Definitions and notations

Zhehao Wang

November 30, 2021

1 Calculus and linear algebra

1.1 Limit

Let $f(x)$ be a function defined on an interval that contains $x = a$, except possibly at $x = a$, then we say that

$$\lim_{x \to a} f(x) = L$$

if for every $\epsilon > 0$ there is some number $\delta > 0$ such that

$$|f(x) - L| < \epsilon \text{ whenever } 0 < |x - a| < \delta$$

1.2 Gradient

Given $f(\vec{x})$ where $\vec{x} = (x_1, \ldots, x_n)$ on $\mathbf{R}^n$

$$\bigtriangledown f(a_1, \ldots, a_n) = (\frac{\partial f}{\partial x_1}(a_1, \ldots, a_n), \ldots, \frac{\partial f}{\partial x_n}(a_1, \ldots, a_n))$$

Intuition: gradient is a vector (the rate of change of your function, when you move in a certain direction), which in a two-dimensional space, tangents the curve at a given point.

1.3 Directional derivative

$$\bigtriangledown_v f(\vec{x}) = \lim_{h \to 0} \frac{f(\vec{x} + h\vec{v}) - f(\vec{x})}{h}$$

Intuition: the rate-of-change of a function $f(\vec{x})$ on direction $\vec{v}$. Remember that this is a scalar (since we are given the direction).

$$f'(x; u) = \bigtriangledown f(x)^T u$$

Meaning gradient on a certain direction vector u is $f(x)$'s directional derivative on u.

1.4 Taylor expansion

Every infinite differentiable function can be approximated by a series of polynomials.
Let $f(x)$ be a real function which is continuous on the closed interval $[a \dots b]$ and $n+1$ times differentiable on the open interval $(a \dots b)$, let $\xi \in (a \dots b)$.
Then given any $x \in (a \dots b)$, there exists some $\eta \in \mathbb{R} : x \leq \eta \leq \xi$ or $\xi \leq \eta \leq x$ such that:

$$f(x) = \sum_{k=0}^{n} \frac{1}{k!}(x-\xi)^k f^{(k)}(\xi) + R_n$$

Where $R_n = \frac{1}{(n+1)!}(x-\xi)^{n+1} f^{(n+1)}(\eta)$ is the error term.
The expression

$$f(x) = \sum_{n=0}^{\infty} \frac{1}{n!}(x-\xi)^n f^{(n)}(\xi)$$

where n is taken to the limit is known as the Taylor series expansion of f about ξ.

1.5 Vector norm

A norm is a function that assigns a stricly positive length or size to each vector in a vector space (except the zero vector which is assigned a length of 0).
Absolute value norm is a norm on the one-dimensional vector spaces formed by real or complex numbers.

$$\|x\| = |x|$$

Euclidean norm on a Euclidean space $\mathbf{R}^n$ is such

$$\|\vec{x}\|_2 = \sqrt{x_1^2 + \cdots + x_n^2}$$

Manhattan or taxicab norm

$$\|\vec{x}\|_1 = \sum_{i=1}^{n} |x_i|$$

p-norm

$$\|\vec{x}\|_p = (\sum_{i=1}^{n} |x_i|^p)^{\frac{1}{p}}$$

Note that when $p = 1$, we get Manhattan norm, and when $p = 2$, we get Euclidean norm.
When $p = \infty$

$$\|\vec{x}\|_\infty = \max_i |x_i|$$

1.6 argmax

Points of the domain of some function at which the function values are maximized.
Given an arbitrary set X, a totally ordered set Y and a function $f : X \to Y$, the arg max over some subset S of X is defined by

$$\arg\max_{x \in S \subseteq X} f(x) = \{x \mid x \in S \wedge \forall y \in S : f(y) \leq f(x)\}$$

1.7 Dot product, inner product

Maps two equal-size vectors to a real value.

$$\cdot : \mathbf{R}^d \times \mathbf{R}^d \to \mathbf{R}$$

Defined geometrically, given two vectors a and b,

$$a \cdot b = \|a\|\|b\|cos(\theta)$$

where θ is the angle between a and b.
Defined algebraically,

$$a \cdot b = a^T b$$

1.8 Inner product space / pre-Hilbert space, Hilbert space

An **inner product space** (over reals) is a vector space $\mathbb{V}$ and an **inner product**, which is a mapping

$$\langle ., . \rangle : \mathbb{V} \times \mathbb{V} \to \mathbf{R}$$

This space has the properties that $\forall x, y, z \in \mathbb{V}$ and $a, b \in \mathbf{R}$,

- symmetry: $\langle x, y \rangle = \langle y, x \rangle$
- linearity: $\langle ax + by, z \rangle = a\langle xz \rangle + b\langle y, z \rangle$
- positive-definitiveness: $\langle x, x \rangle \geq 0$. And $\langle x, x \rangle = 0 \iff x = 0$

A **norm** on the inner product space can then be defined as

$$\|x\| = \sqrt{\langle x, x \rangle}$$

Parallelogram law states that a norm can be written in terms of an inner product on $\mathbb{V}$ iff $\forall x, x' \in \mathbb{V}$,

$$2\|x\|^2 + 2\|x'\|^2 = \|x + x'\|^2 + \|x - x'\|^2$$

And if it can be, the inner product is given by the **polarization identity**

$$\langle x, x' \rangle = \frac{\|x\|^2 + \|x'\|^2 - \|x - x'\|^2}{2}$$

Pythagorean theorem states that two vectors are **orthogonal** if $\langle x, x' \rangle = 0$, denoted as $x \perp x'$. And x is orthogonal to a set S if for all $s \in S$, x is orthogonal to s.
If $x \perp x'$, then $\|x + x'\|^2 = \|x\|^2 + \|x'\|^2$

Roughly, **projection onto a plane** can then be defined as for $x \in \mathbb{V}$, let M be a subspace of inner product space $\mathbb{V}$, then m_0 is the projection of x onto M if $m_0 \in M$ and is the closest point to x in M. Meaning $\forall m \in M$, $\|x - m_0\| \leq \|x - m\|$.
A Hilbert space is a complete* inner product space. E.g. $\mathbf{R}^d$ and the standard inner product. Any finite dimensional inner product space is a Hilbert space.
The **projection theorem** suggests, for a Hilbert space $\mathbf{H}$, M is a closed subspace of $\mathbf{H}$, for any $x \in \mathbf{H}$, there exists a unique $m_0 \in M$ for which

$$\|x - m_0\| \leq \|x - m\| \ \forall m \in M$$

This m_0 is called the orthogonal projection of x onto M. Furthermore, $m_0 \in M$ is the projection of x onto M iff $x - m_0 \perp M$.
Projection reduces norm: let M be a closed subspace of $\mathbf{H}$. For any $x \in \mathbf{H}$, let m_0 be the projection of x onto M, then $\|m_0\| \leq \|x\|$ with equality only when $m_0 = x$. †

1.9 Cauchy-Schwarz inequality

Given two vectors u and v

$$|\langle u, v \rangle|^2 \leq |\langle u, u \rangle| \cdot |\langle v, v \rangle|$$

Where $|\langle u, u \rangle|$ is the inner product. The equality holds only when u and v are linearly independent (parallel).

1.10 Jacobian matrix

Jacobian matrix is the matrix of all first-order partial derivatives of a vector-valued function.
Suppose $\vec{f} : \mathbf{R}^n \to \mathbf{R}^m$, the Jacobian matrix $\vec{J}$ of $\vec{f}$ is defined as follows

$$\vec{J} = \begin{bmatrix} \frac{\partial \vec{f}}{\partial x_1} & \cdots & \frac{\partial \vec{f}}{\partial x_n} \end{bmatrix}$$

or component-wise $\vec{J}_{ij} = \frac{\partial \vec{f}_i}{\partial x_j}$, meaning

*a space is complete if all Cauchy sequences in the space converge.
†think Pythagorean theorem on a $\mathbf{R}^2$ space

$$\vec{J} = \begin{bmatrix} \frac{\partial \vec{f_1}}{\partial x_1} & \cdots & \frac{\partial \vec{f_1}}{\partial x_n} \\ & \cdots & \\ \frac{\partial \vec{f_m}}{\partial x_1} & \cdots & \frac{\partial \vec{f_m}}{\partial x_n} \end{bmatrix}$$

1.11 Order of a matrix

1.12 Inversibility of a matrix

1.13 Eigenvalue, eigenvector

An **eigenvalue** of a **linear transformation** * is a non-zero vector that changes by only a scalar factor when that linear transformation is applied to it.
Meaning the set of $\vec{v}$ that satisfies

$$T(\vec{v}) = \lambda \vec{v}$$

Where T is the linear transformation, λ is a scalar and called **eigenvalue**.
We can compute the eigenvalues λ of a square matrix A of size n by solving this **determinant**

$$|A - \lambda I| = 0$$

Where I is the **identity matrix (unit matrix)** of size n.
For each eigenvalue λ, we can then solve its (corresponding set of) eigenvectors $\vec{v}$ using

$$(A - \lambda I) \cdot \vec{v} = 0$$

Intuition: applying a linear transformation on an eigenvector of that linear transformation yields a vector that is parallel to this eigenvector (multiplier: eigenvalue).

1.14 Positive-definite

A **symmetric** $n \times n$ real matrix is positive definite if the scalar $z^T M z$ is strictly positive for every non-zero column vector $z \in \mathbf{R}^n$.
The identity matrix I is positive definite.
For any real invertible matrix A, the product $A^T A$ is a positive definite matrix.

1.15 Convexity

A **set** C is **convex** if for any $x_1, x_2 \in C$ and any θ with $0 \leq \theta \leq 1$ we have

$$\theta x_1 + (1 - \theta) x_2 \in C$$

*Think square matrix

Intuition: for all x_1, x_2 in C, all the points on the line segment connecting points x_1, x_2 are in C.

A **function** $f : \mathbf{R}^n \to \mathbf{R}$ is **convex** if dom f is a convex set and if for all $x, y \in$ dom f, and $0 \leq \theta \leq 1$, we have

$$f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta)f(y)$$

Intuition: line segment connecting points x, y on the graph of f does not cross the graph of f.
Examples:

- $f(x) = ax + b$ is both convex and concave on $\mathbf{R}$ for all $a, b \in \mathbf{R}$.
- $f(x) = |x|^p$ for $p \geq 1$ is convex on $\mathbf{R}$.
- $f(x) = e^{ax}$ for all a is convex on $\mathbf{R}$.
- Every norm on $\mathbf{R}^n$ is convex.
- $f(x) = \max\{x\}$ is convex on $\mathbf{R}^n$.

A function f is **strictly convex** if the line segment connecting any two points on the graph of f lies strictly above the graph.

- When a function is convex, if there is a local minimum, then it is a global minimum.
- When a function is strictly convex, if there is a local minimum, then it is the unique global minimum.

For a multivariate twice differentiable function, $h : \Theta \subset \mathbb{R}^d \to \mathbb{R}$, $d \geq 2$, the gradient vector is

$$\bigtriangledown h(\theta) = \begin{pmatrix} \frac{\partial h}{\partial \theta_1}(\theta) \\ \cdots \\ \frac{\partial h}{\partial \theta_d}(\theta) \end{pmatrix}$$

A **Hessian matrix** of that function is

$$\bigtriangledown^2 h(\theta) = \begin{pmatrix} \frac{\partial^2 h}{\partial \theta_1 \partial \theta_1}(\theta) \cdots \frac{\partial^2 h}{\partial \theta_1 \partial \theta_d}(\theta) \\ \cdots \\ \frac{\partial^2 h}{\partial \theta_d \partial \theta_1}(\theta) \cdots \frac{\partial^2 h}{\partial \theta_d \partial \theta_d}(\theta) \end{pmatrix}$$

h is concave $\implies x^T \bigtriangledown^2 h(\theta) x \leq 0 \ \forall x \in \mathbb{R}^d, \theta \in \Theta$.
h is strictly concave $\implies x^T \bigtriangledown^2 h(\theta) x < 0 \ \forall x \in \mathbb{R}^d, \theta \in \Theta$.

1.16 The general optimization problem

Standard form: minimize $f_0(x)$ subject to $f_i(x) \leq 0,\ i = 1, \dots m,\ h_i(x) = 0,\ i = 1, \dots m$.* Where f_0 is the objective function and $x \in \mathbf{R}^n$ are the optimization variables.
We can replace $h(x) = 0$ with $h(x) \leq 0$ and $-h(x) \leq 0$, so we don't need the equality constraints.
The set of points satisfying the constraints is called the **feasible set**.
A point in the feasible set is called a **feasible point**.
If x is feasible and $f_i(x) = 0$, then we say inequality constraint $f_i(x) \leq 0$ is **active** at x.
The **optimal value** p^* of the problem is defined as

$$p^* = \inf \{f_0(x) | x \in \text{feasible set}\}$$

x^* is an **optimal point** (a solution to the problem) if x^* is feasible and $f(x^*) = p^*$.

1.17 Lagrangian duality

Given a general optimization problem:

$$\begin{aligned} &\text{minimize} f_0(x) \\ &\text{subject to } \ f_i(x) \leq 0, i = 1, \dots m. \end{aligned}$$

The **Lagrangian** for it is defined as

$$L(x, \lambda) = f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x)$$

Where λ_is are called **Lagrangian multipliers** (**dual variables**).†
Supremum over Lagrangian gives back encoding of objective and constraints:

$$\begin{aligned} \sup_{\lambda \succeq 0} L(x, \lambda) &= \sup_{\lambda \succeq 0} (f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x)) \\ &= \begin{cases} f_0(x), & \text{when } f_i(x) \leq 0\ \forall i \\ \infty, & \text{otherwise} \end{cases} \end{aligned}$$

And equivalent of the **primal form** of the optimization problem:

$$p^* = \inf_x \sup_{\lambda \succeq 0} (f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x))$$

*Assuming the intersections of domains of all f_i and h_i is not empty.
†Irrelevant with convexity.

We get the **Lagrangian dual problem** by swapping the inf and sup.

$$d^* = \sup_{\lambda \succeq 0} \inf_x \left(f_0(x) + \sum_{i=1}^m \lambda_i f_i(x)\right)$$

Weak duality $p^* \geq d^*$ holds for any optimization problem. *
Duality gap is $p^* - d^*$, for convex problems, we often have **strong duality**: $p^* = d^*$.
The **Lagrangian dual function** is

$$g(\lambda) = \inf_x L(x, \lambda) = \inf_x \left(f_0(x) + \sum_{i=1}^m \lambda_i f_i(x)\right)$$

The dual function is always concave.†
Lagrangian dual function gives a lower bound on optimal solution:

$$p^* \geq d^* = \sup_{\lambda \geq 0} g(\lambda)$$
$$p^* \geq g(\lambda) \quad \forall \lambda \geq 0$$

The **Lagrangian dual problem** is a search for best lower bound on p^*: maximize $g(\lambda)$ subject to $\lambda \succeq 0$.
λ is **dual feasible** if $\lambda \succeq 0$ and $g(\lambda) > -\infty$, and λ^* is **dual optimal** or **optimal Lagrange multipliers** if they are optimal for the Lagrange dual problem.

For a general optimization problem, if we have **strong duality**, we get a relationship called **complementary slackness** between

- the optimal Lagrange multiplier λ_i^*, and
- the i-th constraint at optimumm: $f_i(x^*)$

$$\lambda_i^* f_i(x^*) = 0$$

Always have Lagrange multiplier being zero, or constraint is active at optimum, or both.
Lagrangian can be applied to illustrate the equivalence of Tikhanov and Ivanov forms of penalizing complexity measure.

1.18 Convex optimization standard form

Standard form of convex optimization problem: minimize $f_0(x)$ subject to $f_i(x) \leq 0, i = 1, \ldots m$. Where $f_0, \ldots f_m$ are convex functions.
We usually have strong duality for convex optimization problems, but not always. The additional conditions needed are called **constraint qualifications**.

*Hint, for any general $f : W \times Z \to R$, this can be proved given $\inf_{w \in W} f(w, z_0) \leq f(w_0, z_0) \leq \sup_{z \in Z} f(w_0, z)$, add another sup to the leftmost term and inf to the rightmost term.

†Pointwise min of affine functions

2 Probability

2.1 Random variable

A random variable $X : \Omega \to E$ is a measurable function from a set of possible outcomes Ω to a measurable space E. Often times $E = \mathbf{R}$
The probability that X takes on a value in a measurable set $S \subseteq E$ is written as

$$Pr(X \in S) = P(\omega \in \Omega | X(\omega) \in S)$$

Intuition: mapping outcomes of a random process to numbers, like this definition of X

$$X = \begin{cases} 0, & \text{if heads} \\ 1, & \text{if tails} \end{cases}$$

Instead of a traditional algebraic variable that can be solved for one value, a random variable can have different values (each with a probability) under different conditions.

2.2 Law of large numbers

$X_i, X_2, \ldots$ is an infinite sequence of independent and identically distributed (iid) random variables with expected value $\mathbb{E}[X_1] = \mathbb{E}[X_2] = \cdots = \mu$ and $\mathbb{V}[X_1] = \cdots = \sigma^2$, and

$$\bar{X}_n = \frac{1}{n}(X_1 + \cdots + X_n)$$

The weak law states that for any positive number ϵ

$$\lim_{n \to \infty} Pr(|\bar{X}_n - \mu| > \epsilon) = 0$$

The weak law follows immediately from Chebychev's inequality.
The strong law states that

$$Pr(\lim_{n \to \infty} \bar{X}_n = \mu) = 1$$

*

Intuition: law of large numbers is a theorem that describes the average of the results obtained from a large number of trials should be close to the expected value, and will tend to become closer as more trials are performed.
In other words, when n becomes large enough, average is a good i.e. consistent estimator of expectation.

*Or alternatively, $\lim_{n \to \infty} \frac{n(A)}{n} = P(A)$

2.3 Expectation, variance

Expectation

$$\mathbb{E}[X] = \sum_{-\infty}^{+\infty} x P_X(x)$$

$$\mathbb{E}[\Phi(X)] = \sum_{-\infty}^{+\infty} \Phi(x) P_X(x)$$

$$\mathbb{E}[\Phi(X,Y)] = \sum_{-\infty}^{+\infty} \sum_{-\infty}^{+\infty} \Phi(x,y) P_{X,Y}(x,y)$$

Where Φ is a function of random variables X and Y, $P_X(x)$ denotes $P(X = x)$, and $P_{X,Y}(x,y)$ is the joint distribution of X and Y ($P_{X,Y}(x,y) = P(X = x \text{ and } Y = y)$).

Chebyshev's inequality
We have $\mathbb{E}(X^2) = \sum_{-\infty}^{+\infty} x^2 P_X(x)$.
Given any number $\epsilon > 0$, construct X_1 from X s.t.

$$X_1 = \begin{cases} 0, & \text{if } |X| \leq \epsilon \\ \epsilon^2, & \text{if } |X| > \epsilon \end{cases}$$

Then obviously $X_1 \leq X^2$, and $\mathbb{E}[X_1] \leq \mathbb{E}[X^2]$, or equivalently

$$\epsilon^2 P(|X| > \epsilon) \leq \mathbb{E}[X^2]$$

Consequently

$$P(|X| > \epsilon) \leq \frac{1}{\epsilon^2} \mathbb{E}[X^2]$$

In particular if $\mathbb{E}[X^2] = 0$, then $P(|X| > \epsilon) = 0 \forall \epsilon > 0$ hence $X = 0$ with probability 1.

Variance

$$\mathbb{V}[X] = \mathbb{E}[(X - \mathbb{E}[X])^2] = \mathbb{E}[x^2] - \mathbb{E}[x]^2$$

Derivation (recall that $\mathbb{E}[X]$ is a constant):

$$\begin{aligned} \mathbb{V}[X] &= \mathbb{E}[(X - \mathbb{E}[X])^2] \\ &= \mathbb{E}[X^2 - 2\mathbb{E}[X] \cdot X + \mathbb{E}[X]^2] \\ &= \mathbb{E}[X^2] - 2 \cdot \mathbb{E}[X] \cdot \mathbb{E}[X] + \mathbb{E}[X]^2 \\ &= \mathbb{E}[X^2] - \mathbb{E}[X]^2 \end{aligned}$$

Some properties:

$$\mathbb{V}[X - a] = \mathbb{V}[X]$$

$$\mathbb{E}[cX] = c\mathbb{E}[X]$$

$$\mathbb{E}[X - a] = \mathbb{E}[X] - a$$

$$\mathbb{E}[X + Y] = \mathbb{E}[X] + \mathbb{E}[Y]$$

$$\mathbb{E}[XY] = \mathbb{E}[X]\mathbb{E}[Y]$$

if X and Y are indepedent.

$$\mathbb{V}[nX] = n^2\mathbb{V}[X]$$

$$\mathbb{V}[-X] = -\mathbb{V}[X]$$

$$\mathbb{V}[X + Y] = \mathbb{V}[X] + \mathbb{V}[Y]$$

If X and Y are independent.
It's then easy to show a "normalized" random variable $X' = \frac{X - \mathbb{E}[X]}{\sqrt{\mathbb{V}[X]}}$ satisfies $\mathbb{E}[X'] = 0$ and $\mathbb{V}[X'] = 1$.

Intuition of $\mathbb{E}$: say we are to use one value m to represent a random variable X, s.t. mean square error is minimized.

$$\begin{aligned}\mathrm{MSE}(m) &= \mathbb{E}[(X - m)^2] \\ &= (\mathbb{E}[X - m])^2 + \mathbb{V}[X - m] \\ &= (\mathbb{E}[X] - m)^2 + \mathbb{V}[X]\end{aligned}$$

$$\frac{\mathrm{dMSE}(m)}{\mathrm{d}m} = 2m - 2\mathbb{E}[X]$$

Thus when $m = \mathbb{E}[X]$, $\frac{\mathrm{dMSE}(m)}{\mathrm{d}m} = 0$.
So the value μ we should use is the expectation.

The expectation can be estimated from the mean of samples $(y_1, \ldots y_n)$.

$$\hat{\mu} \equiv \frac{1}{n}\sum_{i=1}^{n} y_i$$

*

If samples are i.i.d, the **law of large number** suggests that

$$\hat{\mu} \to \mathbb{E}(Y) = \mu$$

Law of total expectation: for any random variables U and V,

$$\mathbb{E}(U) = \mathbb{E}[\mathbb{E}[U|V]]$$

Now instead of predicting one value for a random variable, we are given input X and want to find a function $f : X \to Y$ s.t. the mean square error is minimal. This optimal function is the **regression function**, and

$$\begin{aligned}
\text{MSE}(f) &= \mathbb{E}[(Y - f(X))^2] \\
&= \mathbb{E}[\mathbb{E}[(Y - f(X))^2|X]] \\
&= \mathbb{E}[\mathbb{V}[Y - f(X)|X] + (E[Y - f(X)|X])^2] \\
&= \mathbb{E}[\mathbb{V}[Y|X] + (E[Y - f(X)|X])^2]
\end{aligned}$$

To minimize this, the first term in the expectation doesn't depend on our prediction, and the second term looks like previously when we are predicting the value of a random variable with one value, only with all expectations conditional on X.
So the optimal function $\mu(x) = \mathbb{E}[Y|X = x]$, and this is called **regression function**.

