1 Course Motivation

This course dwells on the foundational algorithmic and computational aspects of data sciences, machine learning and statistical inference analysis. We shall consider algorithms that automatically "learns" progressively to correctly process input data, i.e it learns from its successes and mistakes. An example is a learning algorithm that is used for processing and filtering spam mail. We shall also consider data science and machine learning algorithms which require a fusion of input modalities to make correct decisions, namely tasks that enable autonomous driving, image or scene recognition and understanding. There is then the challenge of analyzing and inferring big data from genomic expression levels, to cosmic data interpretation.

2 Course Outline

In this course, we will also be covering a range of foundational topics based at the core of algorithmic and computational science and extremely useful for developing effective machine learning solutions . These shall span

- Probability Statistics (discrete and continuous)
- Geometry and topology (algebraic and differential)
- Algebra (linear and polynomial)
- Approximation theory (regression and recovery)
- Sampling theory and representation theory (uniformity and discrepancy)
- Functional analysis (vector and Hilbert spaces)
- Optimization (convex and non-convex)

In the first few lectures, we will talk about geometry and probability in high-dimensional space. The geometry of data distributions in $\mathbb{R}^2$, $\mathbb{R}^3$ and $\mathbb{R}^4$ can already be different, and it will get completely different when the dimension is $10,000$. So, it is important to understand this geometry in each of the dimensions. We need to know where to look and where to sample from, and in the presence of noise, Here there is the probability of determining data outliers and for the basis of predictions with provable guarantees. We will also need linear algebra and multi-linear algebra and some other non-linear (modern) algebra. Approximation theory, will teach us basics about splines, non-linear kernels, their representations and use in regression, classification etc. Sampling theory will teach us about adaptation and dimension reduction. Functional analysis gives us the basic foundations of optimization theory, namely in characterizing and simplifying objective functions and the constraint spaces of our search for optimality.

3 Computational Science Preliminaries

3.1 Splines

Most data are noisy. We want to have stable methods to fit the data. Two possibles ways of finding the best fit are minimizing the least square distance or maximizing the margin. Additionally, one can use polynomial splines. A polygonal path is a C^0-continuous spline (Figure 1(a)). To get a C^1-continuous spline, the tangent vector must vary continuously. A polygonal path that is C^1-continuous must therefore be a trivial straight line segment with points evenly spaced (Figure 1(b)).

We can obtain a C^2-continuous curve by using cubic splines (Figure 1(c)) for an interpolating spline. In addition to *parametric continuity* C^k where $k \in \{-1, 0, 1, 2, \ldots\}$, there is the corresponding notion of *geometric continuity* G^k. A continuous piecewise-defined spline curve $\mathbf{c}(t)$ is said to be G^1-continuous if at each joint point i there exists some $\alpha \neq 0$ such that

$$\frac{\mathrm{d}\,\mathbf{c}_i(t)}{\mathrm{d}\,t} = \alpha \frac{\mathrm{d}\,\mathbf{c}_{i+1}(t)}{\mathrm{d}\,t},$$

with $\mathbf{c}_i$ and $\mathbf{c}_{i+1}$ being the two spline pieces meeting at the joint point.

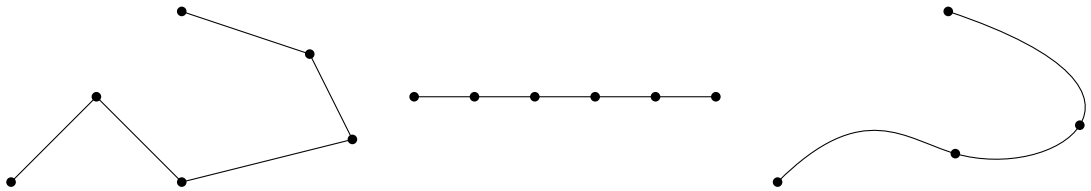


Figure 1: (a) A C^0-continuous polygonal path. (b) A C^1-continuous polygonal path. (c) A C^2-continuous cubic spline.

3.2 Sampling, Approximation and Optimization

In many big data applications, we need to perform sampling. For example, the problem of aligning two images in $\mathbb{R}^2$ can be formulated as an optimization problem over the solution space of all possible translations. Then, we can discretize the solution space by uniformly sampling a set S of translations.

