Fundamentals of Machine Learning:

Introduction and Basic Concepts

Prof. Andrew D. Bagdanov (`andrew.bagdanov AT unifi.it`)

UNIVERSITÀ DEGLI STUDI FIRENZE

DINFO DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

Outline

Introduction

Mathematical Preliminaries: Linear Algebra

Mathematical Preliminaries: Probability and Statistics

Notational Alignment and the Way Forward

Homework and Reading

Introduction

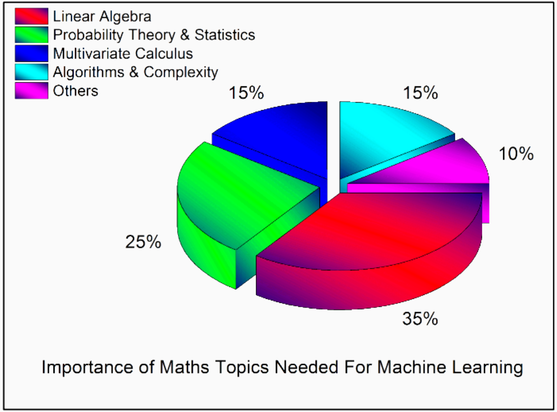
Lecture objectives

At the end of this lecture you will:

- Have refreshed your memory on the basic concepts and operations of linear algebra.
- Have refreshed your memory on the basic rules of probability theory, namely the sum and product rules.
- Have refreshed your memory on conditional probabilities and Bayes theorem.
- Have acquired a basic intuition about probabilistic decision theory.
- Have an intuitive understanding of the Curse of Dimensionality.

The mathematics of the 21st century

- Mastering contemporary machine learning requires a range of tools and disciplines...



Linear algebra

- Skyler Speakerman recently referred to Linear Algebra as the *mathematics for the 21st Century*.
- This might be slightly hyperbolic, but linear algebra is absolutely central to modern machine learning.
- Linear algebra allows us to deal with high dimensional data in a formal and precise way.
- It will allow us to model inputs to ML algorithms as points in high dimensional spaces.
- And subsequently to model functional transformations of these inputs into feature spaces.
- And finally, to model the subsequent transformations that lead to outputs (e.g. decisions or actions or estimates).

Linear algebra (continued)

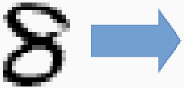
- What is an image? Is it a data structure, with width and height and depth, plus a corresponding array of raw data?
- We can go on... What is an audio recording? Or text document.
- Rather than define *ad hoc* structures, we want to treat everything the same.
- A 512×512 color image is a vector in a $512 \times 512 \times 3$-dimensional vector space.

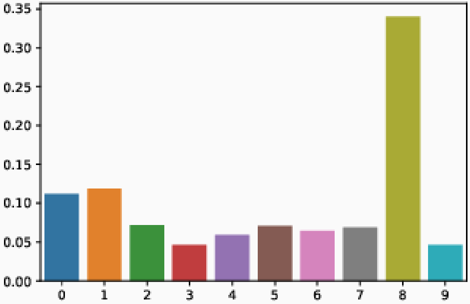
Probability and statistics

- Perhaps somewhat surprisingly, probability and statistics are less important to modern machine learning.
- Sometimes we will want to give a probabilistic interpretation to a model or a model output.
- However, most deep learning models are defined as pure transformations of inputs into outputs.
- Often, these probabilistic interpretations are merely convenient fictions.
- As we will see, statistics and probability are very useful as tools for analyzing results.

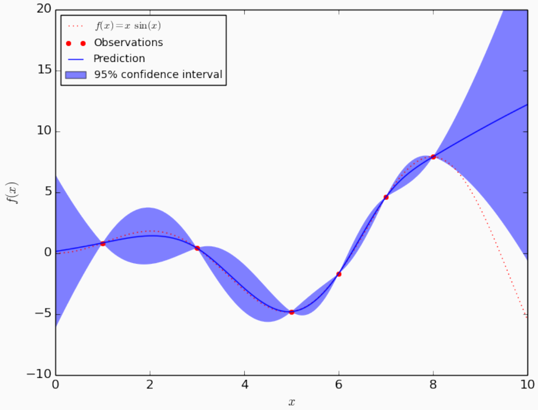
Probability and statistics (continued)

- For many problems we will want our models to output a probability distribution over possible outcomes.
- Take a simple classification problem: given an input image, estimate which digit is depicted.



