ECBM E6040 Neural Networks and Deep Learning

Lecture #4: Machine Learning Basics

Aurel A. Lazar

Columbia University
Department of Electrical Engineering

February 9, 2016

Outline of Part I

2 Summary of the Previous Lecture
- Topics Covered
- Learning Objectives

Outline of Part II

3. Machine Learning Algorithms
   - The Task $\mathcal{T}$, Performance Measure $\mathcal{P}$ and Experience $\mathcal{E}$
   - Example: Linear Regression
4. Estimation, Bias and Variance
   - Point Estimation
   - Bias
   - Variance and Standard Error
   - Trading Off Bias and Variance and MSE
5. Maximum Likelihood Estimation
   - Conditional Log-Likelihood and MSE
   - Properties of Maximum Likelihood

Part I

Review of Previous Lecture

Topics Covered

- Probability and Information Theory
    - Probability Spaces and Random Variables
    - Elements of Information Theory
- Numerical Computation
    - Gradient-Based Optimization
    - Constraint Optimization
    - Linear Least Squares

Learning Objectives

- Reviewing the main concept of probability theory that are used for (i) reasoning in machine learning systems and, (ii) to analyze and predict the performance of learning systems.
- Deep learning algorithms are often the solution to optimization problems.

Part II

Today's Lecture

Learning Algorithms

A machine learning algorithm is an algorithm that is able to learn from data.

A computer program is said to learn from experience $\mathcal{E}$ with respect to some class of tasks $\mathcal{T}$ and performance measure $\mathcal{P}$, if its performance at tasks in $\mathcal{T}$, as measured by $\mathcal{P}$, improves with experience $\mathcal{E}$.

There are very wide variety of experiences $\mathcal{E}$, tasks $\mathcal{T}$, and performance measures $\mathcal{P}$ - no formal definition.

The Task $\mathcal{T}$

Learning is the means of attaining the ability to perform a task.
Common machine learning tasks:

- Classification
- Regression
- Density or Probability Function Estimation

Common Machine Learning Task: Classification

The computer program is asked to specify which of k categories some input belongs to.

- The learning algorithm is usually asked to produce a function $f : \mathbb{R}^n \to \{1, ..., k\}$ which may then be applied to any input. Here the output of $f(\mathbf{x})$ can be interpreted as an estimate of the category that $\mathbf{x}$ belongs to. There are other variants of the classification task, for example, where f outputs a probability distribution over classes.
- Example: object recognition, where the input is an image (usually described as a set of pixel brightness values), and the output is a numeric code identifying the object in the image. Object recognition allows computers to recognize faces and it is used to automatically tag people in photo collections.

Common Machine Learning Task: Regression

The computer program is asked to predict a numerical value given input data.

- The learning algorithm is asked to output a function $f : \mathbb{R}^n \to \mathbb{R}$. This type of task is similar to classification, except that the format of output is different.
- Example: prediction of the expected claim amount that an insured person will make (used to set insurance premia), or the prediction of future prices of securities used for algorithmic trading.

Common Machine Learning Task: Density Estimation

The learning algorithm has to capture the structure of the probability distribution on the space of examples. Density estimation allows us to explicitly capture that distribution.

- The machine learning algorithm is asked to learn a function $p_{model} : \mathbb{R}^n \rightarrow \mathbb{R}$, where $p_{model}(\mathbf{x})$ represents the probability density function (if $\mathbf{x}$ is continuous) or a probability function (if $\mathbf{x}$ is discrete) on the space that the examples were drawn from.
- Example: density estimation can be used to solve the missing value imputation task. If a value x^i is missing and all of the other values, denoted $\mathbf{x}_{-i}$ are given, then the distribution is given by $p(x_i|\mathbf{x}_{-i})$.

The Performance Measure $\mathcal{P}$

The performance measure $\mathcal{P}$ is specific to the task being carried $\mathcal{T}$ out by the system. In the real world we typically evaluate these performance measures using a test set of data that is separate from the data used for training the machine learning system.

- Accuracy of the model: the proportion of examples for which the model produces the correct output.
- Error rate: the proportion of examples for which the model produces an incorrect output.

The Experience $\mathcal{E}$

Machine learning algorithms are broadly categorized as unsupervised or supervised by what kind of experience they are allowed to have during the learning process.