Sampling a translation in $\mathbb{R}^2$ is like sampling a point (t_x, t_y) from a rectangle. The point (t_x, t_y) represents the translation

$$(x, y) \mapsto (x' = x + t_x, y' = y + t_y).$$

We would like to solve some optimize certain objective function F approximately, in the sense that

$$\min_{(t_x,t_y)\in S} \left| F(t_x, t_y) - F(t_x^*, t_y^*) \right| < \varepsilon.$$

It can be too expensive or too hard to be solved exactly. Approximation sometimes also suffices when the data itself has uncertainty. For example, if $O(n^3)$ time is required to solve a problem exactly, can we solve it in $O(n/\varepsilon^2)$ time?

The generic form of an (analytic) optimization problem is the following

$$\min_{x\in R^p} f(x)$$

$$s.t.\ x \epsilon C,$$

where x is the variable of the problem, $f : R^p \to R$ is the objective function of the problem, and $C \subset R^p$ is the constraint set of the problem and defines the search space for the optimization. When used in a machine learning setting, the objective function allows the algorithm designer to encode proper and expected behavior for the machine learning model, such as fitting well to training data with respect to some loss function, whereas the constraint allows restrictions on the model to be encoded, for instance, restrictions on model size.

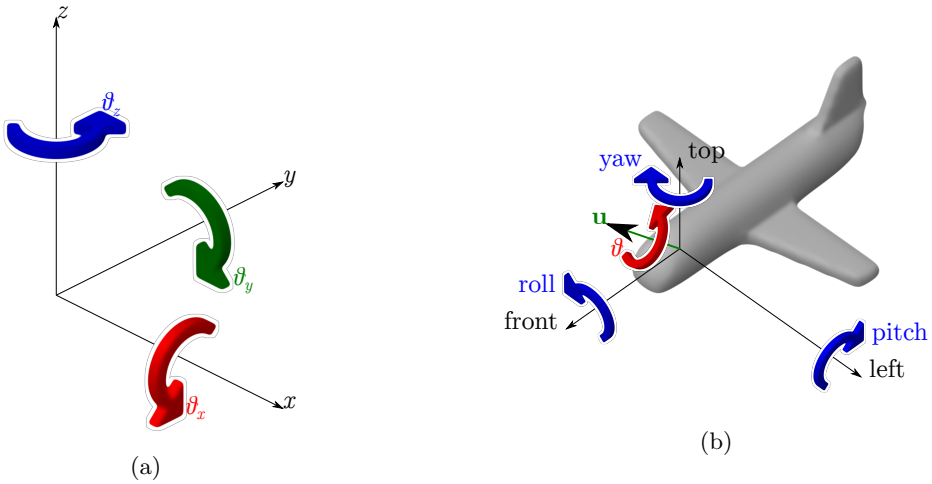
3.3 Representation Theory

As we have discussed, a translation in $\mathbb{R}^2$ is like a point in a rectangle. In $\mathbb{R}^3$ or $\mathbb{R}^d$, it becomes a 3-dimensional cuboid and a d-dimensional cuboid respectively. Actually, to represent such translation vector in $\mathbb{R}^d$, we can use any basis of $\mathbb{R}^d$.

What about rotations (around the origin)? In $\mathbb{R}^2$, the space of all rotations is 1-dimensional (as a manifold). For example, anti-clockwise rotation by angle θ can be represented as the matrix

$$\begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix}.$$

In $\mathbb{R}^3$, the space of all rotations becomes 3-dimensional. We can see this by using the *Euler angles* (Figure 2(a)) or the yaw-pitch-roll axes used by aircraft. How can we represent such rotation? And what is the geometry of this space? One way is to represent a rotation as an 3×3 orthogonal matrix analogous to the



(a) (b)

Figure 2: (a) The three Euler angles $\theta_x, \theta_y, \theta_z$ around the three coordinate axes x, y, z respectively. (b) An angle θ of rotation around the axis rotation $\mathbf{u}$, and the yaw-pitch-roll angles.

$\mathbb{R}^2$ case. In general, to specify a 3×3 matrix requires 9 parameters, which over-parametrizes rotations. We would like to get a minimal representation. This is rotation in world coordinates (control tower coordinate system).