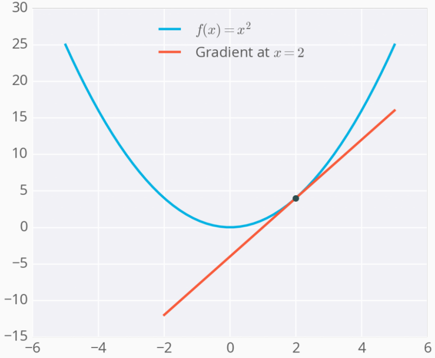


- For other problems we might want to qualify outputs of the model.
- This is the case in many regression problems where outputs at some points might be more certain than others.



Calculus and optimization

- Many (well, most) learning problems are formulated as optimization problems in (potentially very many) multiple variables.
- This means that to learn means to estimate these problems by minimizing some objective function.



Mathematical Preliminaries: Linear Algebra

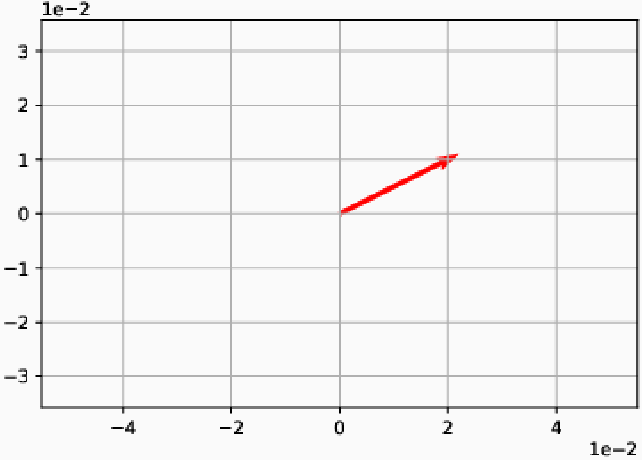
Vectors and vector spaces

- Vectors and vector spaces are fundamental to linear algebra.
- Vectors describe lines, planes, and hyperplanes in space.
- They allow us to perform calculations that explore relationships in multi-dimensional spaces.
- At its simplest, a vector is a mathematical object that has both magnitude and direction.
- We write vectors using a variety of notations, but we will usually write them like this:

$$\mathbf{v} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

- The boldface symbol lets us know it is a vector.

- What does it mean to have direction and magnitude?
- Well, it helps to look at a visualization (in at most three dimensions):



Vectors and vector spaces (continued)

More formally, we say that $\mathbf{v}$ is a vector in n dimensions (or rather, $\mathbf{v}$ is a vector in the vector space $\mathbb{R}^n$) if:

$$\mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}$$

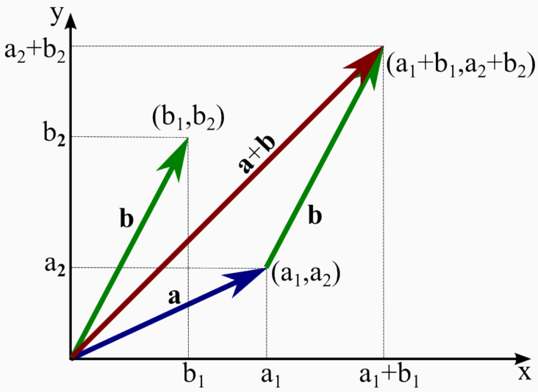
for $v_i \in \mathbb{R}$. Note that we use regular symbols (i.e. not boldfaced) to refer to the individual elements of $\mathbf{v}$.

Definition (Fundamental vector operations)

- Vector addition: if $\mathbf{u}$ and $\mathbf{v}$ are vectors in $\mathbb{R}^n$, then so is $\mathbf{w} = \mathbf{u} + \mathbf{v}$ (where we define $w_i = u_i + v_i$).
- Scalar multiplication: if $\mathbf{v}$ is a vector in $\mathbb{R}^n$, then so is $\mathbf{w} = c\mathbf{v}$ for any $c \in \mathbb{R}$ (we define $w_i = cv_i$).
- Scalar (dot) product: if $\mathbf{u}$ and $\mathbf{v}$ are vectors in $\mathbb{R}^n$, we define the scalar or dot product as:
$$\mathbf{u} \cdot \mathbf{v} = \sum_{i=1}^{n} u_i v_i$$
- Vector norm (or magnitude, or length): if $\mathbf{v}$ is a vector in $\mathbb{R}^n$, then we define the norm or length of $\mathbf{v}$ as:
$$||\mathbf{u}|| = \sqrt{\mathbf{u} \cdot \mathbf{u}}$$