When Y depends on X (causal model), we can use a noise random variable ϵ (whose expectation is 0) and regression function $\mu(x)$ to estimate Y:

$$\mu(X) + \epsilon \to Y$$

When causal relationship $X \to Y$ cannot be assumed, the noise random variable can be dependent on X, thus the general form $Y|X = \mu(X) + \eta(X)$ (without loss of generality, $\eta(X)$'s expectation is 0).

2.4 Bias-variance trade-off

When X takes only a finite set of values and we have enough sample points, we can reliably approximate a regression function (law of large number).
But our sample points is almost always undersampled (e.g. when X is continuous), in which case we need interpolation, extrapolation and smoothing, and there are different methods to do these (different regression methods).

*Where the hat indicates empirical, as seen later on in empirical risk minimizer

Suppose the true regression function is $\mu(x)$ and we use $\hat{\mu}$ to make prediction.* The MSE of $\hat{\mu}$ (at x) can be written as such:

$$\begin{aligned}
\text{MSE}(\hat{\mu}) &= \mathbb{E}[(Y - \hat{\mu}(x))^2] \\
&= \mathbb{E}[(Y - \mu(x) + \mu(x) - \hat{\mu}(x))^2] \\
&= \mathbb{E}[(Y - \mu(x))^2 + 2(Y - \mu(x))(\mu(x) - \hat{\mu}(x)) + (\mu(x) - \hat{\mu}(x))^2] \\
&= \mathbb{E}[\eta^2] + 2(\mu(x) - \hat{\mu}(x))\mathbb{E}[\eta] + \mathbb{E}[(\mu(x) - \hat{\mu}(x))^2] \\
&= \mathbb{V}[\eta] + (\mu(x) - \hat{\mu}(x))^2
\end{aligned}$$

This is our first **bias-variance decomposition**, the second term is a bias by which our predictions are systematically off, and the first term is a variance that affect even the best prediction (which in general depends on x, let's denote it as σ_x^2).
In practice, $\hat{\mu}$ is not a single fixed function: it's something we estimate from sample data. If sample data are random, then exact regression function we get is random, too. Let's call this random function $\hat{M}_n$.†
Now if we are to analyze the prediction error of the *method*, averaging over all the possible training data sets

$$\begin{aligned}
\text{MSE}(\hat{M}_n(x)) &= \mathbb{E}[(Y - \hat{M}_n(X))^2 | X = x] \\
&= \sigma_x^2 + (\mu(x) - \mathbb{E}[\hat{M}_n(x)])^2 + \mathbb{V}[\hat{M}_n(x)]
\end{aligned}$$

This our second bias-variance decomposition. The first term is the same as before. The second term is in using $\hat{M}_n$ to estimate μ, the **approximation error / approximation error**. The third term is the variance in our estimate of the regression function, even if we have an unbiased method $\mu(x) = \mathbb{E}[\hat{M}_n(x)]$, if there is a lot of variance in our estimates, we can expect to make large errors.
The catch is, at least past a certain point, decreasing the approximation bias can only come through increasing the estimation variance. This is the **bias-variance trade-off**.
The trade off doesn't have to be one-for-one, sometimes we can lower the total error by introducing some bias as it gets rid of more variance than it adds approximation error.
In general, both approximation bias and estimation variance depend on n. A method is **consistent** when both of these go to 0 as $n \to \infty$. There can be multiple consistent methods for the same problem, and their biases and variances don't have to go to 0 at the same rate.

*As seen later, the true Bayesian form, and our prediction function which hopes to be as close as possible.

†The previous analysis really is $\text{MSE}(\hat{M_n}(x) | \hat{M_n}(x) = \hat{\mu})$

2.5 Ordinary least squares linear regression, linear smoothers

We choose to approximate $\mu(x)$ by $\alpha + \beta x$ and ask for the best a, b of the those constants.
For the sake of simplicity, we assume X is one dimensional and X and Y have mean 0.

$$\begin{aligned}\text{MSE}(\alpha, \beta) &= \mathbb{E}[(Y - \alpha - \beta X)^2] \\ &= \mathbb{E}[\mathbb{V}[Y|X]] + \mathbb{E}[(\mathbb{E}[Y - \alpha - \beta X|X])^2]\end{aligned}$$

The first term doesn't depend on α or β, so we can drop it. Taking the derivative of the second term, we have

$$\frac{\partial MSE}{\partial \alpha} = \mathbb{E}[-2 \cdot (Y - \alpha - \beta X)]$$

and $a = \mathbb{E}[Y] - b\mathbb{E}[X] = 0$. (we assumed X and Y both have mean 0)

$$\frac{\partial MSE}{\partial \beta} = \mathbb{E}[-2X \cdot (Y - \alpha - \beta X)]$$

and $b = \frac{\mathbb{E}[XY]}{\mathbb{E}[X^2]} = \frac{Cov[X,Y]}{\mathbb{V}[X]} = \frac{\sum_i y_i x_i}{\sum_i x_i^2}$. (we assumed X and Y both have mean 0)

And we are now in a position to see how the least square linear regression model is really a smoothing of the data:

$$\hat{\mu}(x) = \hat{b}x = \sum_i y_i \frac{x_i}{\sum_j x_j^2} x = \sum_i y_i \frac{x_i}{n\hat{\sigma}_X^2} x$$

Where $\hat{\sigma}_X^2$ is the sample variance of X.
The **intuition** is that our prediction is a weighted average of observed values y_i of the dependent variable, where the weights are proportional to how far x_i is from the center (relative to the variance), and proportional to the magnitude of x.
The linear regression line is a special case of **linear smoothers**, which are estimates of the regression function with the following form:

$$\hat{\mu}(x) = \sum_i y_i \hat{w}(x_i, x)$$

They are called linear as the predictions are linear in the responses y_i, as functions of x they are generally nonlinear. In linear regression's case, as shown earlier,

$$\hat{w}(x_i, x) = \frac{x_i}{n\hat{\sigma}_X^2} x$$

This ignores the distance between x_i and x.

2.6 k-Nearest-Neighbor Regression

Consider **nearest-neighbor regression**, which is very sensitive to the distance between x_i and x:

$$\hat{w}(x_i, x) = \begin{cases} 1 & x_i \text{ nearest neighbor of } x \\ 0 & \text{otherwise} \end{cases}$$

This will include the noise into its prediction. We might instead do k-nearest neighbor regression (as noise tend to cancel each other out), where as we increase k we get smoother functions, until $k = n$ where we get a constant.

$$\hat{w}(x_i, x) = \begin{cases} \frac{1}{k} & x_i \text{ one of the } k \text{ nearest neighbor of } x \\ 0 & \text{otherwise} \end{cases}$$

Because k-nearest-neighbors averages over only a fixed number of neighbors, each of which is a noisy sample, it always has some noise in its prediction, and is generally not consistent.

2.7 Kernel smoothers

With kNN regression each testing point is predicted using information from only a few data points, say we want to use all the training data like ordinary linear regression, but in a location-sensitive way.
Kernel smoothing does this. We pick a kernel function $K(x, x)$ which satisfies

- $K(x_i, x) \geq 0$
- $K(x_i, x)$ depends only on the distance $x_i - x$, not the individual arguments. (Hence we can write K as a one-argument function, $K(x_i - x)$)
- $\int xK(0, x)dx = 0$ and
- $0 < \int x^2 K(0, x)dx < \infty$

These conditions together imply $|x_i - x| \to \infty$, $K(x_i, x) \to 0$.*
The Nadaraya-Watson estimate of the regression function is

$$\hat{\mu}(x) = \sum_i y_i \frac{K(x_i, x)}{\sum_j K(x_j, x)}$$

$K(x_i, x)$ is large if x_i is close to x, so this places a lot of weight on training data points close to the point where we are trying to predict.

*Example of such functions include the density of the uniform distribution $(-\frac{h}{2}, \frac{h}{2})$, and the density of the standard Gaussian $\mathbf{N}(0, \sqrt(h))$ distribution. h can be any positive number and is called **bandwidth**, which controls the degree of smoothing, $h \to \infty$ we revert to taking the global mean, and $h \to 0$ results in spikier functions.

2.8 Bernoulli trials, Binomial and Poisson distributions, Normal distribution

A **Bernoulli trial** means identical independent experiments in each of which an event A may occur with probability $P(A)$. In the case of n consecutive Bernoulli trials, each elementary event ω can be described by a sequence of n 0s and 1s. Because of the independence of the trials, the probability of such an event ω with k successes and $n-k$ failures is $p^k(1-p)^{n-k}$ where p is the probability of event A in one trial.
Let random variable ξ denote the number of successes in n Bernoulli trials, $\xi(\omega) = k$ if k successes occur in the event ω.

$$P(\xi = k) = \binom{n}{k} p^k (1-p)^{n-k} = \frac{n!}{k!(n-k)!} p^k (1-p)^{n-k}$$

This is known as the **binomial distribution**, and specified by two parameters n and p.
The random variable ξ can be seen as the sum of n independent random variables $\xi = \xi_1 + \ldots \xi_n$, where each ξ_i has $P(\xi_1 = 1) = p$ and $P(\xi_1 = 0) = 1-p$. It's easy to show that $\mathbb{E}[\xi_i] = p$ and $\mathbb{V}[\xi_i] = p(1-p)$, and $\mathbb{E}[\xi_i] = np$, $\mathbb{V}[\xi_i] = np(1-p)$.
Approximation when n is large and p is small.

$$p_\xi(k) \approx \frac{(np)^k}{k!} e^{-np}$$

A random variable ξ taking integral values $0, 1, 2, \ldots$ is said to have **Poisson distribution** if

$$p_\xi(k) \approx \frac{a^k}{k!} e^{-a}$$

The distribution is specified by a single positive parameter $a = \mathbb{E}[\xi]$. Poisson distribution is the same as the approximation of binomial distribution when the number of trials is large and probability of success is small.

A **normal distribution** probability density follows the form

$$p(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$$

And distribution function follows the form

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{x^2}{2}} dx$$

This has expectation 0 and variance 1.
More generally, a normal random variable is a random variable with probability density

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-a)^2}{2\sigma^2}}$$

It has expectation a and variance σ^2.

2.9 Uncorrelated, independent, Eve's law, Central Limit Theorem

Random variables X and Y are **independent**:

$$F(X,Y) = F(X)F(Y)$$

Where $F(X)$, $F(Y)$ are cumulative distribution functions of X, Y; and $F(X,Y)$ is the joint cumulative distribution function,

$$F(X,Y) = P(X \leq x, Y \leq y)$$

Random variables X and Y are **uncorrelated**:

$$Cov(X,Y) = 0$$

Where the covariance $Cov(X,Y) = \mathbb{E}[XY] - \mathbb{E}[X]\mathbb{E}[Y]$

Eve's law

$$\mathbb{V}[Y] = \mathbb{E}[\mathbb{V}[Y|X]] + \mathbb{V}[\mathbb{E}[Y|X]]$$

Correspondingly, some call the law of total expectation **Adam's law**.

2.10 Generating function

Let ξ be a discrete random variable taking values $0, 1, 2\ldots$ with probabilities $P_\xi(k) = Pr(\xi = k), k = 0, 1, 2\ldots$, then the function

$$F_\xi(z) = \sum_{k=0}^{\infty} P_\xi(k) z^k,\ |z| \leq 1$$

is called the **generating function** of the random variable ξ.
The probability distribution of the random variable ξ is uniquely determined by its generating function $F_\xi(z)$:

$$P_\xi(k) = \frac{1}{k!} F_\xi^{(k)}(0), \quad k = 0, 1, 2\ldots$$

Where $F_\xi^{(k)}(0)$ is the k-th derivative of $F_\xi^{(k)}(z)$.
Since $\mathbb{E}[\Phi(x)] = \sum_{k=-\infty}^{+\infty} \Phi(k)P(k)$, $F_\xi(z)$ at a fixed z is the expectation of the random variable $\Phi(\xi) = z^\xi$:

$$F_\xi(z) = \mathbb{E}[z^\xi], \quad |z| \leq 1$$

The sequence of probability distribution $P_n(k),\ n = 1, 2, \ldots$ with generating functions $F_n(z),\ n = 1, 2, \ldots$ converges weakly to the limiting distribution $P(k)$ iff $\lim_{n\to\infty} F_n(z) = F(z)$ where $F(z)$ is the generating function of $P(k)$.
Given a real random variable ξ, **characteristic function** of ξ is the function

$$f_\xi(t) = \mathbb{E}[e^{i\xi t}],\ \ -\infty < t < \infty$$

For a discrete random variable the characteristic function $f_\xi(z)$ on the boundary of the unit circle $|z| = 1$, i.e.

$$f_\xi(t) = F_\xi(e^{it}) = \sum_{k=0}^{\infty} P_\xi(k)e^{ikt}$$

This represents $f_\xi(t)$ as Fourier series with probability $P(\xi = k)$ as its coefficients.
For a continuous random variable, the characteristic function is the **Fourier transform** of the density $p_\xi(x)$:

$$f_\xi(t) = \int_{-\infty}^{+\infty} P_\xi(x)e^{ixt}dx$$

In both cases, $P_\xi(k)$ is uniquely determined by the characteristic function.

2.11 Different types of convergence

Let $(T_n)_{n\geq 1}$ be a sequence of r.v. and T a r.v. (T can be deterministic)

- Almost surely (a.s.) convergence: $T_n \xrightarrow[n\to\infty]{a.s.} T$ iff $\mathbf{P}[\{\omega : T_n(\omega) \underset{n\to\infty}{\longrightarrow} T(\omega)\}] = 1$
- Convergence in probability: $T_n \xrightarrow[n\to\infty]{\mathbf{P}} T$ iff $\mathbf{P}[|T_n - T| \geq \epsilon] \underset{n\to\infty}{\longrightarrow} 0, \forall \epsilon > 0$
- Convergence in L^p: $T_n \xrightarrow[n\to\infty]{(L^p)} T$ iff $\mathbf{P}[|T_n - T|^p] \underset{n\to\infty}{\longrightarrow} 0$
- Convergence in distribution L^p: $T_n \xrightarrow[n\to\infty]{(d)} T$ iff $\mathbf{P}[T_n \leq x] \underset{n\to\infty}{\longrightarrow} \mathbf{P}[T \leq x]$

Intuition: the first is very strong. It says if you measure one under whatever observation (ω), it's almost the same as if you measure the other. One implies two. For three if you converge in L^p you also converge in L^q for all $q < p$. Two implies four: if you converge in probability, you converge in distribution.

Continuous mapping theorem: if f is a continuous function, then

$$T_n \xrightarrow[n\to\infty]{a.s./\mathbf{P}/(d)} T \implies f(T_n) \xrightarrow[n\to\infty]{a.s./\mathbf{P}/(d)} f(T)$$

Intuition: apply a continuous function on the sequence of r.v. and convergence still holds.

One can add, multiply, divide, limit, etc, when converge almost surely and in probability. If $U_n \xrightarrow[n\to\infty]{a.s./\mathbf{P}} U$ and $V_n \xrightarrow[n\to\infty]{a.s./\mathbf{P}} V$, then

$$V_n + U_n \xrightarrow[n\to\infty]{a.s./\mathbf{P}} V + U$$

$$V_n U_n \xrightarrow[n\to\infty]{a.s./\mathbf{P}} VU$$

Same goes for division if $V \neq 0$. Strong LLN suggests almost surely convergence, and weak LLN suggests convergence in probability.

2.12 Central limit theorem

Given a sequence of i.i.d r.v. $X_i, i = 1, 2, \ldots, n$ with mean $\mu = \mathbb{E}[X]$ and variance $\sigma^2 = \mathbb{V}[X]$, the weak and strong laws of large number says

$$\bar{X_n} = \frac{1}{n}\sum_{i=1}^{n} X_i \xrightarrow[n\to\infty]{\mathbb{P},a.s.} \mu$$

And central limit theorem says

$$\sqrt{n}\frac{\bar{X_n} - \mu}{\sigma} \xrightarrow[n\to\infty]{(d)} N(0,1)$$

Intuition: LLN states that average is a consistent estimator of expectation when n is large enough. CLT says how good it is, and that the distribution of the sum of a large number of iid random varaibles is approximately normal.

Suppose the sequence $\xi_k, k = 1, 2, \ldots$ with finite means $a_k = \mathbb{E}[\xi_k]$ and variance $\sigma_k^2 = \mathbb{V}[\xi_k]$ satisfies **Lyapunov condition**

$$\lim_{n\to\infty} \frac{1}{B_n^2}\sum_{k=1}^{n} \mathbb{E}[(\xi_k - a_k)^2] = 0$$

where $B_n^2 = \mathbb{V}[S_n] = \sum_{k=1}^{n} \sigma_k^2$
Then the sequence of random variables satisfy the central limit theorem.

Slutsky's theorem suggests under certain scenario, you can combine two sequences when one converges in probability and the other converges in distribution.
By Slutsky, we have the **Delta method**, which allows us to apply functions on sequence of r.v. in CLT. I.e.
Let $(Z_n)_{n\geq 1}$ be a sequence of r.v. that asymptotically normal around θ, i.e. satisfies

$$\sqrt{n}(Z_n - \theta) \xrightarrow[n\to\infty]{a.s./\mathbf{P}} \mathbf{N}(0, \sigma^2)$$

Let $g : \mathbb{R} \to \mathbb{R}$ be continuously differentiable at the point θ, then $g(Z_n)$ is also asymptotically normal, more precisely

$$\sqrt{n}(g(Z_n) - g(\theta)) \xrightarrow[n\to\infty]{a.s./\mathbf{P}} \mathbf{N}(0, g'(\theta)^2\sigma^2)$$

2.13 Hoeffding's inequality

Let n be a positive integer and $X, X_1, ..., X_n$ be i.i.d r.v. s.t. $X \in [a, b]$. Let $\mu = E[X]$. Then for all $\epsilon > 0$,

$$P(|\bar{X_n} - \mu| \geq \epsilon) \leq 2e^{-\frac{2n\epsilon^2}{(b-a)^2}}$$

Intuition: average of bounded i.i.d r.v.'s probability distribution function is bounded by a Gaussian distribution (thinner tails) (even without n approaching infinity).

2.14 Degree of freedom

The number of values in the final calculation of a statistic that are free to vary without violating constraints.
In general, the **Degrees of Freedom** of an estimate of a parameter are equal to the number of independent score that go into the estimate minus the number of parameters used as intermediate steps in the estimation of the parameter itself. Hence most of the time sample variance has $N-1$ degrees of freedom, N random points minus the 1 intermediate step, the sample mean.

2.15 Statistical experiment

The ultimate goal of statistics is to say what distribution your data comes from. The purpose of modeling is to restrict the space of possible distributions to a subspace that is possible to estimate.
In that sense, all models are wrong, some models are useful.
A **statistical experiment** is the pair $(E, \{\mathbf{P}_\theta\}_{\theta\in\Theta})$, where E is a set, the **sample space** (the possible values my observations can take), and each of P_θ is a probability distribution (e.g. $Poiss(\theta)$, $Bernoulli(\theta)$, etc. θ is a parameter set, it can be a pair, a couple of numbers, etc. The point is, once θ is known, you know the distribution perfectly.)
Well specified means for an observation X, $\exists\theta \in \Theta$ where $X \sim \mathbf{P}(\theta)$. This is a strong assumption. (X is just any one among the i.i.d $X_1, \dots X_n$).
θ (θ^*) is the true parameter we want to estimate. θ may have physical links to the phenomenon, like in linear regression, it suggests how sensitive it is to a dimension.
A **parametric model** is where θ has finite number of values θ_d, and all $\theta_d \in \mathbb{R}$. Non-parametric estimation can be, e.g. when you have infinite numbers in θ to estimate, or when θ is a function.
Example models:

- A Bernoulli trial is a pair $(\{0,1\}, \{Ber(p)\}_{p\in(0,1)})$.
- An exponential trial is a pair $((0,+\infty]), \{Exp(\lambda)\}_{\lambda>0})$.
- A Poisson trial is a pair $(\mathbb{N}, \{Poiss(\lambda)\}_{\lambda>0})$.
- A normal distribution is a pair $(\mathbb{R}, \{\mathbf{N}(\mu,\sigma^2)\}_{\mu\in\mathbb{R},\sigma^2>0})$
- A Gaussian with known distribution (imagine machine parts with manufacturer labeled standard deviation), $(\mathbb{R}, \mathbf{N}(\mu,\sigma^2)_{\mu\in\mathbb{R}})$, compared with previous, the parameter space only need to specify μ, as σ^2 is known
- Censored data (imagine asking people's salary and getting answers like 5 figures). Given an indicator function I ($X = I(Y > 5)$, meaning $X = 1$ if $Y > 5$ else $X = 0$), and $Y \sim Exp(\lambda)$, then we have $(\{0,1\}, \{Ber(e^{-5\lambda})\}_{\lambda>0})$, the parameter space can also be expressed in p, the λ comes from integral of exponential on $(5, \lambda)$.
- A uniform distribution $X \sim \mathbf{U}([0,\theta])$, $\theta > 0$ is a pair $([0,+\infty), \{Unf(\theta)\}_{\theta>0})$. Note that the sample space is not $[0,\theta]$ as it should not depend on θ.
- Or an arbitrary pmf (probability mass function) for a sample space of $1\ldots100$.

The parameter θ is called **identified / identifiable** (in this model) iff the map $\theta \in \Theta \to \mathbf{P}_\theta$ is **injective**. I.e.

$$\theta \neq \theta' \implies \mathbf{P}_\theta \neq \mathbf{P}_{\theta'}$$

Intuition: it could be the case that two different θs can give me the exact same probabilities, in which case θ is not identifiable (even with infinite amount of data). In the above examples the parameters are identifiable.
($\mathbf{N}(\mu,\sigma^2)$ minus μ then divide by σ gives you standard Gaussian.)
Given a statistical model, we want to estimate θ.
An **estimator** is a measurable function of data $\hat{\theta} = \hat{\theta}(X_1,\ldots,X_n)$ (which does not depend on the parameter) a.k.a a **statistic**. (e.g. average, max, etc. Measurable means that given data, you can compute it, it does not depend on θ. Things that are implicitly defined, such as sup or inf, are not measurable.)
An estimator is **weakly consistent** if it converges in probability to θ, or strongly consistent if it converges almost surely. (When you get more data, the estimator converges to the true parameters.)
The **bias** of an estimator is how far the $\hat{\theta}$ it gets is from θ: $\mathbb{E}[\hat{\theta}] - \theta$. If the above is zero, estimator is unbiased. (If I were to repeat the experiment over and over again, on average we get the true parameter. But often times you don't get to repeat the experiment.)
However being unbiased is not the only thing we are after: consider a Bernoulli distribution, for n i.i.d data points, by linearality $\mathbb{E}[\bar{X}_n] = \frac{1}{n}\Sigma_{i=1}^n \mathbb{E}[X_i] = \mathbb{E}[X_1] = p$. This is saying the bias of n data points is as good as 1 data point.