Most learning algorithms experience an entire dataset. A dataset is a collection of many objects called examples. Each example contains many features that have been objectively measured. Examples are also called data points.

Unsupervised learning algorithms experience a dataset containing many features, then learn useful properties of the structure of this dataset.

Supervised learning algorithms experience a dataset containing features, but each example is also associated with a label or target

The Experience $\mathcal{E}$ (cont'd)

Roughly speaking

- unsupervised learning involves observing several examples of a random vector $\mathbf{x}$, and attempting to implicitly or explicitly learn the probability distribution $p(\mathbf{x})$, or some interesting properties of that distribution. In unsupervised learning, there is no instructor or teacher, and the algorithm must learn to make sense of the data without this guide.
- supervised learning involves observing several examples of a random vector $\mathbf{x}$ and an associated value or vector $\mathbf{y}$, and learning to predict $\mathbf{y}$ from $\mathbf{x}$, e.g., estimating $p(\mathbf{y}|\mathbf{x})$. The term supervised learning originates from the view of the target $\mathbf{y}$ being provided by an instructor or teacher that shows the machine learning system what to do.

Example: Linear Regression

A Simpe Machine Learning Algorithm

Build a linear system that can take a vector $\mathbf{x} \in \mathbb{R}^n$ as input and predict the value of a scalar $\mathbf{y} \in \mathbb{R}^n$ as its output:

$$\hat{y} = \mathbf{w}^T \mathbf{x},$$

where $\mathbf{w} \in \mathbb{R}^n$ is a vector of parameters called weights.

Task $\mathcal{T}$: predict y from $\mathbf{x}$ by computing $\hat{y} = \mathbf{w}^T \mathbf{x}$.

Example: Linear Regression (cont'd)

We assume that the design matrix of m example inputs will not be used for training, only for evaluating how well the model performs. The design matrix of inputs is denoted by $\mathbf{X}^{test}$ and the vector of regression targets by $\mathbf{y}^{test}$.

The mean square error is given by

$$MSE_{test} = \frac{1}{m} \sum_i [\hat{\mathbf{y}}^{test} - \mathbf{y}^{test}]_i^2 .$$

Note that

$$MSE_{test} = \frac{1}{m} ||\hat{\mathbf{y}}^{test} - \mathbf{y}^{test}||_2^2 .$$

Example: Linear Regression (cont'd)

To minimize MSE_{train} we simply solve

$$\nabla_{\mathbf{w}} MSE_{train} = 0$$

that is

$$\nabla_{\mathbf{w}} \frac{1}{m} ||\hat{\mathbf{y}}^{train} - \mathbf{y}^{train}||_2^2 = 0$$

or

$$\frac{1}{m} \nabla_{\mathbf{w}} ||\mathbf{X}^{train}\mathbf{w} - \mathbf{y}^{train}||_2^2 = 0$$

or in inner product for

$$\nabla_{\mathbf{w}} (\mathbf{X}^{train}\mathbf{w} - \mathbf{y}^{train})^T (\mathbf{X}^{train}\mathbf{w} - \mathbf{y}^{train}) = 0,$$

$$\nabla_{\mathbf{w}} (\mathbf{w}^T \mathbf{X}^{(train)T} \mathbf{X}^{train} \mathbf{w} - 2\mathbf{w}^T \mathbf{X}^{(train)T} \mathbf{y}^{train} + \mathbf{y}^{(train)T} \mathbf{y}^{train}) = 0.$$

Example: Linear Regression (cont'd)

Normal Equations

$$2\mathbf{X}^{(train)T}\mathbf{X}^{train}\mathbf{w} - 2\mathbf{X}^{(train)T}\mathbf{y}^{train} = 0$$

or finally

$$\mathbf{w} = (\mathbf{X}^{(train)T}\mathbf{X}^{train})^{-1}\mathbf{X}^{(train)T}\mathbf{y}^{train}.$$

This system of equations is known as the normal equations.

Basic Goals

The field of statistics gives us many tools that can be used to achieve the machine learning goal of solving a task not only on the training set but also to generalize.

Foundational concepts such as parameter estimation, bias and variance are useful to formally characterize notions of generalization, underfitting and overfitting.