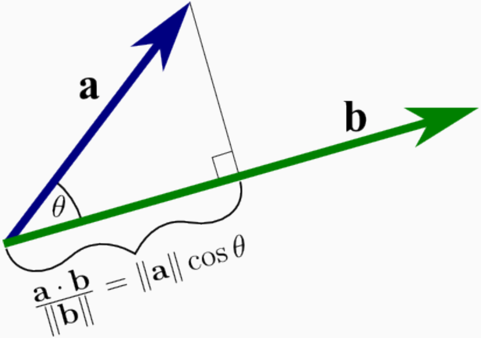
Visualizing vectors (in 2D)

- Vector addition is easy to interpret in 2D:



Visualizing the dot product

- The scalar or dot product is related to the directions and magnitudes of the two vectors:



- In fact, it is easy to recover the cosine between any two vectors.
- Note that these properties generalize to any number of dimensions.
- Question: how can we test it two vectors are perpendicular (orthogonal)?

Formalizing intuition

Definition (Bilinear Map)

A function $\Omega : V \times V \to \mathbb{R}$ is a *bilinear map* from vector space V to $\mathbb{R}$ iff:

$$\begin{aligned} \Omega(\lambda \mathbf{x} + \psi \mathbf{y}, \mathbf{z}) &= \lambda \Omega(\mathbf{x}, \mathbf{z}) + \psi \Omega(\mathbf{y}, \mathbf{z}) \\ \Omega(\mathbf{x}, \lambda \mathbf{y} + \psi \mathbf{z}) &= \lambda \Omega(\mathbf{x}, \mathbf{y}) + \psi \Omega(\mathbf{x}, \mathbf{z}) \end{aligned}$$

for any $\mathbf{x}, \mathbf{y}, \mathbf{z} \in V$.

- Ω is called symmetric if $\Omega(\mathbf{x}, \mathbf{y}) = \Omega(\mathbf{y}, \mathbf{x})$ for all $\mathbf{x}, \mathbf{y} \in V$.
- Ω is called positive definite if:

$$\Omega(\mathbf{x}, \mathbf{x}) \geq 0 \text{ for all } \mathbf{x}, \text{ and } \Omega(\mathbf{x}, \mathbf{x}) = 0 \text{ iff } \mathbf{x} = \mathbf{0}$$

Formalizing intuition

Definition (Inner Product and Inner Product Space)

Let V be any vector space and $\Omega : V \times V \to \mathbb{R}$ any bilinear map from V to $\mathbb{R}$. Then:

- If Ω is symmetric and positive definite, Ω is called an inner product on V. We usually write $\langle \mathbf{x}, \mathbf{y} \rangle$ instead of $\Omega(\mathbf{x}, \mathbf{y})$.
- The pair (V, Ω) (or $(V, \langle \cdot, \cdot \rangle)$) for inner product Ω is called an inner product space or vector space with inner product. If $\Omega(\mathbf{x}, \mathbf{y}) = \mathbf{x}^T \mathbf{y}$, (V, Ω) is called a Euclidean vector space.

Inner products allow us to formalize our geometrical intuitions about length, orthogonality, and distance.

Orthogonal projections

[ORTHOGONAL PROJECTION ON A SUBSPACE]

Matrices: basics

- A matrix arranges numbers into rows and columns, like this:

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

- Note that matrices are generally named as a capital, boldface letter. We refer to the elements of the matrix using the lower case equivalent with a subscript row and column indicator:

$$\mathbf{A} = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \end{bmatrix}$$

- Here we say that $\mathbf{A}$ is a matrix of size 2×3.
- Equivalently: $\mathbf{A} \in \mathbb{R}^{2 \times 3}$.