Hence we also have **(quadratic) risk** of $\hat{\theta}$, $\mathbb{E}[|\hat{\theta}-\theta|^2]$. If this goes to 0 as $n \to \infty$, then this is saying $\hat{\theta} \xrightarrow[n\to\infty]{L^2} \theta$, and this implies convergence in probablity, hence $\hat{\theta}$ is weakly consistent. So going back to the Bernoulli example, $\mathbb{E}[\bar{X_n}]$ is consistent as $n \to \infty$, but $\mathbb{E}[X_1]$ is not.
This risk is the sum of bias and variance.

$$\begin{aligned}\mathbb{E}[|\hat{\theta}-\theta|^2] &= \mathbb{E}[(\hat{\theta}-\mathbb{E}[\hat{\theta}]+\mathbb{E}[\hat{\theta}]-\theta)^2] \\ &= \mathbb{E}[(\hat{\theta}-\mathbb{E}(\hat{\theta}))^2]+(\mathbb{E}[\hat{\theta}]-\theta)^2 \\ &= \mathbf{V}(\hat{\theta})+bias^2\end{aligned}$$

E.g. given $X_1 \dots X_n \sim Ber(\theta)$,

- the estimator $\bar{X_n}$ has 0 bias and $\frac{1}{n}\theta(1-\theta)$ variance. Its risk is the sum of the two.
- the estimator 0.5 has bias $(0.5-\theta)$ and variance 0.
- the estimator X_1 has bias 0 and variance $\theta(1-\theta)$. In this sense the first estimator has smaller risk.

Bias variance tradeoff happens all the time especially in high dimensional and non-parametric estimations.

2.16 Confidence interval

Let $(E, \{\mathbb{P}_\theta\}_{\theta\in\Theta})$ be a statistical model based on observations $X_1, \dots X_n$ and assume $\Theta \subseteq \mathbb{R}$.
Let $\alpha \in (0,1)$, **confidence interval** of level $1-\alpha$ for θ means any random (i.e. depending on $X_1 \dots X_n$) interval I whose boundaries do not depend on θ and such that

$$\mathbb{P}_\theta[I \ni \theta] \geq 1-\alpha, \quad \forall\theta \in \Theta$$

Confidence interval of asymptotic level $(1-\alpha)$ for θ means any random (i.e. depending on $X_1 \dots X_n$) interval I whose boundaries do not depend on θ and such that

$$\lim_{n\to\infty} \mathbb{P}_\theta[I \ni \theta] \geq 1-\alpha, \quad \forall\theta \in \Theta$$

(when you use CLT.)
Intuition: you are given samples and want to understand something about the underlying distribution, in particular, you have sample mean and want to understand something about population mean. Given a level of confidence $x\%$, you want to find the interval which contains the population mean with $x\%$. When dealing with normal distribution, confidence interval depends on whether population standard deviation is known or estimated with sample standard deviation. If not, then student's T distribution is used as the critical value.

2.17 Maximum likelihood estimation

Let $(E, \{\mathbb{P}_\theta\}_{\theta\in\Theta})$ be a statistical model associated with i.i.d r.v. $X_1, \dots X_n$ and assume $\exists\theta^* \in \Theta$ such that $X_i \sim \mathbb{P}_{\theta^*}$: θ^* is the true parameter.
Statistician's goal is, given $X_1, \dots X_n$, find an estimator $\hat{\theta} = \hat{\theta}(X_1, \dots X_n)$ such that $\mathbb{P}_{\hat{\theta}}$ is close to $\mathbb{P}_{\theta^*}$. This means $|\mathbb{P}_{\hat{\theta}}(A) - \mathbb{P}_{\theta^*}(A)|$ is small for all $A \subset E$.
The total variation distance is then defined as $TV(\mathbb{P}_\theta, \mathbb{P}_{\theta'}) = max|\mathbb{P}_\theta(A) - \mathbb{P}_{\theta'}(A)| \ \ \forall A \subset E$
This can then be translated to sum on the two pmf for discrete r.v., or integral on two density functions for continuous r.v..
Given the PMF of X: $\mathbb{P}_\theta(X = x) = p_\theta(x)$ for all $x \in E$,

$$p_\theta(x) \geq 0, \ \sum_{x\in E} p_\theta(x) = 1$$

The total variation distance between $\mathbb{P}_\theta$ and $\mathbb{P}_{\theta'}$ is a function of PMF's p_θ and $p_{\theta'}$:

$$TV(\mathbb{P}_\theta, \mathbb{P}_{\theta'}) = \frac{1}{2}\sum_{x\in E} |p_\theta(x) - p_{\theta'}(x)|$$

And in the continuous case f_θ being the density function,

$$TV(\mathbb{P}_\theta, \mathbb{P}_{\theta'}) = \frac{1}{2}\int_E |f_\theta(x) - f_{\theta'}(x)|dx$$

Properties of total variation distance include: symmetric, non-negative, definite ($TV(\mathbb{P}_\theta, \mathbb{P}_{\theta'}) = 0 \implies \mathbb{P}_\theta = \mathbb{P}_{\theta'}$), and triangle inequality $TV(\mathbb{P}_\theta, \mathbb{P}_{\theta'}) \leq TV(\mathbb{P}_\theta, \mathbb{P}_{\theta''}) + TV(\mathbb{P}_{\theta'}, \mathbb{P}_{\theta''})$
These implies total variation is a **distance**. We then want to minimize the total variation distance of θ and $\theta*$, but we don't know $\theta*$. We build an estimator $\hat{TV}(\mathbb{P}_\theta, \mathbb{P}_{\theta*})$ for all $\theta \in \Theta$. Then find $\hat{\theta}$ that minimizes function $\theta \to \hat{TV}(\mathbb{P}_\theta, \mathbb{P}_{\theta*})$.
The problem then becomes building the above estimator.

Kullback-Leibler (KL) divergence between two distributions is defined by the following. Discrete:

$$KL(\mathbb{P}_\theta, \mathbb{P}_{\theta'}) = \sum_{x\in E} p_\theta(x) log(\frac{p_\theta(x)}{p_{\theta'}(x)})$$

Or continuous:

$$KL(\mathbb{P}_\theta, \mathbb{P}_{\theta'}) = \int_E f_\theta(x) log(\frac{f_\theta(x)}{f_{\theta'}(x)})dx$$

KL makes the distance (e.g. between two Gaussians) easy to compute. KL is not symmetric, and it is not a distance, but definite property still holds. KL looks just like the density of the function log division.

And,

$$KL(\mathbb{P}_\theta, \mathbb{P}_{\theta'}) = \mathbb{E}_{\theta*}[log(\frac{p_{\theta*}(X)}{p_\theta(X)})] = \mathbb{E}_{\theta*}[log(p_{\theta*}(X)] - \mathbb{E}_{\theta*}[log(p_\theta(X)]$$

The first term is constant (to θ change), and the second term can be estimated with LLN, and we get

$$\hat{KL}(\mathbb{P}_{\theta*}, \mathbb{P}_\theta) = \text{constant} - \frac{1}{n}\sum_{i=1}^{n} log(p_\theta(X_i))$$

We still don't know the constant / first term yet, but the minimizer θ of $\hat{KL}(\mathbb{P}_{\theta*}, \mathbb{P}_\theta)$ is the same no matter the constant.

$$\arg\min_{\theta\in\Theta} \hat{KL}(\mathbb{P}_{\theta*}, \mathbb{P}_\theta) \iff \arg\max_{\theta\in\Theta} \prod_{i=1}^{n} p_\theta(X_i)$$

This is the **maximum likelihood principle**.
Intuition: the multiplicative form we arrived at is the joint density of $X_1 \dots n$. Imagine the model is a Gaussian centered at $\theta*$, and we have only one observation at θ', then the maximum likehood estimator is going to estimate a Gaussian centered around θ'. Then if we have n points, we are just multiplying the values mapped onto the vertical line of different θs, and finding the θ such that the product is maximized.
In the real world maximum likelihood estimators often don't have a closed form, hence the many optimizations in machine learning.

The **likelihood of a model** $(E, (\mathbb{P}_\theta)_{\theta\in\Theta})$ associated with a sample of i.i.d r.v $X_1, \dots X_n$ can then be defined as

$$\begin{aligned} L_n : E^n \times \Theta &\to \mathbb{R} \\ (x_1, \dots, x_n, \theta) &\to \mathbb{P}_\theta[X_1 = x_1, \dots, X_n = x_n] \end{aligned}$$

If E is discrete.

$$\begin{aligned} L_n : E^n \times \Theta &\to \mathbb{R} \\ (x_1, \dots, x_n, \theta) &\to \prod_{i=1}^{n} f_\theta(x_i) \end{aligned}$$

where f is the density. If E is continuous. This can be derived from KL divergence.
Example: likelihood of a Bernoulli model (discrete), Gaussian (continuous)
In stats, you first design your estimator as a function of random variables, then you get data and plug them in. (Keep the random variables for as long as you can.)

The **maximum likelihood estimator** of θ is defined as

$$\hat{\theta}_n^{MLE} = \arg\max_{\theta \in \Theta} L(X_1, \ldots, X_n, \theta)$$

In practice, we use the log likelihood estimator as argmax is the same

$$\hat{\theta}_n^{MLE} = \arg\max_{\theta \in \Theta} log\ L(X_1, \ldots, X_n, \theta)$$

Maximum likelihood estimation is what's driving our intuition, e.g. when we use the average to estimate the p of a Bernoulli trial.
MLE minimizes KL divergence.

The **Fisher information** of a statistical model $l(\theta) = log\ L_1(X, \theta)$ is defined as

$$\begin{aligned} I(\theta) &= \mathbb{E}[\bigtriangledown l(\theta) \bigtriangledown l(\theta)^T] - \mathbb{E}[\bigtriangledown l(\theta)]\mathbb{E}[\bigtriangledown l(\theta)]^T \\ &= -\mathbb{E}[\bigtriangledown^2 l(\theta)] \end{aligned}$$

Assuming l is twice differentiable (pseudo-inverse of the covariance matrix?).
If $\theta \subset \mathbb{R}$, we get:

$$I(\theta) = var[l'(\theta)] = -\mathbb{E}[l''(\theta)] = \int \frac{(\frac{\partial f_\theta(x)}{\partial \theta})^2}{f_\theta(x)} dx$$

Intuition what Fisher information says is the more curvy the distribution is, the flatter it is when your estimator is around maximum, and the less sensitive we are to fluctuations (and the larger Fisher information, i.e. how much information of θ your model has).
Let $\theta^* \in \Theta$, assume the model is identified, $\forall \theta \in \Theta$, the support of $\mathbb{P}_\theta$ does not depend on θ, θ^* is not on the boundary of Θ, $I(\theta)$ is invertible in a neighborhood of θ^*, then $\hat{\theta}_n^{MLE}$ satisfies

$$\hat{\theta}_n^{MLE} \xrightarrow[n \to \infty]{\mathbb{P}} \theta^* \quad w.r.t\ \mathbb{P}_{\theta^*}$$

MLE converges in probability to $\theta*$. Bias goes to 0.

$$\sqrt{n}(\hat{\theta}_n^{MLE} - \theta^*) \xrightarrow[n \to \infty]{(d)} \quad N(0, I(\theta^*)^{-1}) w.r.t\ \mathbb{P}_{\theta^*}$$

MLE minus θ^* converges in distribution to normal distribution. θ^* may not even have a closed form, and we still have the same properties of an average. CLT-like. Variance is measured by Fisher information.

2.18 Weierstrass approximation theorem

Let f be a continuous function on the internal $[a, b]$ then for any $\epsilon > 0$ there exists $a_0, a_1, \ldots, a_d \in \mathbb{R}$ such that

$$max_{x\in[a,b]}(|f(x) - \sum_{k=0}^{d} a_k x^k|) < \epsilon$$

Intuition: continuous functions can be arbitarily well approximated by polynomials.
To estimate θ_* with θ (assuming identifiable), if

$$\begin{aligned}\int h(x) f_{\theta^*}(x)dx &= \int h(x) f_\theta(x)dx \\ &= \mathbb{E}_{\theta^*}[h(X)] \\ &\approx \frac{1}{n}\sum_{i=1}^{n} h(X_i)\end{aligned}$$

for all bounded continuous functions h, then $\theta = \theta^*$.
And what Weierstrass approximation theorem tells us is we don't need to look at all $h(x)$, only $d+1$ polynomial functions to the d would work.
The quantity $m_k(\theta) = \int x^k f_\theta(x)dx = \mathbb{E}_\theta[X^k]$ is the kth **moment** of $\mathbb{P}(\theta)$.
Mean is therefore the first moment. Variance is second moment minus first squared.

2.19 Gaussian quadrature

Imagine we have a discrete E with r possible values and are looking at a PMF. Then

$$m_k = \mathbb{E}[X^k] = \sum_{j=1}^{r} p(x_j) x_j^k$$

and $\sum_{j=1}^{r} p(x_j) = 1$ with unknowns $p(x_j)$.
And we can write it in compact form:

$$\begin{pmatrix} x_1^1 \dots x_r^1 \\ \dots \\ x_1^{r-1} \dots x_r^{r-1} \\ 1 \dots 1 \end{pmatrix} \begin{pmatrix} p(x_1) \\ \dots \\ p(x_r) \end{pmatrix} = \begin{pmatrix} m_1 \\ \dots \\ m_{r-1} \\ 1 \end{pmatrix}$$

Using Vandermonde determinant we can check the x matrix is invertible. (as x_i and x_j are all different: they are the different values your r.v. can take.)
So given the moments, there is a unique PMF that has these moments, which can be recovered from the inverse of x matrix.
Rule of thumb is if $\theta \in \Theta \subset \mathbb{R}^d$, we need d moments.

2.20 Method of moments

Let $X_1, ..., X_n$ be an i.i.d. sample associated with a statistical model $(E, (\mathbb{P}_\theta)_{\theta\in\Theta})$.
Assume that $\theta \subseteq \mathbb{R}^d$, for some $d \geq 1$.
Population moment is $m_k(\theta) = \mathbb{E}_\theta[X^k],\ 1 \leq k \leq d$. Empirical moment is $\hat{m_k} = \bar{X_n^k} = \frac{1}{n}\sum_{i=1}^n X_i^k,\ 1 \leq k \leq d$.
Let $\phi : \Theta \subset \mathbb{R}^d \to \mathbb{R}^d$ and $\theta = (m_1(\theta), \ldots, m_d(\theta))$.
Assume ϕ is one to one, then $\theta = \phi^{-1}(m_1(\theta), \ldots, m_d(\theta))$ and **moments estimator** of θ is

$$\theta_n^{\hat{M}M} = \phi^{-1}(\hat{m_1}, \ldots, \hat{m_d})$$

provided it exists.
If ϕ^{-1} is very steep, then this estimator is not very good.
LLN suggests $\theta_n^{\hat{M}M}$ is weakly / strongly consistent.
CLT suggests $\sqrt{n}(\hat{M} - M(\theta)) \xrightarrow[n\to\infty]{(d)} N(0, \Sigma(\theta))$ (w.r.t $\mathbb{P}_\theta$).
Hence by Delta Method,

$$\sqrt{n}(\theta_n^{\hat{M}M} - \theta) \xrightarrow[n\to\infty]{(d)} N(0, \Gamma(\theta))$$

Where $\Gamma(\theta) = [\bigtriangledown\phi^{-1}M(\theta)]^T\Sigma(\theta)[\bigtriangledown\phi^{-1}M(\theta)]$.

Condition number of a matrix.
MLE is typically more accurate than moment estimator, in terms of quadratic risk. Moments is fairly doable, though MLE is sometimes intractable. If likelihood is not concave, then method of moments can start you in the right peak to the maxima.

2.21 Parametric hypothesis testing

Example:

- A coin is tossed X times, and heads are obtained Y times. Can we conclude the coin is significantly unfair?
- Drug testing. A control group is given placebo, and a test group given a cough syrup. Measure the number of coughs of both groups, and how much does their mean need to differ, for us to conclude the cough syrup is effective?

Consider a sample $X_1, \ldots X_n$ of i.i.d r.v. and a statistical model $(E, (\mathbb{P}_\theta)_{\theta\in\Theta})$.
Let Θ_0 and Θ_1 be disjoint subsets of Θ. (anything that's in Θ but not either Θ_0 or Θ_1 we are not going to make any statement about those cases.)
Consider the two hypotheses, $H_0 : \theta \in \Theta_0$ and $H_0 : \theta \in \Theta_1$.
H_0 is the **null hypothesis** (*the status quo*), H_1 is the **alternative hypothesis** (*scientific discovery that goes against status quo*).

If we believe true θ is in either Θ_0 or Θ_1, we may want to test H_0 against H_1, and decide whether to reject H_0 (*look for evidence against H_0 in the data*).
The role of H_0 and H_1 are not equivalent. By default, we fail to reject H_0.

A **test** is a statistic (*a function we compute on the data, in this case, an indicator function*) $\psi \in \{0, 1\}$ s.t. if $\psi = 0$, H_0 is not rejected, and if $\psi = 1$, H_0 is rejected.

In the coin example, $H_0 : p = \frac{1}{2}$, $H_1 : p \neq \frac{1}{2}$.

$$\psi = I\{|\sqrt{n}\frac{\bar{X_n} - .5}{\sqrt{\bar{X_n}(1 - \bar{X_n})}}| > C\}$$

for some $C > 0$. (*above this threshold, this is likely from a Gaussian, below, this is unlikely from a Gaussian.*)
Rejection region $R_\psi = x \in E^n : \psi(x) = 1$.
Type 1 Error: rejecting H_0 when it is actually true. Actually status quo but I claim a discovery.
Type 2 Error: not rejecting H_0 although H_1 is actually true.
The **power of a test** ψ is $\pi_\psi = \arg\min_{\theta \in \Theta_1}(1 - \beta_\psi(\theta))$ where β_ψ is the probability of Type 2 Error. (*Under the worst case, the likelihood of us correctly rejecting the null hypothesis.*)
Example, in a court where subject is innocent unless proven guilty, Type I is judging an innocent subject guilty, and Type II is judging a guilty person innocent. There is tradeoff between Type I and Type II error, think judging everyone innocent vs judging everyone guilty, and this is a multi-objective optimization, in which case the best one can do is usually combining the different objectives using ad hoc heuristics.
In statistics, the usual criteria is to constrain Type I error to be at most 5% (and minimize Type II from there).

Power function $\mathbb{P}_\Theta(\psi = 1)$ of a test is defined on Θ means the probability of rejecting null hypothesis at that θ. The value π is then the smallest value of power function in the domain Θ_1.

A test ψ has **level** α if $\alpha_\psi \leq \alpha$, $\forall \theta \in \Theta_0$. Where $\alpha_\psi = \arg\max_{\theta \in \Theta_0} \alpha_\psi(\theta)$, the maximum Type I error probability when $\theta \in \Theta_0$.

A test has **asymptotic level** α if $\lim_{n \to \infty} \alpha_\psi \leq \alpha$, $\forall \theta \in \Theta_0$. When the number of samples approaches infinity.

In general, a test has the form $\psi = I\{T_n > c\}$ (indicator function) for some statistic T_n and threshold $c \in \mathbb{R}$.
T_n is the **test statistic**, and the rejection region is $R_\psi = \{T_n > c\}$.

Given α (5%), we pick then c such that α is at that value.

Example: given a Bernoulli coin flip, test whether the coin is fair (H_0 is $p = 0.5$). One test statistic we can use is $T_n = \sqrt{n}\frac{\bar{X_n} - p}{\sqrt{\bar{X_n}(1-\bar{X_n})}}$. We want to constrain this $P(T_n > c)$ at the level of 0.05, $\forall p \in \Theta_0$ (which is equivalent to $p = 0.5$).
Then $T_n = \sqrt{n}\frac{\bar{X_n} - 0.5}{\sqrt{\bar{X_n}(1-\bar{X_n})}}$ and by CLT $\lim_{n\to\infty} T_n \sim N(0,1)$ (limit comes in as we are targetting an asymptotic level), and solving for c in $P\{|N(0,1)| > c\} = 0.05$ gives us $c = 1.96$ (often denoted as $q_{\frac{\alpha}{2}} = 1.96$).

The asymptotic **p-value** of a test ψ_α is the smallest asymptotic level α at which ψ_α rejects H_0. It is random and depends on the sample.
Intuition, the best probability of obtaining test results at least as extreme as the results actually observed, if H_0 holds.

The golden rule is that p-value $\leq \alpha \iff H_0$ is rejected by ψ_α at the asymptotic level α.
Intuition: highest probability of Type I error.
The smaller the **p-value**, the more confidently one can reject H_0.
p-value of a standard Gaussian is the probability $P(|N(0,1)| > |T(n)|) = P(1 - \Psi(T_n))$ where $\Psi(T_n)$ is the Gaussian CDF.

A test is usually about, given an estimator that gives us $\hat{\theta}$, we have a $T_n = f(\hat{\theta}, \theta_0, n)$ (and maybe some covariance matrix representing cross-terms for multi-dimensional θ).
T_n conforms to some distribution under H_0 (say, Gaussian, T, or χ^2, etc), and given a desired level (Type I error cap, y-axis of T_n's pdf), find the corresponding point on T_n (x-axis), such that when plugged with different values of $\hat{\theta}$, we can compute T_n and decide whether it sits left or right to the afore-mentioned point, therefore conclude whether $\hat{\theta}$ sits far enough (on a pdf under H_0), and if it does H_0 should be rejected, and we know the likelihood of erroneously rejecting H_0 is capped.

2.22 χ^2 distribution and Student T distribubtion

χ^2_d distribution ($Z_1^2 + \cdots + Z_d^2$, where $Z_i \sim N(0,1)$. Think throwing darts with x, y axis variations being Gaussian as a χ^2_2. Expected value of χ^2_d is d), χ^2_2 is exponential distribution $Exp(\frac{1}{2})$.

Going back to the coin toss example, note that how you formulate your H_0 and H_1 matters: Say a coin shows head 13 out of 30 times, then H_0 being $p = 0.5$ H_1 being $p \neq 0.5$ will not be rejected, but H_0 being $p = 0.5$ H_1 being $p = 0.43$ will be rejected (note that this will be with a different test, say, test whichever hypothesis has the higher probability).
On a high level, given data $X_1, ...X_n$ and a level α, and the test spits out an yes/no. (or given data, test spits out a p-value.)