Point Estimation

Let $\{\mathbf{x}^1, ..., \mathbf{x}^m\}$ be a set of m i.i.d. data points. A point estimator is a function of the data

$$\hat{\theta}_m = g(\mathbf{x}^1, ..., \mathbf{x}^m).$$

The true parameter value θ is fixed but unknown, while the point estimate $\hat{\theta}_m$ is a function of the data. Since the data is drawn from a random process, any function of the data is random. Therefore $\hat{\theta}_m$ is a random variable whose probability distribution is called the sampling distribution.

Bias

The bias of an estimator is defined as

$$\text{bias}\,(\hat{\theta}_m) = \mathbb{E}\,\hat{\theta}_m - \theta$$

where the expectation is over the data (seen as samples from a random variable) and θ is the true underlying value of θ according to the data generating distribution.

An estimator $\hat{\theta}_m$ is said to be unbiased if $\text{bias}\,(\hat{\theta}_m) = 0$, i.e., if $\mathbb{E}(\hat{\theta}_m) = \theta$.

An estimator $\hat{\theta}_m$ is said to be asymptotically unbiased if $\lim_{m\to\infty} \text{bias}(\hat{\theta}_m) = 0$, i.e., if $\lim_{m\to\infty} \mathbb{E}(\hat{\theta}_m) = \theta$.

Bernoulli Distribution Estimator of the Mean

An Example

Is the estimator $\hat{\theta}_m = \frac{1}{m}\sum_{i=1}^{m} x^i$ biased?

$$\text{bias}\,(\hat{\theta}_m) = \mathbb{E}\,[\hat{\theta}_m] - \theta = \mathbb{E}\,[\frac{1}{m}\sum_{i=1}^{m} x^i] - \theta = \frac{1}{m}\sum_{i=1}^{m}\mathbb{E}\,[x^i] - \theta$$

i.e.,

$$\text{bias}\,(\hat{\theta}_m) = \frac{1}{m}\sum_{i=1}^{m}\sum_{x^i=0}^{1}[x^i\theta^{x^i}(1-\theta)^{1-x^i}] - \theta = \frac{1}{m}\sum_{i=1}^{m}\theta - \theta = 0.$$

Gaussian Distribution Estimator of the Mean

An Example

A widely used Gaussian mean estimator is given by

$$\hat{\mu}_m = \frac{1}{m}\sum_{i=1}^{m} x^i .$$

The bias of the sample mean amounts to

$$\text{bias}\left(\hat{\mu}_m\right) = \mathbb{E}\left[\hat{\mu}_m\right] - \mu = \mathbb{E}\frac{1}{m}\sum_{i=1}^{m} x^i - \mu = \frac{1}{m}\sum_{i=1}^{m}\mathbb{E}\, x^i - \mu = 0.$$

Gaussian Distribution Estimators of the Variance
An Example

We consider here first the sample variance estimator:

$$\hat{\sigma}_m^2 = \frac{1}{m}\sum_{i=1}^{m}(x^i - \hat{\mu}_m)^2 \to \mathbb{E}\,\hat{\sigma}^2 = \mathbb{E}\frac{1}{m}\sum_{i=1}^{m}[(x^i)^2 - 2x^i\hat{\mu}_m + \hat{\mu}_m^2].$$

$$\mathbb{E}\,\hat{\sigma}^2 = \frac{1}{m}\sum_{i=1}^{m}\mathbb{E}\,(x^i)^2 - \mathbb{E}\hat{\mu}_m^2 = \frac{1}{m}\sum_{i=1}^{m}\mathbb{E}\,(x^i)^2 - \mathbb{E}\frac{1}{m}\sum_{i=1}^{m}x^i\frac{1}{m}\sum_{j=1}^{m}x^j$$

$$\mathbb{E}\,\hat{\sigma}^2 = \frac{1}{m}(1 - \frac{1}{m})\sum_{i=1}^{m}\mathbb{E}\,(x^i)^2 - \frac{1}{m^2}\sum_{i=1}^{m}\sum_{j\neq i}\mathbb{E}\,x^i x^j$$

$$\mathbb{E}\,\hat{\sigma}^2 = \frac{1}{m}(1 - \frac{1}{m})m(\sigma^2 + \mu^2) - \frac{1}{m^2}m(m-1)\mu^2 = \frac{m-1}{m}\sigma^2.$$