Matrices: arithmetic operations

- Matrices support common arithmetic operations:
- To add two matrices of the same size together, just add the corresponding elements in each matrix:

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} + \begin{bmatrix} 6 & 5 & 4 \\ 3 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 7 & 7 & 7 \\ 7 & 7 & 7 \end{bmatrix}$$

- Each matrix has two rows of three columns (so we describe them as 2×3 matrices).
- Adding matrices $\mathbf{A} + \mathbf{B}$ results in a new matrix $\mathbf{C}$ where $c_{i,j} = a_{i,j} + b_{i,j}$.
- This *elementwise* definition generalizes to subtraction, multiplication and division.

Matrices: arithmetic operations (continued)

- In the previous examples, we were able to add and subtract the matrices, because the operands (the matrices we are operating on) are conformable for the specific operation (in this case, addition or subtraction).
- To be conformable for addition and subtraction, the operands must have the same number of rows and columns
- There are different conformability requirements for other operations, such as matrix multiplication.

Matrices: unary arithmetic operations

- The negation of a matrix is just a matrix with the sign of each element reversed:

$$C = \begin{bmatrix} -5 & -3 & -1 \\ 1 & 3 & 5 \end{bmatrix}$$

$$-C = \begin{bmatrix} 5 & 3 & 1 \\ -1 & -3 & -5 \end{bmatrix}$$

- The transpose of a matrix switches the orientation of its rows and columns.
- You indicate this with a superscript T, like this:

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}^T = \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}$$

Matrices: matrix multiplication

- Multiplying matrices is a little more complex than the elementwise arithmetic we have seen so far.
- There are two cases to consider, scalar multiplication (multiplying a matrix by a single number)

$$2 \times \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} = \begin{bmatrix} 2 & 4 & 6 \\ 8 & 10 & 12 \end{bmatrix}$$

- And dot product matrix multiplication:

$$\mathbf{AB} = \mathbf{C}, \text{ where } c_{i,j} = \sum_{k=1}^{n} a_{i,k} b_{k,j}$$

- What can we infer about the conformable sizes of **A** and **B**? What is the size of **C**.

Matrices: multiplication is just dot products

- To multiply two matrices, we are really calculating the dot product of rows and columns.
- We perform this operation by applying the RC rule - always multiplying (dotting) Rows by Columns.
- For this to work, the number of columns in the first matrix must be the same as the number of rows in the second matrix so that the matrices are conformable.
- An example:

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \cdot \begin{bmatrix} 9 & 8 \\ 7 & 6 \\ 5 & 4 \end{bmatrix} = \begin{bmatrix} ? & ? \\ ? & ? \end{bmatrix}$$

Matrices: inverses

- The identity matrix $\mathbf{I}$ is a square matrix with all ones on the diagonal, and zeros everywhere else.
- So, $\mathbf{IA} = \mathbf{AI}$, and $\mathbf{Iv} = \mathbf{v}$.
- The inverse of a square matrix $\mathbf{A}$ is denoted $\mathbf{A}^{-1}$.
- $\mathbf{A}^{-1}$ is the unique (if it exists) matrix such that:

$$\mathbf{A}^{-1}\mathbf{A} = \mathbf{A}\mathbf{A}^{-1} = \mathbf{I}$$

Matrices: solving systems of equations

- We can now use this to our advantage:

$$\begin{bmatrix} 67.9 & 1.0 \\ 61.9 & 1.0 \end{bmatrix} \begin{bmatrix} m \\ b \end{bmatrix} = \begin{bmatrix} 170.85 \\ 122.50 \end{bmatrix}$$

- Multiplying both sides by the inverse:

$$\begin{bmatrix} 67.9 & 1.0 \\ 61.9 & 1.0 \end{bmatrix}^{-1} \begin{bmatrix} 67.9 & 1.0 \\ 61.9 & 1.0 \end{bmatrix} \begin{bmatrix} m \\ b \end{bmatrix} = \begin{bmatrix} 67.9 & 1.0 \\ 61.9 & 1.0 \end{bmatrix}^{-1} \begin{bmatrix} 170.85 \\ 122.50 \end{bmatrix}$$

- And we have:

$$\mathbf{I} \begin{bmatrix} m \\ b \end{bmatrix} = \begin{bmatrix} m \\ b \end{bmatrix} = \begin{bmatrix} 67.9 & 1.0 \\ 61.9 & 1.0 \end{bmatrix}^{-1} \begin{bmatrix} 170.85 \\ 122.50 \end{bmatrix}$$

Matrices: linear versus affine

- Matrix multiplication computes linear transformations of vector spaces.
- We are also interested in affine transformations that don't necessarily preserve the origin:
- An affine transformation is a linear transformation followed by a translation:

$$f(\mathbf{x}) = \mathbf{A}\mathbf{x} + \mathbf{b}$$

- Note: an affine transformation in n dimensions can be modeled by a linear transformation in $n+1$ dimensions.