Neyman-Pearson's Paradigm, for given hypotheses, among all tests of level / asymptotic level α, is it possible to find one that has maximal power? (e.g. the trivial test $\psi = 0$ that never rejects H_0 has a perfect level $\alpha = 0$ but poor power $\pi_\psi = 0$)

For a positive integer d, **Student's T Distribution** with d DoF (denoted as t_d) is the law of the random variable $\frac{Z}{\sqrt{\frac{V}{d}}}$. Where $Z \sim N(0,1)$, $V \sim \chi^2_d$ and $Z \perp\!\!\!\perp V$.
Why care about this?
For an $X \sim N(\mu, \sigma^2)$, we can standardize it with $Z = \frac{X-\mu}{\sigma} \sim N(0,1)$. To do this, we need to know μ and σ. μ we know from H_0 when doing hypothesis testing (in Student T test, $H_0 : \mu = \mu_0$), but σ we don't know.
Sample variance S_n is a consistent estimator of σ^2.

$$S_n = \frac{1}{n}\sum_{i=1}^{n}(X_i - \bar{X})^2 = \frac{1}{n}\sum_{i=1}^{n} X_i^2 - \bar{X}^2 \approx \sigma^2$$

(Essentially, sample mean S_n (centering with mean) is as if we lose one DoF in the χ^2 distribution, think of the distribution of $\frac{1}{n}\sum_{i=1}^{n} X_i^2$ where $X_i \sim N(0, \sigma^2)$. The $\frac{n}{\sigma^2}$ term below adjusts for σ^2, and we are already centered on $\mu = 0$).

$$\frac{nS_n}{\sigma^2} \sim \chi^2_{n-1}$$

And we want to use S_n to estimate σ. We do

$$\frac{\bar{X_n} - \mu}{\sqrt{S_n}} = \frac{\bar{X_n} - \mu}{\sigma}\frac{\sigma}{\sqrt{S_n}} = \frac{X - \mu}{\sigma}\frac{\sqrt{n}}{\sqrt{\frac{nS_n}{\sigma^2}}} = Z\frac{\sqrt{n}}{\sqrt{\chi^2_{n-1}}}$$

So, denoting $V = \chi^2_{n-1}$

$$\frac{\bar{X_n} - \mu}{\sqrt{S_n}} = \frac{Z}{\sqrt{\frac{V}{n}}}$$

Note that this looks almost the same as the definition for T-distribution, except an $\sqrt{\frac{n}{n-1}}$ term.
We could instead use a different estimator for σ^2, which is also unbiased.

$$S_n = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})^2$$

T-distribution has heavier tails than standard Gaussian (hence wider confidence intervals at the same level). Given law of large numbers, T-distribution with high enough degree of freedom is going to look very much like standard Gaussian. (At the same time S_n becomes a better estimator of σ^2.)

With Student T test, let $X_1, \dots X_d \sim \mathbb{N}(\mu, \sigma^2)$ for some unknown $\mu \in \mathbb{R}$ and let μ_0 be fixed and given.
We want to test $H_0 : \mu = \mu_0$ vs $H_1 : \mu \neq \mu_0$ with asymptotic level $\alpha \in (0, 1)$.
If $sigma^2$ is known, let $T_n = \sqrt{n}\frac{\bar{X_n} - \mu_0}{\sigma}$, then $T_n \sim \mathbb{N}(0, 1)$ and $\psi_\alpha = \mathbb{I}\{|T_n| > q_{\frac{\alpha}{2}}\}$ is a test with level α.
If $sigma^2$ is unknown, let $T_n = \sqrt{n-1}\frac{\bar{X_n} - \mu_0}{\sqrt{S_n}}$, then $T_n \sim t_{n-1}$, Student's T distribution with $n-1$ degrees of freedom. And $\psi_\alpha = \mathbb{I}\{|T_n| > q_{\frac{\alpha}{2}}\}$ is a test with level α (reading from a different table from standard Gaussian, parameterized on n).

T test is not asymptotic and can be run small samples, but still relies on the assumption that the sample is Gaussian.

2.23 Wald's test

Consider a statistical model $(E, (\mathbb{P}_\theta)_{\theta \in \Theta})$ where $\Theta \subset \mathbb{R}^d$ $(d \geq 1)$ and let θ_0 be fixed and given.
Consider the hypotheses $H_0 : \theta = \theta_0$, $H_1 : \theta \neq \theta_0$
Wald's test says something about the MLE estimator.
Under H_0

$$\sqrt{n}\ I(\hat{\theta}^{MLE})^{\frac{1}{2}}\ (\hat{\theta}_n^{MLE} - \theta_0) \xrightarrow[n\to\infty]{(d)} N(0, I(\theta_0)^{-1})\ w.r.t\ \mathbb{P}_{\theta_0}$$

Where I is the Fisher information matrix.

Taking squared of the first part,

$$T_n = n\ (\hat{\theta}_n^{MLE} - \theta_0)^T\ I(\hat{\theta}^{MLE})\ (\hat{\theta}_n^{MLE} - \theta_0) \xrightarrow[n\to\infty]{(d)} \chi_d^2\ w.r.t\ \mathbb{P}_{\theta_0}$$

And Wald's test with asymptotic level α is $\psi = \mathbb{I}\{T_n > q_\alpha\}$. Where q_α is the $(1-\alpha)$-quantile of χ_d^2.

Intuition: given $\theta^{MLE} = (1.2, 0.9, 2.1)$ and $\theta_0 = (1, 1, 2)$, to see if we can reject H_0, instead of calculating $(1.2-1)^2 + (0.9-1)^2 + (2.1-2)^2$, each term should have some weight according to the Fisher Information matrix (and cross terms if Fisher Information matrix is not diagonal). Namely, to answer what it means for two points to be close in a space curved by Fisher Information matrix.

2.24 Likelihood ratio test

There are multiple estimators to choose from (say, MLE and method of moments), and multiple tests one can do. They sometimes agree, and other times differ.

If we are to test $H_0 : \theta = \theta_0$ vs $H_1 : \theta = \theta_1$ (it's very rare that in real life that we'll have hypotheses of this form), we test if θ_0 is more likely than θ_1. Or, the likelihood $\frac{L(X_1,...,X_n,\theta_1)}{L(X_1,...,X_n,\theta_0)} > C$, where C we can calibrate based on the desired level.

2.25 χ^2-test on multinomial distribution

(We build a test statistic that has χ^2 distribution.)
Given a statistical model with discrete event events space and a model identified by pmf P (multinomial distribution), this tests if $H_0 : P = P_0$ vs $H_1 : P \neq P_0$.
E.g. test if zodiac signs of (sampled) wealthiest people follow a uniform distribution. Or test if a sampled set of jurors are representative of the racial demographic breakdown of the entire country.

2.26 Testing goodness of fit

Student T-test requires the samples to be Gaussian. What if I want to test my data is Gaussian?
χ^2-test does something a goodness of fit test for a distributed pmf.
p-value is a random variable (its formula using the data we plug in stays constant, but the data we plug in is random.)
E.g. with Gaussian r.v. and $H_0 : \mu = 0$ vs $H_1 : \mu \neq 0$, Say we know σ^2, p-value is $P(|\sqrt{n}\frac{\bar{X_n}}{\sigma}| > |\sqrt{n}\frac{\bar{x_n}}{\sigma}|) = 2\ \Phi(-\sqrt{n}\frac{\bar{x_n}}{\sigma})$, (the probability of the population being more extreme than the observed, in which case we'd be wrong if we reject H_0.)
Where Φ is the CDF of standard Gaussian, X_n represents the population and x_n represents the observed. Then p-value under H_0 will have a uniform distribution. (Meaning, if we are to generate lots of test sets, then under H_0 their p-values (i.e. probability of generating a test statistic more extreme than this observed) should have a uniform distribution)
GoF tests don't have parametric modeling.
We look at CDF $F(t) = \mathbb{P}(X \leq t) = \mathbb{E}[\mathbb{I}(X \leq t)]$ instead.
The **Empirical CDF**, $F_n(t)$ is an estimator of CDF by replacing the expectation with average.

$$F_n(t) = \frac{1}{n}\ \sum_{i=1}^{n} \mathbb{I}(X_i \leq t) = \frac{1}{n}\ \sum_{i=1}^{n} Bern(F(t))$$

(i.e. the proportion of X_is that are $\leq t$)
And by strong law LLN, for all $t \in \mathbb{R}$, $F_n(t) \xrightarrow[n\to\infty]{a.s.} F(t)$.
And this stronger Glivenko-Cantelli Theorem holds:

$$\sup_{t\in\mathbb{R}} |F_n(t) - F(t)| \xrightarrow[n\to\infty]{a.s.} 0$$

(i.e. LLN holds for any one t, and Glivenko-Cantelli Theorem holds for all t at the same time.)

And by CLT, plugging in the variance of $Bern(F(t))$, for all $t\ in \mathbb{R}$,

$$\sqrt{n}\ (F_n(t) - F(t)) \xrightarrow[n\to\infty]{a.s.} \mathbb{N}(0, F(t)(1-F(t)))$$

Furthermore, Donsker's Theorem suggests the above is not only Gaussian with the specified variance, but also the sup over all ts is bound by a Brownian bridge.

Using this bound, we can build **Kolmogorov-Smirnov test**: let $X_1, \ldots, X_n$ be iid rvs with unknown CDF F and let F_0 be a continuous CDF.
Consider these two hypotheses $H_0 : F = F_0$ vs $H_1 : F \neq F_1$.
Let F_n be the empirical cdf of the sample $X_1, \ldots X_n$, if $F = F_0$ then $F_n(t) \approx F_0(t)$ for all $t \in [0, 1]$.
Let test statistic $T_n = \sup_{t\in\mathbb{R}} \sqrt{t}|F_n(t) - F_0(t)|$.
And KS test with asymtotic level α is

$$\delta_\alpha^{KS} = \mathbb{I}\{T_n > q_\alpha\}$$

where q_α is the $(1-\alpha)$-quantile of Z, the sup of a Brownian bridge.

(**Intuition**: by Donsker theorem, if H_0 holds, the empirical CDF at any point can only deviate from CDF in H_0 by a bounded amount, and we test the sup of the deviation.)

p-value of KS test is $\mathbb{P}[Z > T_n|T_n]$.
In practice, $F_n(t)$ is an incontinuous step function going up (at each ordered $X_{(i)}$), and to compare it with the F_0 continuous CDF we only need compare the left and right points of each observation. This turns the sup into a max.

KS-test offers one way to measure distance between two functions, i.e.

$$d(F_n, F) = \sup_{t\in\mathbb{R}} |F_n(t) - F(t)|$$

There are other ways to measure distance between functions, leading to different tests. E.g.
Cramer-Von Mises uses L^2 norm

$$d^2(F_n, F) = \int_{\mathbb{R}} [F_n(t) - F(t)]^2 dt$$

Anderson-Darling uses

$$d^2(F_n, F) = \int_{\mathbb{R}} \frac{[F_n(t) - F(t)]^2}{F(t)(1-F(t))} dt$$

Another problem in practice is we want F_0 to be fully specified, say Gaussian, what do we use for its μ and σ^2?

We could use sample mean and sample variance, but this tilts the test in favor of H_0, and limits the power of your test (i.e. your ability to reject H_0 just because of wrong μ, σ^2)
In this case Donsker theorem does not hold, and it is a mistake to check the test statistic with a value bounded by Brownian Bridge.
Instead, one can show that the test statistic

$$\sup_{t\in\mathbb{R}} |F_n(t) - \Phi_{\hat{\mu},\hat{\sigma^2}}(t)|$$

(Kolmogorov-Lilliefors test) does not depend on actual μ and σ^2, essentially when pluggin in $\hat{\mu}$ and $\hat{\sigma^2}$ we can check against a more strict table.
This test is of lower power than Kolmogorov-Smirnov test (heavier tail than KS test).

2.27 Quantile-quantile plots

What makes a Gaussian desirable is usually tail decaying at $e^{-\frac{x^2}{2}}$ rate, and having some fluctuations around the peak doesn't matter.
QQ plots is a visual way to perform GoF test. (if the plot F_n is close to that of F, or if F_n^{-1} is close to F^{-1})
In fact, we check if the scatterplot of points
$(F^{-1}(\frac{1}{n}), F_n^{-1}(\frac{1}{n}))$, $(F^{-1}(\frac{2}{n}), F_n^{-1}(\frac{2}{n}))$, $\ldots$, $(F^{-1}(\frac{n-1}{n}), F_n^{-1}(\frac{n-1}{n}))$ are near the line $y = x$. (Checking against standard Gaussian CDF or a $\Phi(\hat{\mu}, \hat{\sigma^2})$ just results in the straight line having an intercept / different slope.)
Having points below the $y = x$ when $(x > 0)$ means empirical distribution having a lighter tail than the Gaussian, when $x < 0$, the opposite. An S curve compared with the straight line then shows lighter tail than Gaussian on both ends, and lighter tails indicate better regulated noise.
(Quantile - inverse of CDF, hence the name quantile-quantile comes from comparing the inverse of F and F_n.)
The piecewise function F_n is not inversible, but we define $F^{-1}(\frac{i}{n}) = X_{(i)}$

2.28 χ^2 goodness-of-fit test

In the finite case,
Let $X_1, \ldots, X_n$ be iid rv on some finite space $E = \{a_1, \ldots, a_K\}$ with some probability measure $\mathbb{P}$.
Let $((\mathbb{P}_\theta)_{\theta\in\Theta})$ be a parametric family of probability distributions on E.
Consider the two hypotheses $H_0 : \mathbb{P} \in (\mathbb{P}_\theta)_{\theta\in\Theta}$ vs $H_1 : \mathbb{P} \notin (\mathbb{P}_\theta)_{\theta\in\Theta}$
Testing H_0 means testing whether the statistical model $(E, (\mathbb{P}_\theta)_{\theta\in\Theta})$ fits the data (e.g. whether data is indeed drawn from a family of binomial distribution).
H_0 is equivalent to $p_j = p_j(\theta)$, $\forall j = 1, \ldots, K$ for some $\theta \in \Theta$.

Let $\hat{\theta}$ be the MLE of θ when assuming H_0 is true.

Let

$$\hat{p}_j = \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}\{X_i = a_j\}$$

If H_0 is true, then $p_j = p_j(\theta)$ so both $\hat{p}_j$ and $p_j(\hat{\theta})$ are good estimators of p_j, and $\hat{p}_j \approx p_j(\hat{\theta})$, $\forall j = 1, \ldots, K$
Define the test statistic

$$T_n = n \sum_{j=1}^{K} \frac{(\hat{p}_j - p_j(\hat{\theta}))^2}{p_j(\hat{\theta})}$$

If H_0 is true, this test statistic converges to a χ^2 distribution

$$T_n \xrightarrow[n\to\infty]{(d)} \chi^2_{K-d-1}$$

Where d is the size of parameter θ. $K - d - 1$ is the degree of freedom.
Intuition: as d becomes bigger, the degree of freedom drops meaning it becomes harder to deviate away from the family of distribution. (This essentially compares the empirical pmf from sample with the pmf of theta estimated with MLE.)

p-value is then $\mathbb{P}[Z > T_n | T_n]$ where $Z \sim \chi^2_{K-d-1}$ and $Z \perp\!\!\!\perp T_n$.

In the infinite case, we partition E into K disjoint bins.
The p_j, $\hat{p}_j$ then becomes the probability of sample falling into bin j.
In practice, deciding how many bins / how to bin to have is ad hoc. It affects the decision you can draw: I can/cannot reject that my data looks like a distribution binned this way.

2.29 Regression

Let X, Y be two real r.v. with two moments and such that $Var(X) \neq 0$.
The theoretical linear regression of Y on X is the best approximation in quadratic means of Y by a linear function of X, i.e. the r.v. $a + bX$ where a and b are two real numbers minimizing $\mathbb{E}[(Y - a - bX)^2]$.
Note that the expectation we are trying to minimize here means minimizing the vertical distance of a point (x, y) to the line $Y = a + bX$.

If we center X on 0 (let $\tilde{X} = X - \mathbb{E}(X)$).

$$\mathbb{E}[(Y - a - bX)^2] = \mathbb{E}[(Y - (a + b\mathbb{E}(X) + b\tilde{X}))^2]$$

Let $\tilde{a} = a + b\mathbb{E}(X)$, and minimize the above by taking derivative,

$$\frac{\partial}{\partial a} = 0 \implies \mathbb{E}[-2(Y - (\tilde{a} + b\tilde{X}))] = 0 \implies \tilde{a} = \mathbb{E}(Y)$$

$$\frac{\partial}{\partial b} = 0 \implies \mathbb{E}[-2\tilde{X}(Y - (\tilde{a} + b\tilde{X}))] = 0 \implies \mathbb{E}[\tilde{X}Y] = b\mathbb{E}(\tilde{X}^2)$$

Since $\tilde{X}$ is centered, $Cov(X, Y) = \mathbb{E}[\tilde{X}Y]$ by definition of covariance, and $Var(X) = \mathbb{E}(X^2)$
Thus $b = \frac{Cov(X,Y))}{Var(X)}$, and we have $a = \mathbb{E}[Y] - b\mathbb{E}[X] = \mathbb{E}[Y] - \frac{Cov(X,Y))}{Var(X)}\ \mathbb{E}[X]$.
(The same closed form can be reached without centering X.)

Intuition: looking at the closed form, the slope is really captured by the covariance relative to the variance of X, and the rest is normalization.
And replacing expectations with average, we plug in the empirical expectation / sample variance and covariance, and get our $\hat{a}$, $\hat{b}$.

Now if we have some noise $Y = a + bX + \epsilon$, for the best fit where $a + b\mathbb{E}(X) = \mathbb{E}(Y)$, we have $\mathbb{E}[\epsilon] = 0$ and $Cov(X, \epsilon) = 0$.
Conversely, we assume $Y = a + bX + \epsilon$ for some $a, b \in \mathbb{R}$ and some centered r.v. that satisfies $Cov(X, \epsilon) = 0$ (if $X \perp\!\!\!\perp \epsilon$ or if $\mathbb{E}[\epsilon|X] = 0$, then $Cov(X, \epsilon) = 0$)
Then $a + bX$ is the theoretical linear regression of Y on X.

The **least squared error** estimator of (a, b) is the minimizer of the sum of squared errors, i.e.

$$\sum_{i=1}^{n}(Y_i - a - bX_i)^2$$

In the multivariate case, $Y_i = X_i\beta + \epsilon_i$, $i = 1, \dots n$.
Where $Y_i \in \mathbb{R}$, $X_i \in \mathbb{R}^p$, and ϵ_i is the noise term satisfying $Cov(X_i, \epsilon_i) = 0$.
Note that we have the first dimension of X being 1, and β_1 is the intercept (a in the univariate case).

Least squared estimator of β is of a similar form

$$\hat{\beta} = \arg\min_{t \in \mathbb{R}^p} \sum_{i=1}^{n}(Y_i - X_i t)^2$$

And written in matrices, we have $Y = X\beta + \epsilon$ where X is a $n \times p$ matrix, and Y and ϵ are dimension n vectors
We have the LSE estimator as

$$\begin{aligned}
\hat{\beta} &= \arg\min_{t \in \mathbb{R}^p} \|Y - Xt\|_2^2 \\
&= \arg\min_{t \in \mathbb{R}^p} \|Y\|_2^2 - \|Xt\|_2^2 + 2Y^T Xt \\
&= \arg\min_{t \in \mathbb{R}^p} \|Y\|_2^2 - t^T X^T Xt + 2Y^T Xt
\end{aligned}$$

Compute the gradient to t (as a column vector) of the above, we have

$$2X^TXt - 2X^TY = 0$$

And the optimal t (the LSE estimator $\hat{\beta}$) is $\hat{\beta} = (X^TX)^{-1}X^TY$, if X^TX is invertible (i.e. X^TX is of rank p).
If X^TX is not invertible, then $\exists v \ \ s.t. X^TXv = 0$, in which case if $\hat{\beta}$ is a solution, so is $\hat{\beta} + \lambda v, \ \forall \ \lambda \in \mathbb{R}$.
(Every time $p > n$, the matrix X^TX cannot be of rank p and is not invertible, and you cannot talk about LSE estimator.)

Also we have that $X\hat{\beta}$ is the orthogonal projection of Y onto the subspace spanned by the columns of X.

$$X\hat{\beta} = PY$$

where $P = X(X^TX)^{-1}X$ (eigenvalue of this vector should be 0 or 1).
And by Pythagorean theorem, we have $\|Y\|_2^2 = \left\|X\hat{\beta}\right\|_2^2 + \left\|Y - X\hat{\beta}\right\|_2^2$.

If we make the following assumptions

- The design matrix X is deterministic and $rank(X) = p$,
- The model is homoscedastic: $\epsilon_1, \ldots, \epsilon_n$ are iid,
- The noise vector ϵ is Gaussian $\epsilon \sim N(0, \sigma^2 I_n)$ where I_n is the unit covariance matrix.

We have linear regression with deterministic design and Gaussian noise, and

$$\begin{aligned}
\hat{\beta} &= (X^TX)^{-1}X^TY \\
&= (X^TX)^{-1}X^T(X\beta + \epsilon) \\
&= \beta + (X^TX)^{-1}X^T\epsilon \\
&= \beta + N_p(0, \sigma^2(X^TX)^{-1}X^TX(X^TX)^{-1}) \\
&= \beta + N_p(0, \sigma^2(X^TX)^{-1})
\end{aligned}$$

And we have these properties

- LSE = MLE. $\hat{\beta} \sim N_p(\beta, \sigma^2(X^TX)^{-1})$
- Quadratic risk of $\hat{\beta}$ $\mathbb{E}[\left\|\hat{\beta} - \beta\right\|_2^2] = \sigma^2 \mathrm{Tr}((X^TX)^{-1})$
- Prediction error $\mathbb{E}[\left\|Y - X\hat{\beta}\right\|_2^2] = \sigma^2(n - p)$

- Reversing the above, we have an unbiased estimator of σ^2: $\hat{\sigma^2} = \frac{1}{n-p}\left\|Y - X\hat{\beta}\right\|_2^2$, and $(n-p)\frac{\hat{\sigma}^2}{\sigma^2} \sim \chi^2_{n-p}$, $\sigma^2 \perp\!\!\!\perp \hat{\beta}$

Where Tr is the **trace** of a matrix: sum of its values on the diagonal. Trace is equal to the sum of eigenvalues.