Gaussian Distribution Estimators of the Variance (cont'd)

An Example

Therefore, the sample variance is a biased estimator and

$$\text{bias}\,(\hat{\sigma}^2) = \mathbb{E}\,\hat{\sigma}_m^2 - \sigma^2 = -\frac{1}{m}\sigma^2.$$

The modified sample variance

$$\tilde{\sigma}_m^2 = \frac{1}{m-1}\sum_{i=1}^{m}(x^i - \hat{\mu}_m)^2$$

has variance

$$\mathbb{E}\,\tilde{\sigma}_m^2 = \mathbb{E}\frac{1}{m-1}\sum_{i=1}^{m}(x^i - \hat{\mu}_m)^2 = \frac{m}{m-1}\mathbb{E}\,\hat{\sigma}_m^2 = \frac{m}{m-1}\frac{m-1}{m}\sigma^2 = \sigma^2.$$

Variance and Standard Error

Recall that the variance is given by

$$\mathrm{Var}\,(\hat{\theta}) = \mathbb{E}\,\hat{\theta}^2 - [\,\mathbb{E}\hat{\theta}\,]^2.$$

and the standard error by

$$\mathrm{SE}\,(\hat{\theta}) = \sqrt{\mathrm{Var}\,(\hat{\theta})}.$$

Also, the standard deviation of the mean amounts to

$$\mathrm{Var}(\hat{\theta}_m) = \sqrt{\mathrm{Var}[\frac{1}{m}\sum_{i=1}^{m} x^i] = \frac{\sigma}{\sqrt{m}}},$$

where σ is the variance of the samples x^i.

Variance and Standard Error (cont'd)

Example: Bernoulli Distribution

We consider the estimator $\hat{\theta}_m = \frac{1}{m}\sum_{i=1}^{m} x^i$, where the samples $\{x^i, ..., x^m\}$ are iid with Bernoulli distribution with parameter θ.

$$\text{Var}(\hat{\theta}_m) = \text{Var}(\frac{1}{m}\sum_{i=1}^{m} x^i) = \frac{1}{m^2}\sum_{i=1}^{m}\text{Var}(x^i) = \frac{1}{m^2}\sum_{i=1}^{m}\theta(1-\theta) = \frac{\theta(1-\theta)}{m}.$$

Variance and Standard Error (cont'd)

Example: Gaussian Distribution Estimators of the Variance

$\{x^i, ..., x^m\}$ are iid with Gaussian distribution with (μ, σ^2).

$$\text{Var}(\frac{m-1}{\sigma^2}\tilde{\sigma}^2) = 2(m-1)$$

or

$$\frac{(m-1)^2}{\sigma^4}\text{Var}\left(\tilde{\sigma}^2\right) = 2(m-1)$$

Finally, since $\hat{\sigma}^2 = \frac{m-1}{m}\tilde{\sigma}^2$ and the relationship to χ^2 distribution

$$\text{Var}(\hat{\sigma}^2) = \frac{2(m-1)\sigma^4}{m^2}.$$

Trading Off Bias and Variance

Gaussian Distribution of the Variance

$$\text{MSE} = \mathbb{E}[\hat{\theta}_n - \theta]^2 = \text{Bias}(\hat{\theta}_n^2) + \text{Var}(\hat{\theta}_n)$$

$$\text{MSE}(\hat{\sigma}_m^2) = \text{Bias}(\hat{\sigma}_m^2)^2 + \text{Var}(\hat{\sigma}_m^2)^2 = (\frac{-\sigma^2}{m})^2 + \frac{2(m-1)\sigma^4}{m^2} = \frac{2m-1}{m^2}\sigma^4.$$

The MSE of the unbiased alternative is given by

$$\text{MSE}(\tilde{\sigma}_m^2) = \text{Bias}(\tilde{\sigma}_m^2)^2 + \text{Var}(\tilde{\sigma}_m^2)^2 = 0 + \frac{2\sigma^4}{m-1} = \frac{2}{m-1}\sigma^4.$$

The Maximum Likelihood Principle

Consider a set of m examples $\mathbb{X} = \{\mathbf{x}^1, ..., \mathbf{x}^m\}$ drawn from the unknown distribution $p_{data}(\mathbf{x})$.