Tensors: A general structure for dense data

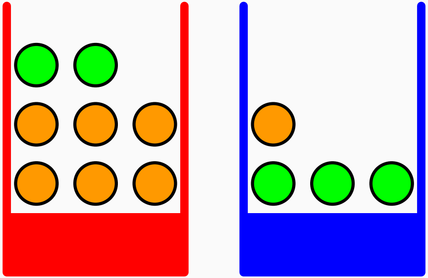
- There is nothing magic about one dimension (vectors) or two dimensions (matrices).
- In fact, the tools we use are completely generic in that we can define dense, homogeneous arrays of numeric data of any dimensionality.
- The generic term for this is a tensor, and all of the math generalizes to arbitrary dimensions.
- Example: a color image is naturally modeled as a tensor in three dimensions (two spatial, one chromatic).
- Example: a batch of b color images of size 32×32 is easily modeled by simply adding a new dimension: $\mathbf{B} \in \mathbb{R}^{b \times 32 \times 32 \times 3}$.

Mathematical Preliminaries: Probability and Statistics

Probability theory: A motivating example

We consider the classical Urn example:

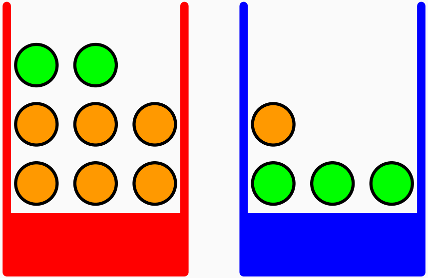
- We have two urns (red and blue) whose exact contents are unknown, but which contain pieces of fruit (apples and oranges).
- One draws a piece of fruit from an urn chosen at random, say with probability 0.4 and 0.6.



Probability theory: A motivating example

We consider the classical Urn (box) example:

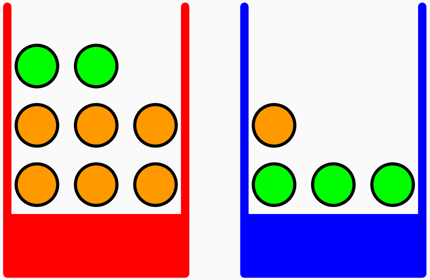
- In this case we have a random variable B (box) defined by its distribution: $p(B = \text{blue}) = 0.6$ and $p(B = \text{red}) = 0.4$
- We also have a random variable F (fruit) which is dependent on B – dependent because its distribution depends on the box chosen.



Probability theory: A motivating example

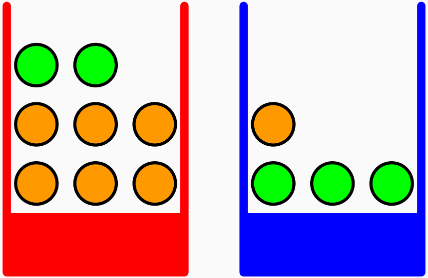
We consider the classical Urn (box) example:

- What is the overall probability of choosing an apple? (i.e. $p(F = \text{apple})$) – a probability that clearly also depends on $p(B)$.
- If the fruit chosen is an orange, what is the probability it came from the blue box (i.e. $p(B = \text{blue} | F = \textit{orange})$)



The joint distribution of two random variables:

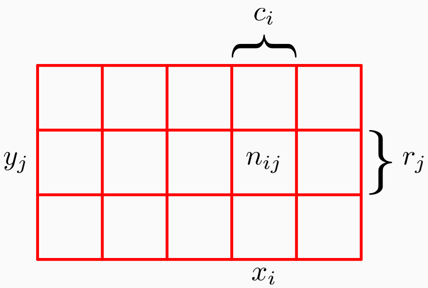
- The key to analyzing such questions is the joint probability distribution of all variables involved.
- We will derive the sum and product rules of probability theory (which are probably closer to the $F = ma$ and $V = IR$ of ML).