To prove that LSE = MLE in this case, our model for $Y = X\beta + \epsilon$ is

$$(P_\theta)_{\theta\in\Theta} = \{N_n(X\beta, \sigma^2 I)\}$$

Which is a multivariate Gaussian, and the density is

$$P(y) = \frac{1}{(\sigma^2 2\pi)^{\frac{n}{2}}} e^{-\frac{\|y - X\beta\|_2^2}{2\sigma^2}}$$

Log-likelihood then becomes

$$\log P(y) = -\frac{n}{2}\log \sigma^2 2\pi - \frac{1}{2\sigma^2}\|y - X\beta\|_2^2$$

MLE is achieved when the above is maximized, i.e. when $\|y - X\beta\|_2^2$ is minimized, hence its equality with LSE.

Three pillars of inference: estimator, confidence interval (tests)

Significance in linear regression: to test whether the j-th explanatory variable is significant in the linear regression ($1 \le j \le p$, i.e. this factor should show up in our formula). We test $H_0 : \beta_j = 0$ vs $H_1 : \beta_j \neq 0$
The basis for inference in linear regression is $\hat{\beta} = \beta + N_p(0, \sigma^2(X^T X)^{-1})$, shown above.

We are interested in

$$\hat{\beta}_j = \beta_j + e_j^T N(0, \sigma^2(X^T X)^{-1})$$

where $e_j = (0, 0, \ldots, 1, \ldots, 0)^T$, a vector that leaves only the $j-$th element in.
And

$$e_j^T N(0, \sigma^2(X^T X)^{-1}) \overset{d}{=} N(0, \sigma^2 e_j^T (X^T X)^{-1} e_j) = N(0, \sigma^2 (X^T X)^{-1}_{jj})$$

The j-th diagonal element of the matrix $X^T X^{-1}$.
And

$$\hat{\beta}_j = \beta_j + N(0, \sigma^2 (X^T X)^{-1}_{jj})$$

(This looks similar to how $t-$test is set up, $\hat{\theta} = \theta + N(0, \frac{\sigma^2}{n})$).
(1) If σ^2 is known, we can use Gaussian quantiles:

Under H_0, we have

$$\hat{\beta}_j \sim N(0, \sigma^2 (X^T X)^{-1}_{jj})$$

And

$$\frac{\hat{\beta}_j}{\sigma\sqrt{(X^T X)^{-1}_{jj}}} \sim N(0,1)$$

The test statistic

$$\Phi = I\{|\frac{\hat{\beta}_j}{\sigma\sqrt{(X^T X)^{-1}_{jj}}}| > q_{\frac{\alpha}{2}}(N(0,1))\}$$

(2) If σ is unknown, we use $\hat{\sigma}^2$ and $t-$test.

$$\frac{\hat{\beta}_j}{\hat{\sigma}\sqrt{(X^T X)^{-1}_{jj}}} = \frac{\hat{\beta}_j}{\sigma\sqrt{(X^T X)^{-1}_{jj}}}\sqrt{\frac{\sigma^2}{\hat{\sigma}^2}} = N(0,1)\frac{1}{\sqrt{\frac{\hat{\sigma}^2(n-p)}{\sigma^2}}}\sqrt{n-p} = \frac{N(0,1)}{\sqrt{\frac{\chi^2_{n-p}}{n-p}}} \sim t_{n-p}$$

The test statistic

$$\Phi = I\{|\frac{\hat{\beta}_j}{\hat{\sigma}\sqrt{(X^T X)^{-1}_{jj}}}| > q_{\frac{\alpha}{2}}(t_{n-p})\}$$

The more there are parameters, the heavier the tail, and the harder it becomes to reject H_0.

Now imagine you run significance tests for all β_js and each of them has a 5% Type I error rate, all of them combined you are going to make mistakes if p is large enough.
In this running-multiple-tests-simultaneously case we correct using **Bonferroni correction**.

$H_0 : \beta_j = 0,\ \forall j \in S$ vs $H_1 : \exists j \in S,\ \beta_j \neq 0$ where $S \subseteq \{1, \dots p\}$
The probability of Type I error (one of the above mentioned t-test at level α says reject whereas we shouldn't have rejected H_0) is

$$P_{H_0}(\bigcup_{i \in S}\{\Phi_i = 1\})$$

Which by union bound of probability, is $\leq \sum_{i \in S} P_{H_0}(\{\Phi_i = 1\}) = |S|\alpha$. (equality happens when all sets are disjoint)

And to control Type I error rate $< \alpha$, we instead run each individual t-test with level $\frac{\alpha}{|S|}$, where $|S| = p$ if you care about all the explanatory variables in your H_0.
In reality the individual tests are rarely completely disjoint, and Bonferroni correction can be more conservative (harder to reject H_0) than it needs to be.

In addition to significance tests, there can be more general tests of the form $H_0 : G\beta = \lambda$ vs $H_1 : G\beta \neq \lambda$
Where G is a $k \times p$ matrix with $rank(G) = k(k \leq p)$ and $\lambda \in \mathbb{R}^k$. (This can accommodate e.g. testing $\beta_2 + \beta_3 = 0$)

Under H_0,
$G\hat{\beta} = G\beta + G\epsilon$
$G\hat{\beta} - \lambda \sim N_k(0, \sigma^2 G(X^T X)^{-1} G^T)$
To build a test we want to turn the right hand side to a pivotal that can be looked up.

If $X \sim N(0, \Sigma)$ then $X^T \Sigma^{-1} X \sim \chi^2_k$.
Consequently, we have

$$\frac{(G\hat{\beta} - \lambda)^T (G(X^T X)^{-1} G^T)^{-1} (G\hat{\beta} - \lambda))}{\sigma^2} \sim \chi^2_k$$

Substituting σ^2 with $\hat{\sigma}^2$ as before, the right hand side becomes $\frac{\chi^2_k}{\frac{\chi^2_{n-p}}{n-p}}$
Divide both the lhs and rhs above by k, we have the rhs as $\frac{\chi^2_k/k}{\chi^2_{n-p}/(n-p)}$, which is called **Fisher (F) Distribution** $F_{k,n-p}$ (with k and $n-p$ degrees of freedom, and the two chi^2 are independent)

We have $T^2_q \sim F_{1,q}$, a square root relation between T-distribution and F-distribution. (Just like χ^2 with 1 DoF is square of the Gaussian.)

Then to test H_0 vs H_1, we have

$$\Phi_\alpha = I\{\frac{(G\hat{\beta} - \lambda)^T (G(X^T X)^{-1} G^T)^{-1} (G\hat{\beta} - \lambda))}{\hat{\sigma}^2 k} \sim \chi^2_k > q_\alpha(F_{k,n-p})\}$$

Closing remarks

- looking at the variance of each term β_j, if you get to choose your Xs as orthogonal as you can, you are going to end up with smaller variance (the innate σ^2 times the diagonal of $X^T X^{-1}$).
- linear regression exhibits correlations, not causality. Causation does not come from statistics.
- The properties we analyzed and tests are built upon normality of the noise. Without it, we only have $\hat{\beta} = \beta + \epsilon$

2.30 Bayesian statistics

What the above sections have been doing is **frequentist statistics**, whose approach

- Observe data
- Assume data were generated randomly
- Make assumptions on the generating process (iid, Gaussian, linear regression function, etc)
- The generating process was associated to some object of interest (a parameter, a density, etc)
- This object is fixed and unknown and we want to find it: we either estimated it or tested a hypothesis about this object.

With the Bayesian approach, we still observe data and assume it to be randomly generated by some process. Under some assumptions (e.g. parametric distribution), this process is associated with some fixed object. And we have a **prior belief** about it, using the data, we want to update the belief and transform it into a **posterior belief**.

Example, say we are to estimate the proportion of men in the population.
Let $X_1, \dots X_d \sim Ber(p)$ where X_i are iid and takes the value 1 for men and 0 for women.
With the frequentist approach, we estimate p using the MLE, construct some confidence interval for p, then do hypothesis testing (e.g. $H_0 : p = 0.5$ vs $H_1 : p \neq 0.5$).
However before analyzing the data, we may believe p is likely to be close to $\frac{1}{2}$.

The Bayesian approach is a tool to include mathematically our prior belief in the statistical procedures, and update our prior belief using the data.
In the example, the prior can look like, e.g. we are 90% sure p is between .4 and .6.
Hence we can model our prior belief using a distribution for p as if it's random. (in reality, the true parameter is not random)
This distribution is called the **prior distribution**.

In our statistical experiment, $X_1, \dots X_n$ are assumed to be iid Bernoulli rv with parameter p conditionally on p.
After observing the available sample $X_1, \dots X_n$, we can update our belief about p by taking its distribution conditionally on the data.
The distribution of p conditionally on the data is called the **posterior distribution**.
A **congujate prior** means the prior and posterior have the same family of distribution.

One conjugate prior is $\beta-$distribubtion, $\beta(a,b) = cx^{a-1}(1-x)^{b-1},\ \forall x \in [0,1]$ and $a > 0,\ b > 0$.
Where a prior distribution $\beta(a,a)$ becomes a posterior distribution

$$\beta(a + \sum_{i=1}^{n} X_i, a + n - \sum_{i=1}^{n} X_i)$$

Another conjugate prior is Gaussian. A prior being conjugate brings nice mathematical properties, and people use these as prior distributions more often.

Think of this in terms of Bayesian rule,

$$P(\theta \mid Data) = \frac{P(Data \mid \theta)P(\theta)}{P(Data)}$$

Where $P(\theta)$ is our prior, $P(Data|\theta)$ is what our statistical experiment assumes, and $P(Data)$ says something about the distribution of data itself if drawn completely randomly, note that it has not to do with θ.

Now formalize the above, let $\pi(\theta)$ be the prior distribution, the posterior distribution looks like

$$\pi(\theta \mid X_1, \dots X_n) = \frac{P(X_1, \dots X_n \mid \theta)\pi(\theta)}{\int_\Theta P(X_1, \dots X_n \mid \theta)\pi(\theta)d\theta}$$

where the $P(X_1, \dots X_n \mid \theta)$ part is what we get from our modeling, which 2.17 defined as likelihood of θ.
I.e. the posterior is a likelihood weighted by the prior.

Diving back to the previous example of Bernoulli with prior distribution as $\beta-$distribution.
We have (1)

$$\pi(p) \ \propto \ p^{a-1}(1-p)^{a-1},\ p \in (0,1)$$

(2) Given p, $X_1, \dots X_n \overset{iid}{\sim} Ber(p)$, so

$$P_n(X_1, \dots X_n \mid \theta) = p^{\sum_{i=1}^n X_i}\ (1-p)^{n-\sum_{i=1}^n X_i}$$

So the posterior distribution is

$$\pi(\theta \mid X_1, \dots X_n) \ \propto \ p^{a-1+\sum_{i=1}^n X_i}\ (1-p)^{a-1+n-\sum_{i=1}^n X_i}$$

which is $\beta(a + \sum_{i=1}^n X_i, a + n - \sum_{i=1}^n X_i)$

Now what if we don't really have a prior?
The formula really says prior is there to weigh some θ higher than others, and you can have a prior that weighs no θ higher than others, i.e. replacing $\pi(\theta)$ with a constant, a **non-informative** prior.
If θ is bounded, this is the uniform prior on Θ.
If not, this does not define a proper pdf on Θ (improper prior, using which we can still do Bayesian statistics).

With a non-informative prior in the Bernoulli with β-distribubtion prior example, the posterior becomes $\beta(1 + \sum_{i=1}^n X_i, 1 + n - \sum_{i=1}^n X_i)$.

Now with a non-informative prior on a Gaussian: given θ, $X_1, \dots X_n \overset{iid}{\sim} N(\theta, 1)$,

$$p_n(X_1, \dots X_n \mid \theta) = \frac{1}{(\sqrt{2\pi})^n} \; exp(-\frac{1}{2}\sum_{i=1}^n (X_i - \theta)^2)$$

And the posterior (note that we now see this as a density on θ and not on X_i)

$$\begin{aligned}
\pi(\theta \mid X_1, \dots X_n) \; &\propto \; exp(-\frac{1}{2}\sum_{i=1}^n (X_i - \theta)^2) \\
&\propto \; exp(-\frac{1}{2}\sum_{i=1}^n (-2X_i\theta + \theta^2)) \\
&\propto \; exp(-\frac{n}{2}(\bar{X} - \theta)^2) \\
&\sim \; N(\bar{X}, \; \frac{1}{n})
\end{aligned}$$

Another generic non-informative prior, **Jeffreys prior** is

$$\pi_J(\theta) \; \propto \; \sqrt{\det \; I(\theta)}$$

where $I(\theta)$ is the Fisher information matrix of the statistical model associated with $X_1, \dots X_n$ in the frequentist approach.
(i.e. I want to put more weights on θ that extracts information from the data.)

Jeffreys prior satisfies reparametrization invariance principle, when you parameterize you model on a $\eta(\theta)$ instead of θ, then your Jeffreys prior is just

$$\tilde{\pi}(\eta) \; \propto \; \sqrt{\det \; \tilde{I}(\eta)}$$

where $\tilde{I}(\eta)$ is the Fisher information of the statistical model parameterized by η instead of θ.

When doing Bayesian inference, given $X_1, \dots X_n$ and a prior $\pi(\theta)$, it spits out not just an estimator $\hat{\theta}$, but an entire posterior distribution $\pi(\theta|X_1, \dots X_n$, from which you can build a confidence region.
The **Bayesian confidence region** with level $\alpha \in (0, 1)$ is a random subset R of the parameter space Θ, which depends on the sample $X_1, \dots X_n$ such that

$$\mathbb{P}[\theta \in R \mid X_1, \dots X_n] = 1 - \alpha$$

Note that R depends on the prior distribution.
And that **a Bayesian confidence region and a frequentist confidence interval are two distinct notions**.
In a frequentist view, an example confidence interval could suggest

$$\mathbb{P}(\theta \in [\bar{X_n} - q_{\frac{\alpha}{2}}, \bar{X_n} + q_{\frac{\alpha}{2}}]) < 95\%$$

And we don't condition on $X_1, \dots X_n$ in this case because if we do, the probability is either 1 (if $\theta \in$ the region) or 0 (if not).
I.e. the frequentist view is saying, if we draw one set of $X_1, \dots X_n$, we get a $\bar{X_n}$ and a confidence interval centered around $\hat{\theta}$, true θ is either in or not in there. And to have a confidence interval at 95% in the frequentist view means if we draw these samples repeatedly, the frequency of times of the confidence interval we build containing θ is 95%.

While the Bayesian confidence region conditions on the data: given this particular set of n samples, it suggests how likely is true θ within this region.

When sample size is large enough, Bayesian methods tends to have the same behavior as a frequentist confidence interval. (the one extra sample, your prior, doesn't matter as much.)
Bayesian methods are particularly useful for one-time draw of samples, and on small sample sizes.

A concrete example: say we have draw once 2 iid samples $X_1 = \theta + \epsilon_1$, $X_2 = \theta + \epsilon_2$, $\epsilon \in \{-1, 1\}$ with equal probability.
Thus the possible observations are $(\theta - 1, \theta - 1)$, $(\theta - 1, \theta + 1)$, $(\theta + 1, \theta - 1)$, $(\theta + 1, \theta + 1)$.
Say we have a Frequentist confidence interval R as defined by

$$\begin{cases} X_1 - 1, \text{ if } X_1 = X_2 \\ \frac{X_1 + X_2}{2}, \text{ if } X_1 \neq X_2 \end{cases} \tag{1}$$

Then this confidence interval has level 75%.
Note that if we condition on the data, then in the second case true θ is always within the region, while in the first only half of the time it is.

Now if we have a Bayesian confidence region instead, say the prior is $\pi(j) > 0, \ \forall j \in \mathbb{Z}$.

And say that we observe $(5, 7)$, then

$$\mathbb{P}((5,7) \mid \theta) = \begin{cases} \frac{1}{4}, \text{ if } \theta = 6, \\ 0, \text{ otherwise} \end{cases} \tag{2}$$

And the posterior

$$\begin{aligned} \mathbb{P}(\theta \mid (5,7)) &= \frac{\mathbb{P}((5,7) \mid \theta)\mathbb{P}(\theta)}{\mathbb{P}((5,7))} \\ &= \frac{\mathbb{P}((5,7) \mid \theta)\mathbb{P}(\theta)}{\mathbb{P}((5,7))\mathbb{P}(\theta = 6) + \mathbb{P}((5,7))\mathbb{P}(\theta \neq 6)} \\ &= \begin{cases} 1, \text{ if } \theta = 6, \\ 0, \text{ otherwise} \end{cases} \end{aligned}$$

And in this sense the Bayesian confidence interval is more meaningful.

The Bayesian framework can also be used to estimate the true underlying parameter, in which case the prior distribution does not reflect a prior belief, but just an artificial tool to define a new class of estimators.
A **Bayes estimator** is defined as the posterior mean:

$$\hat{\theta}^{(\pi)} = \int_{\Theta} \theta d\pi(\theta \mid X_1, \dots X_n)$$

which depends on the choice of the prior distribution.
We can also use other estimators from the posterior, e.g. its mode.

In the previous example where prior is $\beta(a, a)$, the posterior mean $\hat{p}^{(\pi)} = \frac{a/n + \bar{X_n}}{2a/n+1}$, which is independent of a as $n \to \infty$.

With the non-informative prior on a Gaussian example ($N(\theta, 1)$), the posterior is $N(\bar{X_n}, \frac{1}{n})$, $\hat{p}^{(\pi)} = \bar{X_n}$.
Bayes estimator in this case is consistent and asymptotic normal.
In general, the asymptotic properties of Bayes estimator does not depend on the choice of the prior.

2.31 Principal component analysis

The goal is to project a point cloud onto lower dimensions while keeping as much information possible (we don't want points to collide). PCA's answer is to do this by keeping as much covariance structure as possible by keeping orthogonal directions that discriminate well the points of the cloud.

The linear algebra that supports PCA:
Let X be a d-dimensionral random vector and $X_1, \dots X_n$ be n independent copies of X.
Denote by $\mathbf{X}$ the random $n \times d$ matrix (n rows. $X_1, \dots X_n$ stacked together vertically).
Mean of $\mathbf{X}$: $\mathbb{E}[\mathbf{X}] = (\mathbb{E}[X^1], \dots \mathbb{E}[X^d])^T$.
Covariance matrix of $\mathbf{X}$ ($d \times d$) is the matrix $\Sigma = (\sigma_{j,k})$ where $j, k \in \{1 \dots d\}$, and $\sigma_{j,k} = cov(\mathbf{X}^j, \mathbf{X}^k)$. (The covariance between the mean of the $j-$th and $k-$th dimensions, which by definition is $\mathbb{E}[X^j X^k] - \mathbb{E}[X^j]\mathbb{E}[X^k]$). And expanding on this definition:

$$\Sigma = \mathbb{E}[\mathbf{X}\mathbf{X}^T] - \mathbb{E}[\mathbf{X}]\mathbb{E}[\mathbf{X}]^T = \mathbb{E}[(\mathbf{X} - \mathbb{E}[\mathbf{X}])(\mathbf{X} - \mathbb{E}[\mathbf{X}])^T]$$

We then apply the same trick of replacing expectation with average over our samples:
The empirical mean is $\bar{\mathbf{X}} = \frac{1}{n}\sum_{i=1}^n X_i$.
And the empirical covariance matrix (sample covariance matrix) is

$$\begin{aligned} S &= \frac{1}{n}\sum_{i=1}^n X_i X_i^T - \bar{\mathbf{X}}\bar{\mathbf{X}}^T = \frac{1}{n}\sum_{i=1}^n (X_i - \bar{\mathbf{X}})(X_i - \bar{\mathbf{X}})^T \\ &= \frac{1}{n}\mathbf{X}^T\mathbf{X} - \frac{1}{n^2}\mathbf{X}^T\mathbb{I}\,\mathbb{I}^T\mathbf{X} \\ &= \frac{1}{n}\mathbf{X}^T\ H\ \mathbf{X} \end{aligned}$$

Where $\mathbb{I}$ is the $n \times 1$ all 1 vector, $H = I_n - \frac{1}{n}\mathbb{I}\,\mathbb{I}^T$, and I_n is the $n \times n$ identity matrix.
And H is in fact the matrix with diagonal $1 - \frac{1}{n}$ and all other elements $-\frac{1}{n}$. It is an orthogonal projector: $H^2 = H$, $H^T = H$.
Consider what H projects onto, for a $n \times 1$ vector V,

$$HV = V - \bar{V}\mathbb{I}$$

and $\bar{HV} = 0$.
Meaning H projects onto the subspace of vectors that have mean 0, i.e. $V \perp\!\!\!\perp \text{span}(\mathbb{I})$.

Given the above, $S = \frac{1}{n}\mathbf{X}^T\ H^T H\ \mathbf{X} = \frac{1}{n}(H\mathbf{X})^T(H\mathbf{X})$.

Take $u \in \mathbb{R}^d$, by definition of Σ, $u^T\Sigma u = \mathbb{E}[(u^T\mathbf{X})(\mathbf{X}^T u)] - \mathbb{E}[(u^T\mathbf{X})]\mathbb{E}[(\mathbf{X}^T u)]$, which is the variance of $u^T\mathbf{X}$.
Similarly, $u^T S u$ (a number) is the sample variance of $u^T\mathbf{X_1}, \dots u^T\mathbf{X_n}$.

Think of the $u^T\Sigma u$ as the variance you get, after projecting the matrix $\mathbf{X}$ onto the direction u.

If we want to do dimensionality reduction by projecting our points onto a lower dimension, and we don't want the projected points to spread as much as possible. Variance is one measurement of spread, and in this case we want a variance as large as possible.

And the idea of primary component analysis is to identify these directions along which we have a high variance.
If we can find a $u \in \mathbb{R}^d$ and $\|u\|_2 = 1$ that maximizes $u^T S u$, i.e. empirical variance of the points projected onto the direction u.
This maximization is not convex, but we can solve it with linear algebra.
In particular, Σ and S are symmetric and positive semi-definite.
Theorem (spectral decomposition / eigen decomposition): any real matrix $A \in \mathbb{R}^{d \times d}$ has the decomposition $A = PDP^T$, where P is a $d \times d$ orthogonal matrix, i.e. $PP^T = P^T P = I_d$ and D is diagonal.
The diagonal elements of D are the eigenvalues of A and the columns of P are the corresponding eigenvectors of A.

A is semi-definite positive iff all its eigenvalues are non-negative.

To do PCA, by the above theorem, let $S = PDP^T$, we know D is diagonal.
And $P = (v_1, \dots v_d)$, an orthogonal matrix i.e. $\|v_j\|_2 = 1$, $v_j^T v_k = 0, \forall v \neq k$.
Without loss of generality, let $\lambda_1 \geq \lambda_2 \cdots \geq \lambda_d$ be the diagonal values of D.

Note that D is empirical covariance matrix of the $P^T X_i$'s, $i = 1 \dots n$.
In particular, λ_1 is the empirical variance of $v_1^T X_i$'s.
In other words, each λ_i measures the spread of the point cloud in the direction v_j.

