course will primarily focus the case where y is scalar.

In *unsupervised learning*, the feature vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ are *not* accompanied by output variables. The goal of unsupervised learning is usually not prediction but *inference*, that is, one seeks to understand (or infer) structure in the unlabeled data itself, or to infer some characteristic of the underlying probability distribution.

In *clustering*, the objective is to assign the unlabeled data points to groups called *clusters*, such that data points in the same group are more similar to each other than to other points. An example would be to automatically organize the handwritten digit data into 10 groups without knowing the true labels.[1] In *dimensionality reduction* the goal is to find a lower dimensional representation of a data set that does not lose too much information. For example, the MNIST data are $28 \times 28 = 784$ dimensional, but many of those dimensions are irrelevant for understanding the structure of the data set. In *density estimation*, the goal is to estimate the probability density function from which the $\boldsymbol{x}_i$ are drawn. Density estimates can be used to solve several problems in machine learning.

These are the main supervised and unsupervised learning problems we will study in this course, although there are several others. For example, ranking can be cast as a supervised learning problem, where the feature vector represents a sets of items, and the label is a ranking of those items.

There is also an entire spectrum of machine learning problems in between supervised and unsupervised learning. Such problems are often described as forms of *weakly supervised learning*. One example is *semi-supervised learning*, in which the learner is presented with both labeled and unlabeled data, and the goal is to design an accurate predictor. The idea is that labeled data can be expensive, but unlabeled data are cheap, and an abundance of unlabeled data can facilitate the discovery of structure that improves prediction. Other examples include outlier/anomaly detection and classification with noisy labels.

2.2 Offline vs. Online Learning

In off-line learning, often referred to as *batch learning*, all of the training data are presented at once. In online learning, the data arrive sequentially. All of the supervised and unsupervised problems described

[1] In many clustering problems, the clusters are not associated with class labels

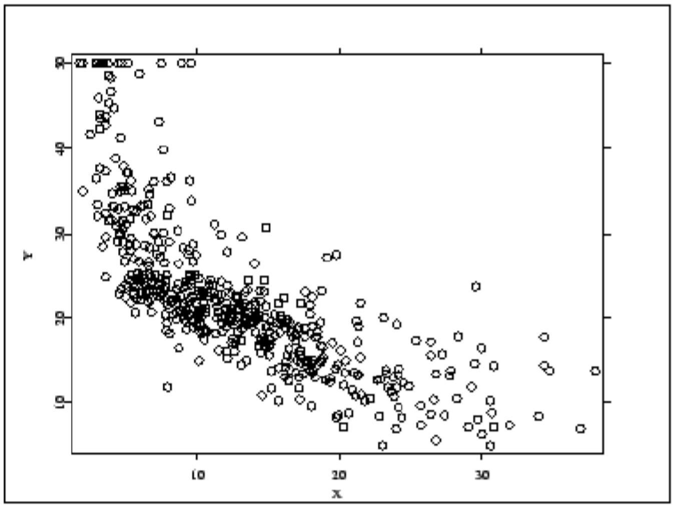

Figure 2: In regression the goal is to predict a continuous output y from the input x.

above can be cast in the online setting, where the objective is to update the solution as each new training instance arrives. There are also machine learning problems that can only be posed in an online setting. One example is *active learning*. In this problem, the learner is presented with an unlabeled data set, and requests the labels for one or more instances at a time. The idea is that acquiring labels is expensive, and to only request the labels for those data points that are most essential in designing a predictor. Another example is *reinforcement learning*, where an agent navigates an environment by taking actions and receiving rewards for those actions. The goal here is to maximize the cumulative reward. This course primarily focuses on batch learning.

2.3 Distributional Shift

A final category to describe a machine learning problem is whether it involves a distributional shift or not. In conventional machine learning (the focus of this course), it is assumed that the training and testing data follow the same distribution. In many applications, however, this is an unreasonable assumption. In *multitask learning*, the objective is to simultaneously solve several related prediction problems by leveraging similarities between these problems. In *domain adaptation*, the learner is presented with one or more data sets, and the goal is to optimize prediction performance on a new data set that is somehow related to the training data set(s). In *zero-shot learning*, training data are available only for some classes, and the learner must generalize to new classes given only semantic information relating the classes. *Transfer learning* is a broad term that is often used to mean domain adaptation, zero-shot learning, or some similar problem, and in the context of deep learning, it often refers to initializing with a *pre-trained network*, i.e., a neural network that was trained on a different task. Many forms of weakly supervised learning can be viewed as learning with distributional shift. We will not spend much time on distributional shift in this course, although these topics make for excellent projects.

3 Categories of Machine Learning Methods

Machine learning methods can be categorized based on various factors. The terms below will make more sense after we have covered some methods in detail. The definitions below are not meant to be totally precise (there are examples of methods that are difficult to categorize or fall into multiple categories), but rather to give us qualitative terms to discuss the various algorithms we will encounter.

3.1 Discrimantive vs. generative

A machine learning method is *generative* if it is based on a full probabilistic model for the data, in other words, a model that could be used to generate data (without needing to have seen data previously). A method is *discriminative* if it is not generative. Discriminative methods forego data modelling and focus directly on solving the problem of ultimate interest. For example, a discriminative classification method may aim to directly model the decision boundary. Some methods can be derived from both generative and discriminative perspectives.

3.2 Parametric vs. nonparametric

A method is said to be *nonparametric* if the amount of space needed to store the method's output (e.g., a classifier or density estimate) grows with the size of the training data. Otherwise, the method is said to be *parametric*. Most generative methods are parametric, because the learning algorithm amounts to estimating the parameters of the generative model, whose number is typically not dependent on sample size.

3.3 Linear vs. nonlinear

In general, a machine learning method is *linear* if its output is an affine function of the data. The precise definition of a linear method may vary from problem to problem. If a machine learning algorithm is not linear, it is said to be *nonlinear*. Linear methods are valuable because they are typically easy to implement and understand, and because they are often the basis of more complex, nonlinear methods.

3.4 Constraints

In many applications, we need a machine learning method to solve a learning problem while also satisfying some type of constraint. Prominent examples include scalability (computing or memory constraints), privacy, fairness, and safety.

4 About the Lecture Notes

These notes have not been subjected to the usual scrutiny reserved for formal publications. They may be distributed outside this class only with my permission. While I have made every effort to polish the text, the figures range from computer-generated to hand-drawn. I have also not yet had time to add appropriate references throughout.

Exercises may be found at the end of each set of lecture notes. Many of these (or ones very similar) will be given during the course as lecture polls, homework or exam problems. Every problem is labeled with one or more stars to indicate its difficulty and importance for the course. Filled stars indicate problems that you are expected to be able to solve, and may be assigned. Unfilled stars indicate problems that you will not be asked to solve, and are intended for those students who wish to gain further exposure to and mastery of the subject matter. All students are encouraged to at least read all exercises as they often convey important or interesting points about machine learning.

Table 1: Excercise labels.

Label	Difficulty	Essential?
(★)	Easy, definitions and basic comprehension	Yes
(★★)	Medium, requires use of concepts and techniques already covered	Yes
(★★★)	Challenging, may require solution techniques that do not mimic those already covered	Yes
(☆)	Easy	No
(☆☆)	Medium	No
(☆☆☆)	Challenging	No

References

[1] Christopher Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.