The general case

Let's consider a the general case:

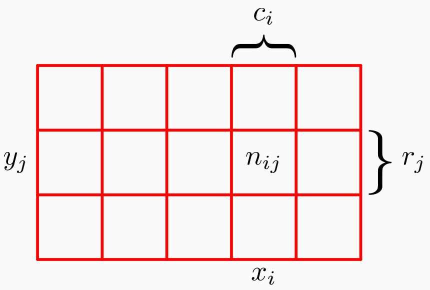
- How do we estimate the joint distribution of random variables X and Y (without any prior knowledge)?
- We draw a sample: $\{\, (x_i, y_i) \mid i = 1, 2, \ldots N \,\}$ independently drawn from the joint distribution $p(X, Y)$ and make a histogram.



The general case

Let's consider the general case:

- Define n_{ij} to be the number of samples falling in cell (i, j) – that is the cell corresponding to (x_i, y_j) – of the histogram.
- Also: c_i will be the total number of times X takes the value x_i and r_j the total number of times Y takes value y_j



The general case

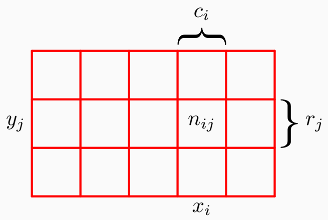
This histogram captures (well, estimates) everything we need:

- The joint probability $p(X, Y)$ is:

$$p(X = x_i, Y = y_j) = \frac{n_{ij}}{N}$$

- The marginal probability of X taking value x_i is:

$$p(X = x_i) = \frac{c_i}{N} = \sum_j p(X = x_i, Y = x_j)$$



The general case

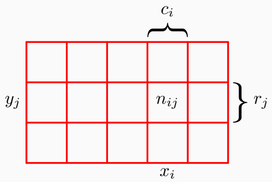
Now, let's see how to condition probabilities:

- Look at only those joint events for which $X = x_i$.
- We write the fraction of such events for which $Y = y_j$ as:

$$p(Y = y_j | X = x_i) = \frac{n_{ij}}{c_i}$$

- We can derive this from the joint probability:

$$p(X = x_i, Y = y_j) = \frac{n_{ij}}{N} = \frac{n_{ij}}{c_i}\frac{c_i}{N} = p(Y = y_j | X = x_i) p(X = x_i)$$



Probability Theory for Machine Learning

- We will frequently invoke the two rules of probability:

$$\begin{aligned} \text{sum rule:} \quad p(X) &= \textstyle\sum_Y p(X, Y) \\ \text{product rule:} \quad p(X, Y) &= p(Y \mid X)p(X) \end{aligned}$$

- And we will make frequent use of Bayes' rule:

$$p(Y \mid X) = \frac{p(X \mid Y)p(Y)}{p(X)}$$

- This takes on special significance when applied it to parameter inference:

$$\begin{aligned} p(\mathbf{w} \mid \mathcal{D}) &= \frac{p(\mathcal{D} \mid \mathbf{w})p(\mathbf{w})}{p(\mathcal{D})} \\ \text{posterior} &\propto \text{data likelihood} \times \text{prior} \end{aligned}$$

- An important operation using probabilities is finding weighted averages of functions:

$$\mathbb{E}[f] = \sum_x p(x)f(x) \quad (\text{or} \int p(x)f(x)dx)$$

- In either case, if we have a finite sample of N points from the distribution $p(x)$ we can approximate the expectation:

$$\mathbb{E}[f] \approx \sum_i p(x_i)f(x_i)$$

- The Gaussian distribution will be our friend, so covariances are important:

$$\begin{aligned} \text{cov}(\mathbf{x}, \mathbf{x}) &= \mathbb{E}_{\mathbf{x}}[\{\mathbf{x} - \mathbb{E}[\mathbf{x}]\}\{\mathbf{x}^T - \mathbb{E}[\mathbf{x}^T]\}] \\ &= \mathbb{E}_{\mathbf{x}}[\mathbf{x}\mathbf{x}^T] - \mathbb{E}[\mathbf{x}]\mathbb{E}[\mathbf{x}^T] \end{aligned}$$

The Gaussian distribution (speaking of covariance)

- The univariate Gaussian distribution is super important:

$$\mathcal{N}(x \mid \mu, \sigma) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\{-\frac{1}{2\sigma^2}(x-\mu)^2\}$$

- As is the multivariate Gaussian distribution, which we will use extensively:

$$\mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\boldsymbol{\Sigma}|^{1/2}} \exp\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\}$$

- Here $\boldsymbol{\mu}$ is a D dimensional vector (the mean) and $\boldsymbol{\Sigma}$ is the is the $D \times D$ covariance matrix.