And the problem of maximizing empirical variance post-projection (i.e. find the u that maximizes the variance of $u^T X_1 \dots u^T X_n$) becomes picking the u associated with the largest λ_i, which is assumed to be λ_1, thus $u = v_1$.
We can prove that for any vector (including non-eigenvector of S) other than v_1, the resulting empirical variance will be no bigger.

Thus, if PCA wants to reduce the dimension to k, we take the k orthogonal directions in which the cloud is the most spread out, and these k directions correspond exactly to the eigenvectors associated with the k largest eigenvalues of S.

If you have higher dimensionality than points, then this method does not apply.
Otherwise, the above maths can be turned into a PCA algorithm involving

- first calculate the empirical covariance matrix of input point cloud $\mathbf{X} \in \mathbb{R}^{n \times d}$,
- then do a spectral decomposition on it,

- use the k eigenvectors of the empirical covariance matrix associated with the k-largest eigenvalues to build the projection matrix $P_k = (v_1, \dots v_k) \in \mathbb{R}^{d \times k}$. The $(v_1, \dots v_k)$ are called principal components.
- The resulting point cloud $\mathbf{Y} = \mathbf{X}P_k \in \mathbb{R}^{n \times k}$

How to choose k? Multiple ways

- Experimental rule: take k where there is an inflection point in the decreasing sequence $\lambda_1 \dots \lambda_d$ (inflection point of two lines going through two clusters of decreasing points in the scree plot; roughly, after this point you stopped gaining a lot of variance)
- Define a criterion: take k such that $\frac{\lambda_1 + \dots + \lambda_k}{\lambda_1 + \dots + \lambda_n} \geq 1 - \alpha$. For some $\alpha \in (0, 1)$ that determines the approximation error. The numerator is called variance explained by the PCA and denominator (the Trace of S.) is called total variance.
- Or for visualization, take $k = 2$ or 3.

In practice, PCA is used to reduce dimensionality of point clouds that are not necessarily random. In statistics if $n \gg d$, the empirical covariance matrix S is a consistent estimator of population covariance matrix Σ. In many applications such as gene expression, $n \ll d$ but the rank of Σ could be fairly low, and one can do sparse PCA first before regression.

2.32 Generalized Linear Model

A linear model says $Y = X^T\beta + \epsilon$.
If $\epsilon \sim N(0,\ \sigma^2 I)$, $Y|X \sim N(\mu(x),\ \sigma^2 I)$, where $\mu(x) = \mathbf{X}^T\beta$ in the linear case.
The two components are

- Random component: the response variable $Y|X$ is continuous and normally distributed with $\mu = \mu(x) = \mathbb{E}(Y|X)$
- Link: between the random and covariates $X = (X^{(1)}, \dots X^{(p)})^T$: $\mu(X) = X^T\beta$.

In GLM, we relax the random component to be $Y|X \sim$ some exponential family distribution.
And we relax the link component to $g(\mu(X)) = X^T\beta$, where g is called link function.

Example: modeling the expected number of new cases (μ_i) on day i for the early stages of a disease epidemic, assuming new cases increase exponentially,

$$\mu_i = \gamma e^{(\delta t_i)}$$

This model can be turned into GLM form by using a log link:

$$\log \mu_i = \log \gamma + \delta t_i = \beta_0 + \beta_1 t_i$$

Since this is a count, the Poisson distribution with expected value μ_i is probably a reasonable distribution.

Another example, $\mu_i = \frac{\alpha x_i}{h+x}$ for a concave $(0,1)$ plateauing at some value, we can use a reciprocal link to turn this into linear.

In the Gaussian case, we have a closed form for LSE, and MLE is LSE, but in these more general cases, we need a different model fitting method.

A family of distribution $\{P_\theta : \theta \in \Theta\}$, $\Theta \subset \mathbb{R}^k$ is said to be a **k-parameter exponential family** on $\mathbb{R}^k$, if there exist real valued functions $\eta_1, \dots \eta_k$ and B of θ, $T_1, \dots T_k$ and h of $x \in \mathbb{R}^q$ s.t. the identity function (pmf or pdf) of ¶$_\theta$ can be written as

$$p_\theta(x) = \exp[\sum_{i=1}^{k} \eta_i(\theta) T_i(x) - B(\theta)] h(x)$$

Intuition is to constrain ways θ and x can interact to $p_\theta(x) = \exp(\theta x) f(\theta) g(x)$. $f(\theta)$ is pushed into the exp as $B(\theta)$, which functions like a normalizing factor. $g(x)$ ($h(x)$ in the formula) allows us to account for discrete and continuous x, as if the density is expressed on a transformed x.

We then want to show a Gaussian is of the exponential family: $X \sim N(\mu, \sigma^2)$, $\theta = (\mu, \sigma^2)$, $p_\theta = \frac{1}{\sigma\sqrt{2\pi}} \exp(-\frac{(x-\mu)^2}{2\sigma^2})$.
This holds when $T_1(x) = x$, $\eta_1(\theta) = \frac{\mu}{\sigma^2}$, $T_2(x) = x^2$, $\eta_2(\theta) = -\frac{1}{2\sigma^2}$, $B(\theta) = \frac{\mu^2}{2\sigma^2} + \log\left(\sigma\sqrt{2\pi}\right)$, $h(x) = 1$. Note that this is not the only solution, the constant term can go into B or h.

When σ^2 is known, $\theta = \mu$ and it becomes a one-parameter exponential family on $\mathbb{R}$. $\eta = \frac{\mu}{\sigma^2}$, $T(x) = x$, $B(\theta) = \frac{\mu^2}{2\sigma^2}$, $h(x) = \frac{\exp(-\frac{x^2}{2\sigma^2})}{\sigma\sqrt{(2\pi)}}$ The term before $h(x)$ explains how the Gaussian scales with respect to a standard Gaussian. And $B(\theta)$ carries information about the mean, likelihood, etc (shown later).

Discrete distributions, such as Bernoulli and Poisson, are also of this family. As is the continuous distributions, Γ, inverse Γ, inverse Gaussian, χ^2, β, binomial and negative binomial. The uniform distribution is not of this family.

Canonical exponential family for $k = 1$, $y \in \mathbb{R}$ is

$$f_\theta(y) = \exp(\frac{y\theta - b(\theta)}{\phi} + c(y, \phi))$$

for some known functions b and c.
If ϕ is known, this is a one-parameter exponential family with θ being the canonical parameter. If not, this may or may not be a two-parameter exponential family. ϕ is called the dispersion parameter.

The rest assumes ϕ is known.

Expectation of the first derivative of the log likelihood function looks like:

$$\begin{aligned}
\mathbb{E}_\theta(\frac{\partial \log f_\theta(x)}{\partial \theta}) &= \mathbb{E}_\theta[\frac{\frac{\partial}{\partial \theta} f_\theta(x)}{f_\theta(x)}] \\
&= \int \frac{\frac{\partial}{\partial \theta} f_\theta(x)}{f_\theta(x)} f_\theta(x) dx \\
&= \int \frac{\partial}{\partial \theta} f_\theta(x) dx \\
&= \frac{\partial}{\partial \theta} \int f_\theta(x) dx \\
&= \frac{\partial}{\partial \theta} 1 \\
&= 0
\end{aligned}$$

Similarly, we can derive that the expectation of the second derivative of the log likelihood function $\mathbb{E}_\theta[\frac{\partial^2}{\partial \theta^2} \log f_\theta(x)] = \mathbb{E}_\theta[(\frac{\partial \log f_\theta(x)}{\partial \theta})^2]$.

Let $l(\theta) = \log(f_\theta(Y))$ denote the log-likelihood function.
The mean $\mathbb{E}(Y)$ and variance $\text{var}(Y)$ can be derived using the above conclusions: $\mathbb{E}(\frac{\partial l}{\partial \theta}) = 0$, and $\mathbb{E}(\frac{\partial^2 l}{\partial \theta^2}) + \mathbb{E}(\frac{\partial l}{\partial \theta})^2 = 0$ obtained from $\int f_{\theta(y)} dy = 1$.
Note that $l(\theta) = \frac{Y\theta - b(\theta)}{\phi} + c(Y; \phi)$, therefore $0 = \mathbb{E}(\frac{\partial l}{\partial \theta}) = \frac{\mathbb{E}(Y) - b'(\theta)}{\phi}$, and $\mathbb{E}(Y) = \mu = b'(\theta)$.
Similarly, $\text{var}(Y) = b''(\theta)\phi$. Note that variance is positive, and this means $b(\theta)$ is convex.

For example, Poisson distribution $f(y) = \frac{\mu^y}{y!} e^{-\mu}$ written in canonical form has $b(\theta) = \mu$, $\theta = \log \mu$, $c(y, \phi) = -\log(y!)$, $\phi = 1$. Thus $b(\theta) = b'(\theta) = b''(\theta) = e^\theta = \mu$, which is the mean and variance of Poisson distribution.

However, in GLM we don't care about $\mathbb{E}[Y]$, but rather $\mathbb{E}[Y|X]$, and this is where the link function comes in.
We want the link function g to be strictly increasing, continuously differentiable, $g(\mu)$ spans over $\mathbb{R}$ (image of the function g).
These imply g^{-1} exists and increasing.

Some examples of link functions include

- identity function for linear models.
- log for Poisson data ($Y|X \sim Poisson(\mu(X)), \mu(X) > 0$, link function should map $(0, +\infty)$ to $\mathbb{R}$).

- $\log\left(\frac{\mu(X)}{1-\mu(X)}\right)$ for Bernoulli / Binomial data where $0 < \mu < 1$ and g should map $(0, 1)$ to $\mathbb{R}$ (logit, logistic regression). Or we can use the inverse of normal CDF (probit), or a complementary log-log function.

The function g that links the mean μ to the canonical parameter θ is called the **canonical link**. ($\mu = b'(\theta)$)

$$g(\mu) = \theta = (b')^{-1}(\mu)$$

Ultimately we want to express our log likelihood function using β (the parameter we are estimating in order to maximize likelihood), and β is connected with $\mu(x)$ ($\mathbb{E}(Y|X)$) via $g(\mu(x)) = \beta^T X$, and $\mu(x)$ with θ via b' or $(b')^{-1}$.
So we have

$$\theta = (b')^{-1}(\mu) = (b')^{-1}(g^{-1}(\mu)) = (b')^{-1}(g^{-1}(X^T\beta))$$

If we have n independent pairs of observation $(X_i, Y_i) \in \mathbb{R}^p \times \mathbb{R}$, $i = 1 \dots n$. The conditional distribution of Y_i given $X_i = x_i$ has density in the canonical exponential family:

$$f_{\theta_i}(y_i) = \exp(\frac{y_i\theta_i - b(\theta_i)}{\phi} + c(y_i, \phi))$$

and $\theta_i = (b')^{-1}(g^{-1}(X_i^T\beta))$.
Note that for different observations (X_i, Y_i), β does not depend on i.

Denote the function $(b')^{-1}(g^{-1}(X_i^T\beta))$ as $h(X_i^T\beta)$.
(If g is the canonical link function, h is identity.)

The log-likelihood is then

$$l_n(\beta; Y, X) = \sum_i \frac{Y_i h(X_i^T\beta) - b(h(X_i^T\beta))}{\phi}$$

When we use the canonical link function g, we get

$$l_n(\beta, \phi; Y, X) = \sum_i \frac{Y_i X_i^T\beta - b(X_i^T\beta)}{\phi}$$

The first term is linear to β, which is both convex and concave, and when added on to the second term, does not affect the convexity of the entire function.

We know that $b''(\theta)$ is positive, the second term is strictly convex, then $l_n(\beta, \phi; Y, X)$ is (strictly) concave.
Hence when using canonical link, the MLE estimator is unique.

Now, to optimize l_n (a strictly concave function).
We have **Newton-Raphson Method**, a black box. This performs quadratic approximation (second-order Taylor expansion).
Suppose f has a continuous Hessian map at x_0. Then we can approximate f quadratically in a neighborhood of x_0 using

$$f(x) \approx f(x_0) + \nabla_f^T(x_0)(x - x_0) + \frac{1}{2}(x - x_0)^T H_f(x_0)(x - x_0)$$

Where $\nabla_f(x_0)$ is the gradient of f at x_0, and $H_f(x_0)$ is the Hessian (second order partial derivative matrix) of f at x_0, which is negative definite.
Maximizing the right hand side (its derivative $\nabla = 0$), we have

$$\nabla_f(x) = \nabla_f(x_0) + H_f(x_0)(x - x_0) = 0$$

And the x^* that maximize the right hand side is

$$x^* = x_0 - H_f(x_0)^{-1}\nabla_f(x_0)$$

This is a Newton iteration: starting at an arbitrary x_0, we use the above to get to x^* that maximizes the right hand side, and keep iterating until convergence. Meaning

$$x^{(k+1)} = x^{(k)} - H_f(x^{(k)})^{-1}\nabla_f(x^{(k)})$$

This second-order method (requiring a known Hessian) is more powerful than Gradient Descent, in practice inverting the matrix H is extremely painful, and people use pseudo Hessian inverse to emulate the Hessian.

Newton-Raphson method is general enough (works for a deterministic case without random data), however exploiting the structure of our specific GLM maximum likelihood optimization problem, computation can be made easier by replacing the Hessian with its expected value.

The idea is to replace the Hessian with its expected value. Recall that $\mathbb{E}_\theta(H_{l_n}(\theta)) = -I(\theta)$ is the Fisher information. (l_n is the log-likelihood)
The **Fisher-scoring** is then using the Fisher information matrix as a stand-in for the Hessian in the Newton-Raphson algorithm, giving the update:

$$\theta^{(k+1)} = \theta^{(k)} + I(\theta^{(k)})^{-1}\nabla_{l_n}(\theta^{(k)})$$

This has the same convergence properties as Newton-Raphson, but I is often easier to compute than H_{l_n}.

Iteratively re-weighted least squares is an algorithm for fitting GLM obtained by Newton-Raphson / Fisher-scoring. The above iteration is turned into each step solving a least square problem.

According to the chain rule,

$$\begin{aligned}\frac{\partial l_n}{\partial \beta_j} &= \sum_{i=1}^{n} \frac{\partial l_i}{\partial \theta_i}\frac{\partial \theta_i}{\partial \beta_j} \\ &= \sum_i \frac{Y_i - \mu_i}{\phi}\ h'(X_i^T\beta)\ X_i^j \\ &= \sum_i (\tilde{Y}_i - \tilde{\mu}_i)\ W_i\ X_i^j\end{aligned}$$

Where $W_i = (\frac{h'(X_i^T\beta)}{g'(\mu_i)\phi})$, $\tilde{Y}_i = (g'(\mu_1)Y_1, \dots g'(\mu_n)Y_n)^T$, and $\tilde{\mu_i} = (g'(\mu_1)\mu_1, \dots g'(\mu_n)\mu_n)^T$.
Rewriting the above in matrix form, the gradient is

$$\nabla_{l_n}(\beta) = \frac{\partial l_n}{\partial \beta_j} = X^T\ W\ (\tilde{Y}_i - \tilde{\mu}_i)$$

Where W is a diagonal matrix with $W_1, \dots W_n$ on the diagonal.
From the above before replacing with W, take second derivative to β_k,

$$\frac{\partial^2 l_n}{\partial \beta_j \beta_k} = \sum_i \frac{Y_i - \mu_i}{\phi}\ h''(X_i^T\beta)\ X_i^j\ X_i^k - \frac{1}{\phi}\sum_i (\frac{\partial \mu_i}{\partial \beta_k})\ h'(X_i^T\beta)\ X_i^j$$

Note that

$$\frac{\partial \mu_i}{\partial \beta_k} = \frac{\partial b'(h(X_i^T\beta))}{\partial \beta_k} = b''(\theta_i)h'(X_i^T\beta)X_i^k$$

And we take the expectation of $\frac{\partial^2 l_n}{\partial \beta_j \beta_k}$, we have

$$\mathbb{E}[\frac{\partial^2 l_n}{\partial \beta_j \beta_k} \mid X_1, \dots X_n] = -\frac{1}{\phi}\sum_i b''(\theta_i)\ [h'(X_i^T\beta)]^2 X_i^k X_i^j$$

And the matrix form

$$\mathbb{E}[H_{l_n}(\beta) \mid X_1, \dots X_n] = -\frac{1}{\phi}\sum_i b''(\theta_i)\ [h'(X_i^T\beta)]^2 X_i X_i^T$$

Recall that

$$b''(\theta_i)\ h'(X_i^T\beta) = \frac{1}{g'(\mu_i)}$$

As a result,

$$\mathbb{E}[H_{l_n}(\beta) \mid X_1, \dots X_n] = -\frac{h'(X_i^T\beta)}{g'(\mu_i)\phi} X_i X_i^T$$

Therefore $I(\beta) = -\mathbb{E}[H_{l_n}(\beta) \mid X_1, \dots X_n] = X^T W X$.
Plugging this back to the Fisher scoring method iteration,

$$\begin{aligned}\beta^{(k+1)} &= \beta^{(k)} + I(\beta^{(k)})^{-1}\nabla_{l_n}(\beta^{(k)}) \\ &= \beta^{(k)} + (X^T W^(k) X)^{-1} X^T W^{(k)}(\tilde{Y} - \tilde{\mu})\end{aligned}$$

Recall that with least squares, given $Y = X\beta + \epsilon$, we have $\hat{\beta} = (X^T X)^{-1} X^T Y$. If $\epsilon \sim N(0, I_d)$.
However if $\epsilon \sim N(0, W^{-1})$, where W^{-1} is a $n \times n$ diagonal matrix. When variances are different, the regression is said to be **heteroskedastic** (trust some areas less than you do others).
And **weighted least squares** comes in to solve this problem. We want

$$\arg\min_{\beta} (Y - X\beta)^T W (Y - X\beta)$$

And the solution is

$$(X^T W X)^{-1} X^T W Y$$

(we can get to this by plugging $\sqrt{W}$ into the terms before and after, since W is diagonal.)
Plugging this back to our problem, the response $Y = \tilde{Y} - \tilde{\mu} + X\beta^{(k)}$, design matrix X is X, and weight matrix $W = W(\beta^{(k)})$.
And we can obtain $\beta^{(k+1)}$ using any system solving weighted least square.

2.33 Markov chains

Consider a physical system with the following properties

- Can occupy any of a finite or countably infinite number of states $\epsilon_1, \epsilon_2, \ldots$
- Starting from some initial state at time $t = 0$, change its state randomly at the times $t = 1, 2, \ldots$. Let $\xi(t)$ denote the state of the system at time t, the evolution of the system can be described in steps $\xi(0) \to \xi(1) \ldots$
- At $t = 0$, the system occupies the state ϵ_i with initial probability $p_i^0 = P(\xi(0) = \epsilon_i), i = 1, 2, \ldots$
- Suppose at $t = n$ the system is at state ϵ_i, then the probability that the system goes into state ϵ_j at the next step is given by transition probability $p_{ij} = P(\xi(n+1) = \epsilon_j | \xi(n) = \epsilon_i)$, regardless of its behavior before time n.

A random process described by this model is a **Markov chain**.
p_{ij} can define a transition matrix P. Given $P(0) = I$ (unit matrix), $P(n) = P^n$.
A state is persistent if the probability of sooner or later returning to it is 1, or transient if the probability of sooner or later returning to it is less than 1.
If a Markov chain has a finite number of states, each accessible (probability of going from src to dst is positive) from every other state, then the states are all persistent.

A **continuous Markov process** has similar four conditions to the discrete one introduced before, just that transition probability is defined on continuous time t and don't depend on depend on s.

$$p_{ij}(t) = P(\xi(s+t) = \epsilon_j | \xi(s) = \epsilon_i) \quad i,\ j = 1, 2, \ldots$$

3 Statistical learning

Let $\mathbf{X}$ denote the input space, $\mathbf{Y}$ denote the output space, and $\mathbf{A}$ denote the action space.
A **decision function / prediction function** $f : \mathbf{X} \to \mathbf{A}$ maps an input to an action.
A **loss function** $l : \mathbf{A} \times \mathbf{Y} \to \mathbb{R}$ evaluates an action in the context of an output, and maps the error to a real. In traditional problems we can usually assume action has no effect on the output (like stock market prediction).

3.1 Risk, Bayesian function, empirical risk

Assume there is a data generating distribution $P_{\mathbf{X}\times\mathbf{Y}}$.
Risk of a decision function can then be defined as

$$R(f) = \mathbb{E}l(f(x), y)$$

Where an input/output pair (x, y) is generated independently and identically distributed from $P_{\mathbf{X}\times\mathbf{Y}}$.
A **Bayesian decision function / target function** $f^* : \mathbf{X} \to \mathbf{A}$ is a function that achieves the minimal risk among all possible functions.

$$f^* = \arg\min_f R(f)$$

Risk cannot be calculated as we are not given $P_{\mathbf{X}\times\mathbf{Y}}$.
Let $\mathbf{D}_n = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ be drawn iid from $P_{\mathbf{X}\times\mathbf{Y}}$.
The **empirical risk** of $f : \mathbf{X} \to \mathbf{A}$ with respect to $\mathbf{D}_n$ is

$$\hat{R}_n(f) = \frac{1}{n}\sum_{i=1}^{n} l(f(x_i), y_i)$$

By strong law of large numbers (2.2), when n is big enough, empirical risk can be used to approximate risk.

3.2 Empirical risk minimization, constrained empirical risk minimization

A function $\hat{f}$ is an **empirical risk minimizer** if

$$\hat{f} = \arg\min_f \hat{R}_n(f)$$

The prediction function that produces the smallest empiricall risk over set $\mathbf{D}_n$. This naturally leads to overfit.
So we minimize empirical risk, but only within a hypothesis space $\mathbf{F}$, being a set of prediction functions.

Constrained empirical risk minimizer can then be defined as

$$\hat{f}_n = \arg\min_{f \in \mathbf{F}} \frac{1}{n} \sum_{i=1}^{n} l(f(x_i), y_i)$$

If we plug Risk in instead, a **constrained risk minimizer** becomes

$$f^*_{\mathbf{F}} = \arg\min_{f \in \mathbf{F}} \mathbb{E} l(f(x), y)$$

Approximation error is the risk difference between the constrained risk minimizer (in $\mathbf{F}$) and the target

$$R(f_{\mathbf{F}}) - R(f^*)$$

Estimation error is the risk difference between the constrained empirical risk minimizer and the constrained risk minimizer (both in $\mathbf{F}$)

$$R(\hat{f}_n) - R(f_{\mathbf{F}})$$

Optimization error is the risk difference between a function $\tilde{f}_n$ (which we find in practice) and the constrained empirical risk minimizer *

$$R(\tilde{f}_n) - R(\hat{f}_n)$$

Excess risk is the risk between a function and the target

$$\text{Excess Risk}(\hat{f}_n) = \text{Estimation Error} + \text{Approximation Error} = R(\hat{f}_n) - R(f^*)$$

$$\begin{aligned}\text{Excess Risk}(\tilde{f}_n) &= \text{Optimization Error} + \text{Estimation Error} + \text{Approximation Error} \\ &= R(\tilde{f}_n) - R(f^*)\end{aligned}$$

3.3 Complexity measure, l_1, l_2 regularization, ridge regression, lasso regression, coordinate gradient descent

Regularization is a mechanism to reduce overfitting. By narrowing down the hypothesis space to only those functions who satisfies being within a complexity measure (or penalize the empirical risk minimizer with another term representing complexity).
Complexity measure $\Omega : \mathbf{F} \to [0, \infty)$ for linear decision functions $f(x) = w^T x$ can be defined using the p-norm (to the p) of w.