Let $p_{model}(\mathbf{x}; \theta)$ be a parametric family of probability distributions over the same space indexed by θ. In other words, $p_{model}(\mathbf{x}; \theta)$ maps any configuration $\mathbf{x}$ to a real number estimating the true probability $p_{data}(\mathbf{x})$.

The maximum likelihood estimator for θ is then defined as

$$\theta_{ML} = \arg\max_{\theta} p_{model}(\mathbb{X}; \theta) = \arg\max_{\theta} \prod_{i=1}^{m} p_{model}(\mathbf{x}^i; \theta).$$

Note that it is easier to work with the formulation

$$\theta_{ML} = \arg\max_{\theta} \sum_{i=1}^{m} \log p_{model}(\mathbf{x}^i; \theta).$$

The Maximum Likelihood Principle (cont'd)

By rescaling the cost function (dividing by m) we obtain

$$\theta_{ML} = \arg\max_{\theta} \mathbb{E} \log p_{model}(\mathbf{x}^i; \theta).$$

Maximum likelihood estimation is equivalent with minimizing the dissimilarity between the empirical distribution defined by the training set and the model distribution, with the degree of dissimilarity between the two measured by the KL divergence. The KL divergence is given by

$$\text{DKL}(\hat{p}_{data}||p_{model}) = \mathbb{E}[\log \hat{p}_{data}(\mathbf{x}) - \log p_{model}(\mathbf{x})].$$

The term on the left is a function only of the data generating process, not the model. This means when we train the model to minimize the KL divergence, we need only minimize

$$-\mathbb{E} \log p_{model}(\mathbf{x}).$$

Conditional Log-Likelihood and MSE

In order to predict $\mathbf{y}$ given $\mathbf{x}$, we need to estimate the conditional probability $\mathbb{P}(\mathbf{y}|\mathbf{x};\theta)$. This is actually the most common situation because it forms the basis for most supervised learning, the setting where the examples are pairs $(\mathbf{x},\mathbf{y})$.

If $\mathbf{X}$ represents all our inputs and $\mathbf{Y}$ all our observed targets, then the conditional maximum likelihood estimator is

$$\theta_{ML} = \arg\max_{\theta} \mathbb{P}(\mathbf{Y}|\mathbf{X};\theta).$$

If the examples are assumed to be i.i.d., then this can be decomposed into

$$\theta_{ML} = \arg\max_{\theta} \sum_{i=1}^{m} \log \mathbb{P}(\mathbf{y}^i|\mathbf{x}^i;\theta).$$

Example: Linear Regression

The conditional density of $\mathbf{y}$, given $\mathbf{x} = \mathrm{x}$, is a Gaussian with mean $\mu(\mathbf{x})$ that is a learned function of $\mathbf{x}$, with unconditional variance σ^2. Since the examples are assumed to be i.i.d., the conditional log-likelihood

$$\log \mathbb{P}(\mathbf{Y}|\mathbf{X};\theta) = \sum_{i=1}^{m} \log \mathbb{P}(\mathbf{y}^i|\mathbf{x}^i;\theta)$$

$$= \sum_{i=1}^{m} \frac{-1}{2\sigma^2} ||\hat{\mathbf{y}}^i - \mathbf{y}^i)||^2 - m \log \sigma - \frac{m}{2} \log(2\pi)$$

where $\mathbf{y}^i = \mu(\mathbf{x}^i)$ is the output of the linear regression on the i-th input $\mathbf{x}^i$ and m is the dimension of the $\mathbf{y}$ vectors.

Example: Linear Regression (cont'd)

If σ is fixed, maximizing the above is equivalent (up to an additive and a multiplicative constant that do not change the value of the optimal parameter) to minimizing the training set mean squared error, i.e.,

$$\mathrm{MSE}_{train} = \frac{1}{m}\sum_{i=1}^{m} ||\hat{\mathbf{y}}^i - \mathbf{y}^i)||^2.$$

Note that the MSE is an average rather than a sum, which is more practical from a numerical point of view (so you can compare MSEs of sets of different sizes more easily).

Properties of Maximum Likelihood

The maximum likelihood estimator has the property of consistency. That parametric mean squared error decreases as m increases, and for m large, the Cramér-Rao lower bound shows that no consistent estimator has a lower mean squared error than the maximum likelihood estimator.