Decision theory and supervised classification

- Let's try to expand our growing intuition to include classification problems.
- Probability theory gives us a principled way to represent and quantify uncertainty, so let's use it!
- Suppose we have an input $\mathbf{x}$ together with a vector $\mathbf{y}$ of target variables.
- For regression problems, $\mathbf{y}$ will be continuous variables, where for classification problems it will represent class labels.
- The joint distribution $p(\mathbf{x}, \mathbf{y})$ gives us a complete picture of the uncertainty associated with these variables.

Decision theory and supervised classification

- As an example, let's assume $\mathbf{x}$ is a 512×512 pixel X-ray of patient and we want to decide if the patient has cancer:

$$f(\mathbf{x}) = \begin{cases} 0 & \text{if patient has cancer} \\ 1 & \text{otherwise} \end{cases}$$

- What might our dataset look like? Well, probably a set of pairs:

$$\mathcal{D} = \{ (\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_N, y_N) \}$$

- We must first address the inference problem: determine the joint distribution $p(\mathbf{x}, y)$ (usually extremely hard).
- Then we must decide how to act optimally for a specific $p(\mathbf{x}', y)$ (often very easy).

Decision theory and supervised classification

- So, when we obtain an image $\mathbf{x}$, our goal is to decide which of the two classes it belongs to.
- We can derive information about this decision from the posterior distribution:

$$
\begin{aligned}
p(\mathcal{C}_k \mid \mathbf{x}) &= \frac{p(\mathbf{x} \mid \mathcal{C}_k) p(\mathcal{C}_k)}{p(\mathbf{x})} \\
&= \frac{\text{data likelihood} \times \text{prior}}{\text{evidence}}
\end{aligned}
$$

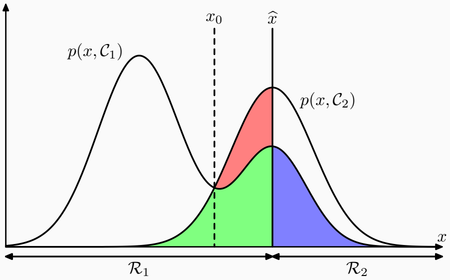
- But, the salient question remains: how do we decide?

Decision theory and supervised classification

The theoretical optimal decision:

- Minimize the expected misclassification rate.

$$
\begin{aligned}
p(\text{misclass}) &= p(\mathbf{x} \in \mathcal{R}_1, \mathcal{C}_2) + p(\mathbf{x} \in \mathcal{R}_2, \mathcal{C}_1) \\
&= \int_{\mathcal{R}_1} p(\mathbf{x}, \mathcal{C}_2) d\mathbf{x} + \int_{\mathcal{R}_2} p(\mathbf{x}, \mathcal{C}_1) d\mathbf{x}
\end{aligned}
$$

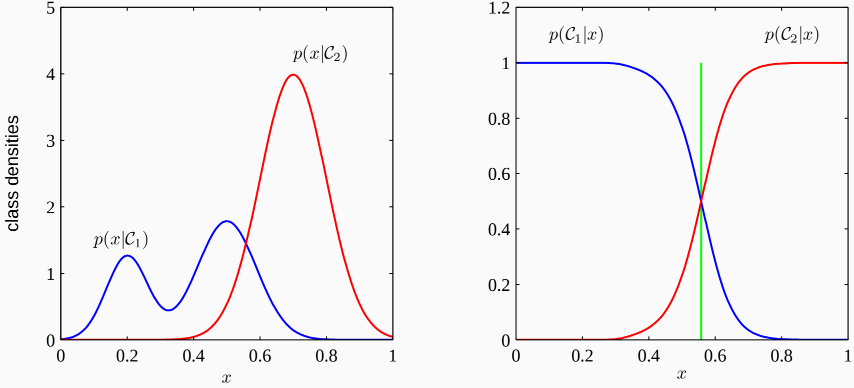


- Option 1: estimate the class-conditional densities $p(\mathbf{x}|\mathcal{C}_k)$ individually, along with prior probabilities $p(\mathcal{C}_k)$, then use Bayes theorem to compute the posterior.