- l_0 complexity is then the number of non-zero coefficients
- l_1 **lasso** complexity is $\sum_{i=1}^{d} |w_i|$

*Optimization can be negative, however, this $\hat{R}(\tilde{f}_n) - \hat{R}(\hat{f}_n)$ can't. In this hypothesis space, this function can fit the overall world better than the constrained empirical risk minimizer, but it can't on the training set, on which the constrained ERM performs best in this space

- l_2 **ridge** complexity is $w^T w = \sum_{i=1}^{d} w_i^2$

To factor complexity measure in constrained empirical risk minimization, we can use **Ivanov** or **Tikhonov** regularization. where

$$\arg\min_{f \in \mathbf{F}} \frac{1}{n} \sum_{i=1}^{n} l(f(x_i), y_i) \text{ s.t. } \Omega(f) \leq r$$

is Ivanov regularization, and

$$\arg\min_{f \in \mathbf{F}} \frac{1}{n} \sum_{i=1}^{n} l(f(x_i), y_i) + \lambda \Omega(f)$$

is Tikhonov regularization. The two can be shown to be equivalent for many performance measure of f, and Ω, using Lagrangian duality theory.
Plugging in the lasso / ridge definition of Ω to either form, we get lasso / ridge regression.
For example, ridge regression in Tikhonov form looks like

$$\hat{w} = \arg\min_{w \in \mathbf{R}^d} \frac{1}{n} \sum_{i=1}^{n} (w^T x_i - y_i)^2 + \lambda \|w\|_2^2$$

Lasso regression typically lead to feature sparsity (more coefficient equal to 0). This can be shown intuitively by comparing the two on the intersection of contour plot (of empirical risk minimizer) and the area constrained by lasso (diamond on a $f(x) = ax + b$ prediction function) / ridge (circle on the same prediction function) regression, as show in Figure **??**[*].
Lasso regression (l_1-norm) is not differentiable, thus we cannot apply gradient descent as-is. We split each coefficient into positive and negative parts: rewrite the vector $w = w^+ - w^-$ ($|w| = w^+ + w^-$), and we have this equivalent problem[†]:

$$\arg\min_{w^+, w^-} \frac{1}{n} \sum_{i=1}^{n} ((w^+ - w^-)^T x_i - y_i)^2 + \lambda (w^+ + w^-)$$

subject to $w_i^+ \geq 0 \ \forall i$, $w_i^- \leq 0 \ \forall i$
Now we have a constraint and two vectors, to find the minimum, we can use **projected SGD**[‡] or **coordinate descent**.
Coordinate gradient descent on a vector w works by adjusting only a single w_i in each step, as opposed to possibly alter the entire w in one step as in regular gradient descent. We iteratively adjust each coordinate several times, where we can pick a random one to adjust, or do cyclic adjustment.

[*] https://niallmartin.wordpress.com/2016/05/12/shrinkage-methods-ridge-and-lasso-regression/

[†] whose equivalence can be proved. And this can be plugged in to a quadratic solver to give us w^+ and w^-

[‡] Normal gradient descent but project / reset each coordinate to the constrained space after each step

Coordinate gradient descent has a **closed form solution** for lasso (in the form below*), where we can use a formula to solve each w_i, instead of having to loop over them.

$$\hat{w}_j = \arg\min_{w_j \in \mathbf{R}} \frac{1}{n} \sum_{i=1}^{n} (w^T x_i - y_i)^2 + \lambda \|w\|_1$$

3.4 Elastic nets

Consider what happens in lasso and ridge regressions when features are linear: lasso would assign all the weight to the feature with larger scale, and ridge would assign the weights proportional to the features' scales.
It's similar when features are highly correlated (near-linear relationship between features): think Figure **??** instead of parallel lines, we get elongated ellipses, whose intersection with the hypothesis space would reflect the scales of correlated features.
Elastic net combines lasso and ridge penalties.

$$\hat{w} = \arg\min_{w \in \mathbf{R}^d} \frac{1}{n} \sum_{i=1}^{n} (w^T x_i - y_i)^2 + \lambda_1 \|w\|_1 + \lambda_2 \|w\|_2^2$$

Geometrically, we end with a hypothesis space like a diamond with edges bulging out (between a circle and a diamond).

3.5 Regression loss functions

A **distance-based loss function** is a loss function ($l(\hat{y}, y) \in \mathbf{R}$) that only depends on the **residual** $r = \hat{y} - y$. Most regression losses are distance-based.
Distance-based losses are translation-invariant: $l(\hat{y} + a, y + a) = l(\hat{y}, y)$.
Square loss ($l(r) = r^2$) penalizes outlier points more heavily than absolute (Laplacian) loss ($l = |r|$) does, and is considered less robust.
Downside with absolute loss is that it's not differentiable.
We are able to construct **Huber loss function** that is robust and differentiable: quadratic for $|r| \leq \delta$ and linear for $|r| > \delta$.

3.6 Classification loss functions

If we have action space $\mathbf{A}$ and outcome space $\mathbf{Y}$ both being $\{-1, 1\}$.
0-1 loss for $f : \mathbf{X} \to \{-1, 1\}$:

$$l(f(x), y) = 1 \ (f(x) \neq y)$$

*This form is not differentiable, but it can be shown that for coordinate gradient descent to work there is a weaker condition, which lasso satisfies. *This would indicate that the split into positive and negative shown above is not required, for solving using coordinate gradient descent?*

Where 1 $(f(x) \neq y)$ denotes score 1 whenever $(f(x) \neq y)$.

If we allow **real-valued prediction function** $f : \mathbf{X} \to \mathbf{R}$, meaning action space $\mathbf{A} = \mathbf{R}$ where the value represents confidence of our prediction.
We can define **margin** m as $m = y\hat{y}$, where $\hat{y}$ stands for the predicted score. We want to maximize the margin. Most classification losses are **margin-based**.
Empirical risk for 0-1 loss is

$$\hat{R}_n(f) = \frac{1}{n}\sum_{i=1}^{n} 1\ (y_i f(x_i) \leq 0)$$

$\hat{R}_n(f)$ is non-convex, not differentiable, discontinuous, and its optimization is NP-hard.

SVM/Hinge loss is defined as

$$l_{Hinge} = max\{1 - m, 0\} = (1 - m)_+$$

It is a convex, upper bound on 0-1 loss, and not differentiable at $m = 1$. And we have **margin error** when $m < 1$.
Using hinge loss function, we define **soft margin linear support vector machine** (a constrained empirical risk minimizer) as

$$\arg\min_{w \in \mathbf{R}^d} \frac{1}{n}\sum_{i=1}^{n} (1 - y_i f_w(x_i))_+ + \lambda \|w\|_2^2$$

With $l2$ regularization, and hypothesis space is linear $\{f(x) = w^T x | w \in \mathbf{R}^d\}$.

Logistic/log loss is defined as

$$l_{Logistic} = \log(1 + e^{-m})$$

It always wants more margin, and is differentiable.

0-1 loss, hinge and Logistic loss functions are illustrated in Figure **??** *.

3.7 SVM and Lagrangian duality

Adding a **bias** term b to the constrained empirical risk minimizer of **support vector machine**, we have

$$\arg\min_{w \in \mathbf{R}^d,\ b \in \mathbf{R}} \frac{c}{n}\sum_{i=1}^{n} \max(0, 1 - y_i(w^T x_i + b)) + \frac{1}{2}\|w\|_2^2$$

whose solution gives us the SVM prediction function. c is a constant that can be tweaked to decide how much regularization matters.

*http://fa.bianp.net/blog/2013/loss-functions-for-ordinal-regression/

This problem is not differentiable due to max. We can turn it into an equivalent problem.

$$\begin{aligned} \text{minimize} \quad & \frac{c}{n}\sum_{i=1}^{n}\xi_i + \frac{1}{2}\|w\|_2^2 & \\ \text{subject to} \quad & -\xi_i & \leq 0,\ \forall i \in \{1,\dots n\} \\ & (1 - y_i(w^T x_i + b)) - \xi_i & \leq 0,\ \forall i \in \{1,\dots n\} \end{aligned}$$

This has a differentiable objective function, $n+d+1$ unknowns and $2d$ affine constraints. This can be solved by off-the-shelf QP solver. $\sum_{i=1}^{n}\xi_i$ now represents the margin loss.
We apply **Lagrangian multiplier** to the constrained optimization problem to get the following

$$\begin{aligned} L(w,b,\xi,\alpha,\lambda) &= \frac{1}{2}\|w\|_2^2 + \frac{c}{n}\sum_{i=1}^{n}\xi_i + \sum_{i=1}^{n}\alpha_i(1 - y_i(w^T x_i + b) - \xi_i) - \sum_{i=1}^{n}\lambda_i\xi_i \\ &= \frac{1}{2}w^T w + \sum_{i=1}^{n}\xi_i(\frac{c}{n} - \alpha_i - \lambda_i) + \sum_{i=1}^{n}\alpha_i(1 - y_i(w^T x_i + b)) \end{aligned}$$

Its **primal** and **dual problems** look like

$$\begin{aligned} p^* &= \inf_{w,\xi.b}\sup_{\alpha,\lambda\geq 0} L(w,b,\xi,\alpha,\lambda) \\ &\geq \sup_{\alpha,\lambda\geq 0}\inf_{w,\xi.b} L(w,b,\xi,\alpha,\lambda) \\ &= d^* \end{aligned}$$

This satisfies strong duality by **Slater's constraint qualification**. *
The **Lagrangian dual function** then becomes

$$\begin{aligned} g(\alpha,\lambda) &= \inf_{w,\xi.b} L(w,b,\xi,\alpha,\lambda) \\ &= \inf_{w,\xi.b}(\frac{1}{2}w^T w + \sum_{i=1}^{n}\xi_i(\frac{c}{n} - \alpha_i - \lambda_i) + \sum_{i=1}^{n}\alpha_i(1 - y_i(w^T x_i + b))) \end{aligned}$$

This is convex and differentiable: quadratic in w, and linear in ξ_i and b. The global minima is at $\frac{\partial L}{\partial w} = 0$, $\frac{\partial L}{\partial \xi} = 0$, $\frac{\partial L}{\partial b} = 0$.
Solving the three partial derivations (**first order conditions**), we have

*Where a convex problem + affine constraints $\implies$ strong duality iff problem is **feasible**. This problem is feasible if we simply let $w = b = 0$ and $\xi_i = 1\ \forall i \in \{1,\dots n\}$

$$\frac{\partial L}{\partial w} = 0 \implies w = \sum_{i=1}^{n} \alpha_i y_i x_i$$
$$\frac{\partial L}{\partial \xi} = 0 \implies \alpha_i + \lambda_i = \frac{c}{n}$$
$$\frac{\partial L}{\partial b} = 0 \implies \sum_{i=1}^{n} \alpha_i y_i = 0$$

Replacing the w, ξ and bs from the **dual function**, the **dual problem** then becomes

$$\begin{aligned} \sup_{\alpha} \quad & \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \alpha_i \alpha_j y_i y_j x_j^T x_i \\ s.t. & \sum_{i=1}^{n} \alpha_i y_i = 0 \\ & \alpha_i \in \left[0, \tfrac{c}{n}\right] \quad \forall i \in \{1, \dots n\} \end{aligned}$$

Note

- Given a solution α^* to dual, primal solution is $w^* = \sum_{i=1}^{n} \alpha_i^* y_i x_i$.
- w^* is a linear combination of the data $x_1, \dots x_n$. The x_i's corresponding to $\alpha_i^* > 0$ are called **support vectors**.
- $\alpha^* \in \left[0, \frac{c}{n}\right]$, so c controls the maximum weight on each sample. This is **robust**.
- This is a quadratic objective with n unknowns and $n + 1$ constraints.
- Efficient minimization algorithm, sequential minimal optimization, exists.

Since we have **strong duality**, **complementary slackness** holds, where the Lagrangian multiplier $\times$ the constraint function is 0.

$$\alpha_i^*(1 - y_i f^*(x_i) - \xi_i) = 0$$
$$\lambda_i^* \xi_i^* = (\frac{c}{n} - \alpha_i^*)\xi_i^* = 0$$

This means

- $y_i f^*(x_i) > 1 \implies \alpha_i^* = 0,\ \xi_i^* = 0$
- $y_i f^*(x_i) < 1 \implies \alpha_i^* = \frac{c}{n},\ \xi_i^* > 0$
- $y_i f^*(x_i) = 1 \implies \alpha_i^* \in \left[0,\ \frac{c}{n}\right]$

- $\alpha_i^* = 0 \implies \xi_i^* = 0$ (margin loss is 0), so $y_i f^*(x_i) \geq 1$
- $\alpha_i^* = \frac{c}{n} \implies y_i f^*(x_i) \leq 1$
- $\alpha_i^* \in \left(0, \frac{c}{n}\right) \implies y_i f^*(x_i) = 1$

Suppose we have an i s.t. $\alpha_i^* \in \left(0, \frac{c}{n}\right)$, using the complementary slackness conditions we can solve the bias term b^*, whose value stays the same for any choice of i satisfying $\alpha_i^* \in \left(0, \frac{c}{n}\right)$. *

$$b^* = y_i - x_i^T w^*$$

If there are no such α_i^*s, then we have a degenerate SVM training problem.

3.8 Level sets / contours, sublevel sets and convex optimization

Let $f : \mathbf{R}^d \to \mathbf{R}$ be a function. A **level set** or **contour line** for the value c is the set of points $x \in \mathbf{R}^d$ for which $f(x) = c$.
A **sublevel set** for the value c is the set of points $x \in \mathbf{R}^d$ for which $f(x) \leq c$.
If $f : \mathbf{R}^d \to \mathbf{R}$ is **convex**, then sublevel sets are convex. Level sets and superlevel sets of convex functions are not generally convex, hence, the **standard form of convex optimization problem** uses ≤ 0 to express constraints.
The **implicit form of convex optimization problem** goes as

$$\begin{aligned} &\text{minimize} && f(x) \\ &\text{subject to} && x \in C \end{aligned}$$

where f is a convex function and C is a convex set.
Also, intersection of convex sets is convex.

3.9 First-order approximation

Suppose $f : \mathbf{R}^d \to \mathbf{R}$ is differentiable.
We can use **linear / first-order approximation** to predict $f(y)$ given $f(x)$ and $\bigtriangledown f(x)$

$$f(y) \approx f(x) + \bigtriangledown f(x)^T (y - x)$$

Now suppose $f : \mathbf{R}^d \to \mathbf{R}$ is convex and differentiable.
The linear approximation to f at x is a **global underestimator** of f.

$$\forall\ x, y \in \mathbf{R}^d, \quad f(y) \geq f(x) + \bigtriangledown f(x)^T (y - x)$$

*With numerical error, it's more rubust to average over all eligible i and use the mean of resulting b^*

And if $\bigtriangledown f(x) = 0$ then x is a global minimizer of f *.

3.10 Subgradients

A vector $g \in \mathbf{R}^d$ is a **subgradient** of $f : \mathbf{R}^d \to \mathbf{R}$ at x if $\forall\ z$,

$$f(z) \geq f(x) + g^T(z - x)$$

f is **subdifferentiable** at x if $\exists$ at least one subgradient at x.
The set of all subgradients at x is called the **subdifferential**: $\partial f(x)$
f is convex and differentiable $\implies \partial f(x) = \{\bigtriangledown f(x)\}$. At any point x, there can be 0, 1, or infinite many subgradients. $\partial f(x) = \emptyset \implies f$ is not convex.
If $0 \in \partial f(x)$, then x is a **global minimizer** of f.
As a reminder, for function $f : \mathbf{R}^d \to R$,

- graph of function lives in $\mathbf{R}^{d+1}$
- gradient, subgradient of f live in $\mathbf{R}^d$, and
- contours, level sets, sublevel sets are in $\mathbf{R}^d$

Gradient at x is orthogonal to level set at x. (assuming f continuously differentiable.)

Now to figure out the **direction** on which to do a subgradient descent.
Let $f : \mathbf{R}^d \to \mathbf{R}$ have a subgradient g at x_0, hyperplane H orthogonal to g at x_0 must **support** the level set $S = \{x \in \mathbf{R}^d | f(x) = f(x_0)\}$, meaning H contains x_0 and all of S lies on one side of H. The proof of which† suggests the following: Points on subgradient g side of H have larger f-values than $f(x_0)$, and points on $-g$ side may not have smaller f-values, meaning $-g$ may not be a descent direction.

In **subgradient descent**, suppose we repeatedly step in a negative subgradient direction $x = x_0 - tg$ where $t > 0$ is the step size and $g \in \partial f(x_0)$.
We can prove $-g$ gets us closer to minimizer. Meaning, suppose f is convex, let $x = x_0 - tg$ for $g \in \partial f(x_0)$, and let z be any point for which $f(z) < f(x_0)$, then for small enough $t > 0$, $\|x - z\|_2 < \|x_0 - z\|_2$.
Proof:

$$\begin{aligned}
\|x - z\|_2^2 &= \|x_0 - tg - z\|_2^2 \\
&= \|x_0 - z\|_2^2 - 2tg^T(x_0 - z) + t^2\|g\|_2^2 \\
&\leq \|x_0 - z\|_2^2 - 2t(f(x_0) - f(z)) + t^2\|g\|_2^2
\end{aligned}$$

Consider

*Where local information gives global information!

†Using the definition of subgradient, and inner product $g^T \cdot (y - x_0) > 0$ if y is on the side g points in

$$g(t) = -2t(f(x_0) - f(z)) + t^2\|g\|_2^2$$

It's a convex quadratic facing upwards, has zeros at $t = 0$ and $t = 2(f(x_0) - f(z))/\|g\|_2^2 > 0$, so

$$g(t) < 0 \ \forall \ t \in \left(0, \frac{2(f(x_0)-f(x))}{\|g\|_2^2}\right)$$

Thus we have $-g$ gets us closer to a smaller $f(z)$.

3.11 Features extraction

Mapping an input space $\mathbf{X}$ (sound wave, image, DNA sequence) to a vector $\mathbf{R}^d$ is called **feature extraction** or **featurization**.
A **feature template** is a group of features all computed in a similar way (e.g. $last\_three\_chars\_of(x) = \_\_\_$).*
A **one-hot encoding** is a feature template that always has exactly one non-zero value.
Features can be encoded with an array (good for dense features), or a map (good for sparse features).
Some factors to consider when featurizing

- Non-monotonicity. E.g. temperature as a feature for health prediction, health is not monotonic with temperature. We can transform the input $\Phi x = [1, \{temperature(x) - 37\}^2]$†, but this would require domain knowledge, instead, we could do $\Phi(x) = [1, temperature(x), \{temperature(x)\}^2]$, where having one extra feature, we increase the flexibility and make it easier to use.‡

- Saturation. E.g. find products relevant to user's query, where given a product x, we have a feature map $\Phi(x) = [1, N(x)]$ where $N(x) =$ number of people who bought x. However at some point x should saturate: a product bought 50000 times is not necessarily 10 times more relevant than a product bought 5000 times, as a linear model would suggest. We could do $log(1+N(x))$, or $arctan(x)$§ to slow down the growth. Or we could do a discretization, like $\Phi(x) = [1(x < 10), 1(10 \leq x < 100), 1(100 \leq x)]$¶.

- Interaction. E.g. predicting health with weight and height, it's not that they individually matter, but rather weight relative to height. You can add a cross term $h(x)w(x)$, making $\Phi(x) = [1, h(x), w(x), h^2(x), x^2(x), h(x)w(x)]$

*With regularization, our resulting prediction function won't be too complicated.
†Array representation of features, where 1 is a bias term
‡Think less, make the computer do more
§A sigmoid function. Note how this transformation does not need to be convex.
¶Where if not bucketed carefully, buckets with few items will have coefficients 0, due to regularization. Instead, we could have $\Phi(x) = [1(x \geq 5), 1(x \geq 10), 1(x \geq 100)], whereas small bucket would fall back to the previous feature.$

A **predicate** of the input space is a function $P : \mathbf{X} \to \{\text{True, False}\}$.
What about given features $\Phi(x) = [x_1, x_2]$, we want to classify if a point will fall into a circle in the two-dimensional input space? We could add $x_1^2 + x_2^2$ as another feature.
Similaryly, output space can be transformed as well.
We essentially grow the (linear) hypothesis space by adding more features.

3.12 Kernel methods

From a high level, goal is to allow access to huge feature spaces without suffering heavy computational cost.
With featurization, we can map input space $\mathbf{X} \subset \mathbf{R}^n$ to $\mathbf{R}^{d*}$, and the hypothesis space is of affine functions on feature space looks like

$$\mathbf{H} = \{x \to w^T \Phi x + b | w \in \mathbf{R}^d, b \in \mathbf{R}\}$$

To get expressive hypothesis spaces using linear models, we need high-dimensional feature spaces (say, adding in all $x_i^{k_i}$ where $\sum k_i \leq C$). This complicates computation, and may cause overfitting. Overfitting can be handled with regularization, and kernel methods can help with memory and computational costs.
For example, recall a featurized SVM prediction function, which is the solution to the following

$$\underset{w \in \mathbf{R}^d, b \in \mathbf{R}}{\arg\min} \; (\frac{1}{2} w^T w + \frac{c}{n} \sum_{i=1}^{n} (1 - y_i [w^T \Phi(x_i) + b])_+$$

whose dual is

$$\begin{aligned} \sup_{\alpha} \quad & \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \Phi(x_j)^T \Phi(x_i) \\ s.t. & \sum_{i=1}^{n} \alpha_i y_i = 0 \\ & \alpha_i \in \left[0, \tfrac{c}{n}\right] \quad \forall i \in \{1, \dots n\} \end{aligned}$$

Where Φx only shows up as inner products with other xs.
A method is **kernerlized** if inputs only appear inside inner products. $\langle \Phi(x), \Phi(x') \rangle$ for $x, x' \in \mathbf{X}$. The **kernel function** corresponding to Φ and inner product $\langle ., . \rangle$ is

$$k(x, x') = \langle \Phi(x), \Phi(x') \rangle$$

For example, consider quadratic feature map for $x = (x_1, \dots x_d) \in \mathbf{R}^d$

$$\Phi(x) = (x_1, \dots x_d, x_1^2, \dots x_d^2, \sqrt{2} x_1 x_2, \dots, \sqrt{2} x_{d-1} x_d)^T$$

*And as it would seem, this optimization makes sense when $n < d$

has dimension $O(d^2)$, but for any $x, x' \in \mathbf{R}^d$

$$k(x, x') = \langle \Phi(x), \Phi(x') \rangle = \langle x, x' \rangle + \langle x, x' \rangle^2$$

Bringing down the computation cost from $O(d^2)$ to $O(d)$.*
Continuing on with the SVM example, with linear kernel $k(x, x') = x^T x'$, we define the **kernel matrix** (**Gram matrix**) as such:

$$K = (\langle x_i, x_j \rangle)_{i,j} = \begin{pmatrix} \langle x_1, x_1 \rangle \dots \langle x_1, x_n \rangle \\ \dots \\ \langle x_n, x_1 \rangle \dots \langle x_n, x_n \rangle \end{pmatrix}$$

Then for the standard Euclidean inner product $\langle x_i, x_j \rangle = x_i^T x_j$, we have

$$K = XX^T$$

Where $X = (x_1, \dots x_n)^T$, an $n \times d$ vector, and K is all the $Phi(x)-$terms we are interested in.†
And the dual problem becomes

$$\begin{aligned} \sup_{\alpha} \quad & \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j K_{ji} \\ s.t. & \sum_{i=1}^{n} \alpha_i y_i = 0 \\ & \alpha_i \in \left[0, \tfrac{c}{n}\right] \quad \forall i \in \{1, \dots n\} \end{aligned}$$

This also gives us the flexibility to change kernels to a different K of $n \times n$ dimension, which can, e.g., correspond to a high dimensional feature space. This is **kernel trick**.

