

Overview and Examples

Sanjay Lall and Stephen Boyd

EE104
Stanford University

Machine learning overview

Overview

- ▶ we describe here the general ideas and methods used in machine learning, at a high level
- ▶ we will go over all of these topics later, in much more detail
- ▶ don't worry if some of this seems abstract at this point, or there are terms you don't know yet

Artificial intelligence approaches

- ▶ we would like computers to perform complicated tasks, e.g., medical diagnosis
- ▶ distinguish two approaches
 - ▶ *knowledge-based*: a computer program whose logic encodes a large number of properties of the world, usually developed by a team of experts over many years
 - ▶ *machine learning*: extract information directly from historical data and extrapolate to make predictions
- ▶ this class is about machine learning

Machine learning tasks

generic tasks:

- ▶ build a *model* from some *data*
 - ▶ choose how to map raw data to feature vectors
 - ▶ choose a model form
 - ▶ choose parameter values in the model
- ▶ *test* or *validate* the model
 - ▶ evaluate the model on unseen data to assess its performance

'model' can mean several things, depending on context

Machine learning model taxonomy

- ▶ *supervised* versus *unsupervised* models
- ▶ supervised learning models *predict something, given some other things*
 - ▶ called a *prediction model*
 - ▶ called *regression* when we predict a real scalar or vector value
 - ▶ called *classification* when we predict a value from a finite set such as $\{\text{TRUE}, \text{FALSE}\}$
- ▶ unsupervised learning models just create a model of the data
 - ▶ called a *data model*
- ▶ point estimates versus probability estimates
 - ▶ a *point estimate* predicts a single value
 - ▶ a *probability estimate* predicts a distribution of values
- ▶ the lines between these can be blurred

Examples

what kind of models would each of these tasks use?

- ▶ predict tomorrow's rainfall, given the date and the last 10 days of rainfall data
- ▶ determine from a photo of a face if the user is who she claims to be
- ▶ estimate the probability of 10 possible diagnoses, given some patient data, test results
- ▶ cluster customers into 22 different groups with similar buying habits
- ▶ estimate the risk (probability) of an auto accident at a location given the hour and weather
- ▶ build an *anomaly detector*, that rates how suspicious some new data is
- ▶ build a *simulator* that generates fake new data that 'looks like' the given data

Performance metrics

- ▶ we judge performance of a model on some data using a *metric* such as
 - ▶ mean-square or RMS prediction error (for regression)
 - ▶ error rate (for classification)
 - ▶ log likelihood (for probabilistic models)
- ▶ examples:
 - ▶ our predictor predicts tomorrow's maximum temperature with an RMS error of 1.3°C
 - ▶ our classifier predicts the topic of a newspaper article (from a set of 50 choices) with an error rate of 5%

Training and validation

- ▶ our goal is to develop a model that performs well for *new, unseen data*
- ▶ standard practice is to divide the given data set into two parts
 - ▶ a *training* data set, used to choose or train the model
 - ▶ a *test* or *validation* data set, used to evaluate tentative models
- ▶ we can look at the performance metrics on the *training* and *test* data sets
- ▶ if the model performs well on the training set, but poorly on the test set, it is *overfit* (and probably useless in practice)
- ▶ if the model performs well on the test set, it's likely going to perform well on new unseen data

Learning a model

a common method of choosing a model:

- ▶ choose the *model structure* or *form* or *type*
- ▶ the model contains a number of *model parameters*
- ▶ choose a *loss function* that rates how badly the model performs on a single data point or example
- ▶ choose the parameter value by minimizing an average loss over the training data

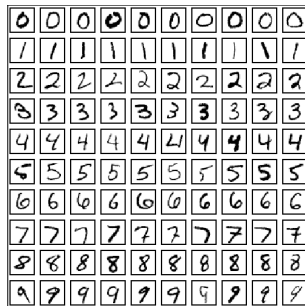
this general scheme, called *empirical risk minimization*, is used to fit a wide variety of models

Examples

Example: diagnosis

- ▶ goal is to predict if a patient has a disease, based on whether or not she exhibits 10 symptoms
- ▶ historical data consists of a large number of *patient records*
- ▶ each record contains
 - ▶ 10 Booleans, specifying the presence or absence of the 10 symptoms
 - ▶ a Boolean specifying whether that patient had the disease
- ▶ machine learning algorithm observes these data, produces a *predictor*
- ▶ predictor takes as input 10 Booleans, returns a single Boolean prediction
- ▶ this is a *classifier*, since we are predicting an outcome that takes only two values
- ▶ we will judge model by its error rate on a separate test set of data, not used to develop the model
- ▶ a *probabilistic model* returns a probability that the patient has the disease, not just a Boolean

Example: digit recognition



- ▶ 8-bit grayscale images of handwritten digits, 28×28 pixels
- ▶ goal is to guess the digit (*i.e.*, $0, \dots, 9$) from the image
- ▶ 60,000 training images, 10,000 test images
- ▶ preprocessed by antialiasing, scaling and centering
- ▶ data originally by NIST, modified by Le Cun, 1998

Data

- ▶ Kaggle: datasets and competitions
- ▶ ImageNet dataset: 14m images
- ▶ Street view house numbers: 600,000 digit images
- ▶ Waymo open dataset: self-driving car data
- ▶ many other large datasets

Software

- ▶ Flux, PyTorch, Keras, TensorFlow, JAX, and many others, ...
- ▶ Scikit-Learn, Spark/MLlib
- ▶ Flux, in Julia
- ▶ generic packages in R, Matlab, Python, Java, Julia ...

Course outline

The major parts of the course

1. *regression* — to predict one or more real values
2. *classification* — to predict one of a finite number of possible outcomes
3. *probabilistic supervised learning* — to predict a distribution of outcomes
4. *unsupervised learning* — to develop a data model
5. *optimization* — to fit or choose parameters in all of the models above