$$p(\mathcal{C}_k|\mathbf{x}) = \frac{p(\mathbf{x}|\mathcal{C}_k)p(\mathcal{C}_k)}{p(\mathbf{x})}$$

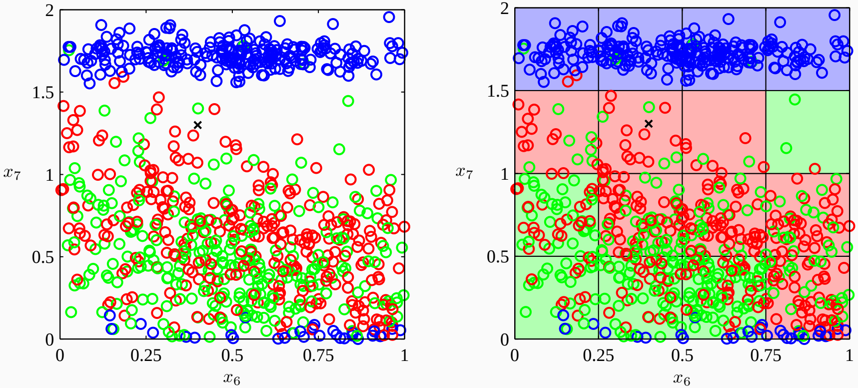
- Option 2: directly estimate the posterior probabilities.
- Option 3: skip all the Bayesian mumbo jumbo and directly estimate a discriminant function.

There are practical reasons for choosing an approach:



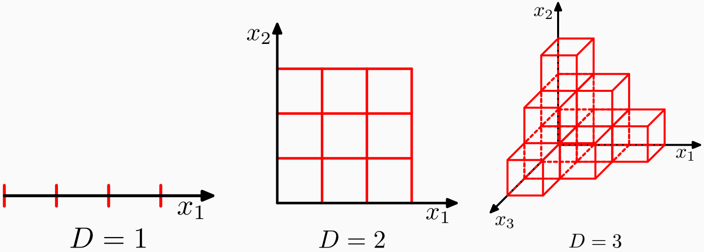
The Curse of Dimensionality

- Consider a 3-class classification problem with a measly two input dimensions:



The Curse of Dimensionality

- As we add input dimensions, the number of bins in any discretization of the space grows exponentially.
- The moral: enriching input (by adding dimensions) doe not make our problem easier.



Notational Alignment and the Way Forward

Notational Alignment

- Vectors will be denoted in lower-case, roman, boldface font: $\mathbf{x}$.
- All vectors are assumed to be column vectors.
- Uppercase, bold roman letters denote matrices: $\mathbf{M}$.
- The vector and matrix transpose is indicated by the superscript T: $\mathbf{x}^T, \mathbf{M}^T$.
- The notation $(w_1, \ldots, w_n)$ denotes a row vector of n dimensions.
- The corresponding column vector is written as $\mathbf{w} = (w_1, \ldots, w_n)^T$.
- The expectation of $f(x, y)$ with respect to a r.v. x is written as $\mathbb{E}_x[f(x, y)]$.
- If x is conditioned on z, the conditional expectation is $\mathbb{E}_x[f(x) \mid z]$
- The variance of $f(x)$ is denoted by $\mathrm{var}[f(x)]$, and the covariance as $\mathrm{cov}[\mathbf{x}, \mathbf{y}]$.

The way forward

- In this lecture we saw a brief and high level overview of some of the basic concepts of linear algebra and probability theory.
- This is just enough theory to get us started on our Machine Learning journey.
- We will, as needed, introduce more advanced concepts as we proceed (e.g. gradient-based optimization, special properties of the Gaussian density, Hilbert spaces, etc).
- Up next:
  - We will dive into a study of linear models for regression.
  - We will see how to model continuous output predictions using linear functions of the input.
  - We will see how to fit these models, how non-linear basis projections can enrich them, and how to quantify belief in their predictions.

Homework and Reading

Homework and Reading

Reading Assignment:

- Bishop: Chapters 1 and 2 (1.5, 2.3)

Homework:

1. A linear map $\pi : V \to U$ from vector space V to vector space U is called a projection if $\pi \odot \pi = \pi$ (i.e. π is idempotent). Prove that the orthogonal projection from V onto a subspace U is indeed a projection.
2. Show that the mode of a Gaussian distribution $\mathcal{N}(\mu, \sigma)$ is given by μ.