Another way to represent a rotation in $\mathbb{R}^3$ is by using a quaternion $\mathbf{q}$ to represent the pair $(\theta, \mathbf{u})$, where θ is the angle of rotation around the unit vector $\mathbf{u} \in \mathbb{R}^3$ (Figure 2(b)). The rotation $\mathbf{q}$ is decomposable into local yaw-pitch-roll rotations (cockpit view). The quaternion algebra is non-commutative, because in composing rotations, the order of composition matters.

3.4 Polynomial

A 3×3 matrix is *orthogonal* if it satisfies certain orthogonal constraint. Notice that such constraints can actually be expressed using polynomials. The generic form of a degree-1 polynomial equation in $\mathbb{R}^2$ can be written as

$$a_{10}x_1 + a_{01}x_2 + a_{00} = 0,$$

which represent a line. In $\mathbb{R}^3$, a degree-1 polynomial equation represents a plane. A system of two such equations will define a line (the intersection of two planes) as long as they are not independent.

Similarly, we can generalize and talk about system of polynomials. A degree-2 polynomial equation in $\mathbb{R}^2$ has the form

$$a_{20}x_1^2 + a_{11}x_1x_2 + a_{02}x_2^2 + a_{10}x_1 + a_{01}x_2 + a_{00} = 0,$$

while a degree-0 polynomial equation degenerates to $a_{00} = 0$. In general, a degree n polynomial is given by

$$\mathbf{x} = (x_1, \ldots, x_d) \in \mathbb{R}^d \mapsto \sum_{i_1 + \ldots i_d \leq n} \mathbf{a_i x^i} \in \mathbb{R}.$$

Analogous to linear equations, a polynomial equation gives rise to a surface and the intersection of two surfaces may be a curve. Two plane curves of degree d_1 and d_2, by Bézout Theorem with some caveats, intersects at $(d_1 \cdot d_2)$ points.

Techniques using semi-definite programming (SDP) are useful for polynomial optimization, just like linear programming (LP) is used to solve linear optimization on polytopes. We call a polynomial constraint *algebraic* if it is an equality, and *semi-algebraic* if it is an inequality.

The set of polynomials, just like the set of integers, form a algebraic structure called *ring*. A ratio of two polynomials is called a *rational function*, just like the ratio of two integers is called a rational number.

3.5 Probability in High-Dimensional Space

We all know how to calculate the area of a circle, or the volume of a sphere. How about a sphere in a high-dimensional space? Let $B_d = \{(x_1, x_2, \ldots, x_d) \,|\, x_1^2 + x_2^2 + \ldots + x_d^2 \leq 1\}$ be the unit ball in $\mathbb{R}^d$. (The boundary of B_d is the $(d-1)$-sphere.) What is the volume of B_d as $d \to \infty$? It is surprising that $\text{vol}(B_d) \to 0$. Moreover, the volume is concentrated around the equator.

The study of the geometry of spheres in high dimension is linked with probability. This is because level sets of isotropic Gaussian distribution form spheres, and Gaussian is the limiting distribution in the Central Limit Theorem. In fact, points randomly sampled from a sphere or a spherical Gaussian distribution all have essentially the same pairwise distances for large d. This is related to the Law of Large Numbers or Chernoff Bounds.

4 Applications

There are many and growing. See secondary notes, where I highlight several of them and from different data domains.

Bibliographics Notes

See [BHK] for references on several intertwined topics in data sciences, [SS] for the theory and algorithms in machine learning, [B2] for brief introduction on polynomial equations, and functional analysis, [JK] for references on optimization, [M1] on topics in discrete computational geometry, and [M2] for sampling and discrepancy .

References

[B2] Chandrajit Bajaj *A Mathematical Primer for Computational Data Sciences* https://www.cs.utexas.edu/ bajaj/math-ds.pdf

[BHK] Avrim Blum, John Hopcroft, Ravindran Kannan, . *Foundations of Data Science.* https://www.cs.cornell.edu/jeh/book.pdf

[JK] Prateek Jain, Purshottam Kar *Non-convex Optimization for Machine Learning* ArXiv: 1712.07897

[M1] Jiri Matousek *Lectures in Discrete Geometry*, Springer Verlag. Short version: https://kam.mff.cuni.cz/ uli/aspr-short.pdf

[M2] Jiri Matousek *Geometric Discrepancy: Illustrated Guide*, Springer Verlag.

[SS] Shai Shalev-Shwartz and Shai Ben-David *Understanding Machine Learning: From Theory to Algorithms.* http://www.cs.huji.ac.il/ shais/UnderstandingMachineLearning