And there are other non-linear kernels, our quadratic example earlier and **polynomial kernel** $k(x, x') = (1 + \langle x, x' \rangle)^M$, which corresponds to a feature map with all monomials up to degree M, the computational cost of the kernel does not grow with M, while plugging in the featurized terms directly the computational cost grows rapidly with M.
And **Radial Basis Function / Gaussian kernel**, $\forall x, x' \in \mathbf{R}^d$

$$k(w, x) = e^{-\frac{\|x - x'\|^2}{2\sigma^2}}$$

‡

*Often useful to think of the kernel function as a **similarity score**

†Recall with ridge regression, we worked with $X^T X$, which is $d \times d$, meaning we are interested in kernel methods when $n \leq d$

‡Note how this acts like a similarity score, and with this kernel, your featurization dimension d may grow, but to compute the kernel is always $n \times n$ where n is the number of data points.

3.13 Representer theorem and its usage in kernelization

Consider SVM objective function in a more general term,

$$\arg\min_{w \in \mathbf{H}} J(w)$$

And let $J(w)$ be

$$J(w) = R(\|w\|) + L(\langle w, \Phi(x_1)\rangle, \ldots, \langle w, \Phi(x_n)\rangle)$$

Where $\|.\|$ corresponds to the inner product on space $\mathbf{H}$. $R : [0, \infty) \to \mathbf{R}$ is the regularization term and $L : \mathbf{R}^n \to \mathbf{R}$ is the loss term.*
If $J(w)$ has a minimizer, then it has a minimizer of the form $w^* = \sum_{i=1}^{n} \alpha_i \Phi(x_i)$.†
With this, we don't have to search over the feature space w^* is in, $\mathbf{R}^d$, but rather having computed the n $\Phi(x_i)$s, we search over that space of n dimensions. This lets you deal with infinite dimensional feature space.
As an example, a kernelized objective function for SVM looks like

$$\arg\min_{\alpha \in \mathbf{R}^n} R(\sqrt{\alpha^T K \alpha}) + L(K\alpha)$$

Where there is no direct access to $\Phi(x_i)$, and all references are via kernel matrix K.

3.14 Performance evaluation

Intuition,

- Always spend some time building a linear baseline model. (lasso / ridge / elastic net regression; l1 / l2 regularized logistic regression or svm)
- Prefer simpler models if performance is the same.
- Consider building an oracle model, which is helpful to get an upper bound on achievable performance. E.g. fit your validation data without regularization. This can give estimate of the approximation error of our hypothesis space.

Confusion matrix for binary classification problem, the 4-cell matrix of true / false positive / negatives.

- **True positive**: predicted positive, and it's right
- **True negative**: predicted negative, and it's right
- **False positive / type 1 error**: predicted positive, and it's wrong

*Each inner product represents a (linear) prediction. Ridge regression and SVM are both of this form. Lasso is not, as l_1 norm does not correspond to an inner product.

†This is the same conclusion we reached at the end of SVM: w^* is a linear combination of features. This can be proved using Pythagorean theorem and projection theorem: do a projection from w^* to M, span of the data

- **False negative / type 2 error**: predicted negative, and it's wrong

Criteria based on the confusion matrix,

- **Accuracy** $(TP + TN)/(TP + TN + FP + FN)$
- **Error rate** $(FP + FN)/(TP + TN + FP + FN)$
- Accuracy + Error rate = 1.
- Accuracy by itself is often not enough, consider a no-information classifier (always produced 0, e.g.) on a distribution that's predominantly one-sided (e.g. 0).
- **Precision** $TP/(TP + FP)$: high precision means low false alarm rate.
- **True positive rate / recall / sensitivity** $TP/(TP + FN)$: high recall means you are not missing many positives.
- **False negative rate / miss rate** $FN/(FN + TP)$
- **False positive rate / fall-out / false-alarm rate** $FP/(FP + TN)$
- **True negative rate / specificity** $TN/(FP + TN)$
- F_1 **score**. $F_1 = 2\frac{precision \cdot recall}{precision + recall}$. F_β. Weigh towards precision or recall

In a classification problem, the real-valued prediction function (score function) does not have to be thresholded at 0*. Thresholding the score function differently causes the performance metrics to shift. We can threshold it at a value to meet our performance goals, e.g. $recall > 80\%$.
Measurement curves based on these rates.

- **Precision-recall curve**: as you change the threshold, how precision and recall change.
- **Receiver-operation characteristic (ROC) curve**: as you change threshold, how TP rate and FP rate change.
- **AUC / AUC ROC**: area under the ROC curve.

3.15 Probablistic modeling

Instead of producing a single value or classification (regression or classification), produce a probability distribution of values, with which we can decide how likely a test data point is. This is applicable in e.g. anonamaly detection.
The approach usually entails selecting a particular probability distribution, and fit the samples to derive the parameters of that distribution. (E.g. the mean value of Poisson distribution)

*> 0 predict positive, < 0 predict negative

Stratification means partitioning our input data into groups, and treat each group separately. E.g. we stratify each symbol traded into time ranges like 9:30A to 10:30A on every weekday, and train a model for that particular time range.
Bucketing / binning is the opposite of stratification, where we combine natural groups of data into a single group, and train a model for that single group. E.g. we can bucket multiple tech stock symbols into one group, and train one model to predict their overall price movement.
Stratification brings out specificity (also more bias), bucketing smoothes bias out but lose specificity.

3.16 Neural nets

Linear prediction functions, like SVM, ridge, lasso generate feature vector $\Phi(x)$ by hand, and learn parameter vector w from data. A neural net adds **hidden nodes** between the $\Phi(x)$ and score.
A single **perceptron** is of the form

$$y = g(x_0 + v_i^T \Phi(x))$$

whereby we pass the linear combination of inputs $\Phi(x)$ through a **non-linear activation function** g to get the result y.
A perceptron can be multi-output. A layer where all inputs ($\Phi(x)$) connect to all outputs (linear combination of inputs) is called a **dense** layer.
One hidden layer can consist of some number of perceptrons, and then the end result is the linear combination of the results coming from each of the perceptrons in this layer. E.g. in this one hidden layer representation:

$$y_i = g(v_i^T \Phi(x))$$
$$score = w^T h + b$$

We need to learn the vectors v_i (with a bias term), w, and scalar b.
We can compose perceptrons in multiple layers, forming a deep neural network.
The rationale of an activation function is to introduce non-linearity into the network (and deal with non-linear data). This non-linearity allows approximating arbitrary functions.
Some common activation functions include:
Sigmoid

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

Mapping $R \to (0, 1)$, often suitable for probability problems.
Hyperbolic tangent

$$\sigma(x) = tanh(x) = \frac{e^x + e^{-x}}{e^x + e^{-x}}$$

Rectified linear function (ReLU). Desirable for its simplicity, and often works well in practice.

$$\sigma(x) = max(0, x)$$

One way to think about neural nets is that it's learning the non-linear featurization functions we want to create.
Our empirical loss is then a simple function of the network weights. This function is not necessarily convex.
To learn the weights, we can use gradient descent: The loss function is a function of all the weights, and then starting from a random point, compute the gradient, and take a small step (learning rate) along negative gradient where loss is minimized. Continue until we converge (to a local minimum).
How do we compute the gradient: **backpropagation**. Imagine we have a simple perceptron on a single feature where:

$$\hat{y} = g(x_1 * w_1) * w_2$$
$$loss = some_loss_function(y - \hat{y})$$

and $loss$ is a function of (w_1, w_2).
To know $\frac{\partial loss}{\partial w_2}$, we can use the chain rule:

$$\frac{\partial loss(w_1, w_2)}{\partial w_2} = \frac{\partial loss(w_1, w_2)}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial w_2}$$

And to know $\frac{\partial loss}{\partial w_1}$, we need to apply the chain rule one more time.
The loss surface can have many local minima in practice.
Setting learning rate in practice: trial and error, or dynamically adapted.

3.16.1 In practice

Computing loss function's gradient to all weights (with all data points factored in) is not practical. We use **Mini-batching** instead (i.e. randomize a set of data points first, and compute loss function of that set's gradient), a balance between the expensive full descent and noisy sgd.
Mini-batching is parallelizable (by the partial derivatives to each weight, I guess?).
Overfit and regularization:
One regularization commonly seen in nnets is **dropout**: during training, randomly set some (predefined percent) of the activations to 0. (you can't overly rely on some pathways too strongly)
Another is **early stop**: stop iterating when test set loss does not go down.
Universal approximation theorem: a neural network with one (possibly huge) hidden layer can uniformly approximate any continuous function on a compact set iff the activation function is not a polynomial. (the bias term is necessary)

3.17 Deep sequence modeling

The above section describes a feed forward nnet, where we have fixed static input to fixed static output, i.e. $x \in R^m \rightarrow \hat{y} \in R^n$.
Many practical problems involve sequential processing / modeling (text, audio wave, stock prices, genetic data).
Some use cases include:

- tweet sentiment classification (one sequence to a static classification)
- label an image (one static input to a sequence)
- machine translation, music generation (one sequence to another)

The problem is then about how to add the temporal dimension to our models? If we are to stitch a sequence of feed forward nnets (at t_i, a feed forward nnet of $x_i \rightarrow \hat{y}_i$) together, some states from t_0 needs to be factored in for the feed forward nnet at t_1, i.e. neurons with recurrence.
We then introduce a memory between the feed forward nnets at each time step:

$$\hat{y}_t = f(x_t, h_{t-1})$$

meaning our current network at time t depends on the past memory at the previous timestamp.
The key idea for **recurrent neural network** is then a memory $h(t)$ updated at each time step a sequence is processed, where the cell state is

$$h(t) = f_W(x_t, h_{t-1})$$

The same f_W is used at every time step of processing the sequence.
Then, given an input vector x_t, we update the hidden state, e.g.

$$h(t) = tanh(W_{hh}^T h_{t-1} + W_{xh}^T x_t)$$

then

$$\hat{y}(t) = W_{hy}^T h(t)$$

At every time step, we are reusing the same weight matrices W_T.
Taking a step back, to model sequences, we need to be able to

- handle variable-length sequences
- track long-term dependencies
- maintain information about order
- share parameters across the sequence

RNN meets all of these sequence modeling criteria. While e.g. feed forward networks have inputs of fixed dimensionality.

3.17.1 Predicting next word

Now imagine we are predicting the next word in a sentence: how do we represent language in an RNN?
Start with a corpus of words, then each unique word is represented by an index in this words array.
Then we use some **embedding** to transform indexes into a vector of fixed size, e.g. one-hot embedding, or learned embedding (like a PCA on words and decomposing each word into a lower dimension vector).

3.17.2 Training RNN: backpropagation through time (BPTT)

The overall loss is the sum of loss at each time step, and computing the gradient of the overall loss wrt h_0 is going to involve many factors of W_{hh} and repeated gradient calculations. This introduces two problems: exploding gradients and vanishing gradients (multiply too many small numbers together).
(how parallelizable is this?)
Gradient clipping is often used to address the former, and to address the latter, choosing activation function (ReLU's derivative is not continuous, and for all values > 0, its derivative is 1, making it ideal to tackle vanishing gradients), weight initialization (initialize weights to identity matrixes and biases to 0), and network architecture can help.
Network architceture is most robust among all 3: the idea is to use a more complex recurrent unit with gates (**Gated Cells**) to control what information is passed through.
Long Short Term Memory (LSTM) is a gated cell approach. Networks built using LSTM are particularly good at maintaining long term dependency throughout the data / tracking information across multiple time steps.
An LSTM node looks something like
To summarize, LSTMs

- maintain a separate cell state c_t from what is outputtted,
- use gates to control the flow of information
- backpropagation through time with uninterrupted gradient flow.

In particular, the gates

- **forget**: gets rid of irrelevant current information,
- **store**: relevant information from this time step,
- selectively **update** cell state,
- **output** gate returns a filtered version of the cell state),

Attention module. Transformers.

3.18 Convolutional neural network

Computer vision (what is where, what actions are taken, to predict and anticipate, given a 2D list of numbers).
Problems include regression and classification (which president is this, what's present in the image), traditionally doable via manual feature extraction based on domain knowledge, but facing challenges like viewpoint, illumination, scale variation of the same object.
The deep learning approach tries to learn a hierarchy of features directly from the data instead of hand engineering. This requires a different architecture from the previous sections' feed forward dense nnets or RNNs.
We can have a fully connected nnet where we connect neuron in hidden layer to all neurons in input layer (1D vector), but this means the spatial information from the input is lost, and introduces many parameters.
How can we build spatial information (which node is close to which) back into the nnet? We can instead connect patches of input to neurons in the hidden layer, where a patch maps a 2D area to one single input, and can be generated by applying a sliding window on the original image.
The idea of **convolution** is to preserve the spatial relationship between pixels by learning features in small little patches.
We do an element-wise multiplication between the filter matrix (with learnt weights) and the patch of input image, and add up the result. * The resulting 2D matrix is a feature map that tells where this filter was activated in the input image.
By applying different (constructed) filter matrices, we could learn to sharpen the image, detect edges, etc. Historically effort had gone into hand engineering these filters. CNN allows learning the weights defining these filters.
To summarize what CNN does:

- apply a set of weights (filter matrix) to extract local features
- use multiple filters to extract different features
- spatially share the parameters of each filter

Pass feature maps from convolution layer through non-linearity (often ReLU to replace all negative values) and pooling (downsampling operation on each feature map), and pass the result through a fully connected hidden layer.
The feature map from convolution layer can be represented with a 3D structure where each plane is the application of one filter, and we have depth filters.
Pooling is used to reduce dimensionlity while keep spatial invariance, e.g. taking the max (or mean) of each sliding window.
We can apply many layers of convolution and pooling (convolution on the feature map from previous pooling), and the resulting feature map represent high-level features of the input.

*This is what a convolution means: element-wise multiplication and sum up the results.

The fully connected layer uses those features to classify an image (or whatever other tasks, note that e.g. an object detection task can share the same feature map as a classification task. softmax activation is often used for classification), and express an output as a probability of image belonging to a particular class.

3.18.1 Some applications

Object detection (output bounding boxes): one naive way is to draw random bounding boxes and pass them through CNN classification. This is slow, instead we could use R-CNN: find regions we think have objects (brittle!), use CNN to classify.
Faster R-CNN tries to learn the region proposals through another nnet.
Semantic segmentation (output selection boxes like magic wand): fully convolutional network that classifies every single pixel using upsampling.
Captioning may feed feature map from convolution into another RNN.
How are CNN trained?

3.19 Deep generative models

Unsupervised learning.
Generative modeling: take as input training samples from some distribution and learn a model that represents that distribution.
Key problem: how can we learn $P_{model}(x)$ similar to $P_{data}(x)$.
Can we learn the true explanatory factors (e.g. latent variables) from only observed data?

3.19.1 Autoencoders and variational autoencoders

Autoencoders: unsupervised approach for learning a lower-dimensional feature representation from unlabeled training data.
Encoders learn mapping from the data x to a lower dimensional latent space z (through a series of nnets). But how?
Train the model to use features z to reconstruct the original data, and we try to minimize the loss between original data and reconstructed data.
Decoders learn mapping back from the latent space z to a reconstructed observation $\hat{x}$ (through a series of nnets).
Autoencoding is a form of compression and the lower dimensionality of latent space the poorer your reconstruction quality.
Bottleneck hidden layer forces network to learn a compressed latent representation. **Reconstruction loss** forces the latent representation to capture (encode) as much information about the data as possible.
Smaller latent space will force a larger training bottleneck.
A traditional encoder learns a deterministic latent space z (the same if given the same data). **Variational autoencoders (VAE)** imposes a stochastic twist on this process: instead of learning the latent variable z directly, it learns a mean μ and variance σ associated with that latent variable (or its probability

distribution). We sample the μ and σ to get latent samples and to reconstruct data.
The encoder computes $q_\phi(z|x)$ (distribution of z given x under the encoder parameters ϕ, inferred latent distribution) and decoder computes $p_\theta(x|z)$.
The loss of VAE is

$$L(\phi, \theta, x) = \text{reconstruction loss} + \text{regularization term}$$

The regularization term places a fixed prior on the distribution of z (denoted $p(z)$), i.e. some initial hypotheses about what we expect z to look like (e.g. standard Gaussian). It's a function of the divergence between $q_\phi(z|x)$ and $p(z)$ (e.g. KL-divergence between the two distributions).
Regularization enforces

- continuity: points that are close in latent space results in similar content after decoding
- completeness: sampling from latent space should result in meaningful content after decoding

There is a problem with VAE training: we cannot backpropagate gradients through sampling layers. Instead, we reparametrize the sampling layer: consider the sampled latent vector $z = \mu + \sigma \bigodot \epsilon$ where $\epsilon \tilde{N}(0, 1)$ (a fixed μ vector plus a fixed σ vector scaled by random constants drawn from the prior distribution).
Latent perturbation: keep all other latent variables the same and tweak one.
Ideally we want latent variables that are uncorrelated with each other (disentanglement).
β-VAE applies a multiplier on the regularization term to encourage disentanglement, i.e.

$$L(\phi, \theta; x, z, \beta) = \mathbb{E}_{q_\phi(z|x)}[logp_\theta(x|z)] - \beta D_{KL}(q_\phi(z|x)||p(z))$$

One application: debiasing, use latent distribution to create fair and representative datasets.

3.19.2 Generative adversarial networks (GANs)

What if we don't model the distribution of the underlying latent variables, instead just sample to generate new instances.
Problem: we want to sample from a very complex distribution. Instead we sample from something simple (noise), learn a transformation to the data distribution, and then sample from this distribution to generate data as close to our input as possible.
The key breakthrough of GAN is to have two nnets (a generator and a discriminator) compete against each other. The generator (G) turns noise into an imitation of the data to try to trick the discriminator (D). The discriminator tries to identify real data from fakes generated by the generator.

Training of GAN defines adversarial objectives for the generator and discriminator. For the discriminator:

$$\arg\max_D \mathbb{E}_{z,x}[logD(G(z)) + log(1 - D(x))]$$

The first term is to maximize the discriminator's probability to identify fake data as fake, and the second is to maximize the discriminator's probability to identify real data as real.
For the generator:

$$\arg\min_G \mathbb{E}_{z,x}[logD(G(z)) + log(1 - D(x))]$$

We minimize the discriminator's ability to successfully tell fake data from real. And combining the two, we want to find a generator that's best at fooling the discriminator.

$$\arg\min_G \arg\max_D \mathbb{E}_{z,x}[logD(G(z)) + log(1 - D(x))]$$

Progressive growth of GANs: we can iteratively build more details into the generator.
StyleGAN: progressive growth + style transfer.
Conditional GAN, application in paired translation.
Cycle GAN, learns transformation across domains without pairs of data, e.g. synthesizing speech.

3.20 Deep reinforcement learning

Given state-action pairs, maximize future rewards over many time steps.
Concepts

- Agent: takes actions.
- Environment: the world in which the agent exists and operates.
- Action: a move agent can make in the environment. An action in action space $a_t \in A$, can be discrete or continuous.
- Observation: of the environment after taking actions.
- State: a situation which the agent perceives.
- Reward: measures the success / failure of the agent's action. Total reward since time step t into the future: $R_t = \Sigma_{i=t}^{\infty} r_i$, often future rewards needs to be discounted $R_t = \Sigma_{i=t}^{\infty} \gamma^i r_i$

The **Q-function** captures the expected total future reward an agent in state s can receive by executing an action a.

$$Q(s_t, a_t) = \mathbb{E}[R_t | s_t, a_t]$$

Ultimately, the agent needs a policy $\pi(s)$ to infer the best action to take at its state s, i.e.

$$\pi^*(s) = \arg\max_a Q(s, a)$$

Value learning: Q-functions are unintuitive, and this tries to use a nnet to model Q-functions / predict expected return.
Action + State – Deep Q Network –¿ $Q(s, a)$
Or more efficiently,
State – Deep Q Network –¿ a vector of $Q(s, a_1), Q(s, a_2), ...$
To train the DQN, we use Q-loss (minimize the difference between the target return of taking all the best actions and predicted return)

$$\mathbf{L} = \mathbb{E}[\left\| (r + \gamma \arg\max_{a'} Q(s', a')) - Q(s, a) \right\|^2]$$

Then use the learned Q-function to infer the optimal policy.
Some downsides include: This can model action spaces that are discrete and small and cannot handle continuous spaces. Policy is deterministically computed from Q function by maximizing the reward, and this cannot learn stochastic policy.

3.20.1 Policy learning

: policy gradient tries to directly learn the policy $\pi(s)$ to decide which action to take.
The nnet predicts a probability distribution, the probability of each action being the best action, i.e. (for all is, $p(a_i|s)$).
This can handle continuous action space.
Say we assume a Gaussian distribution, the nnet then needs to predict the μ and σ^2.
We sample this distribution to get an action.
To train policy gradient: we initialize the agent, run a policy till termination, record all state, action, rewards triplets, and decrease the probability of actions that resulted in a low reward, increase the probability of actions that resulted in a high reward.
How to do the probability increase/decrease?
Loss function can be expressed as the log-likelihood of action multiplied by reward: $-logp(a_t|s_t)R_t$. (action is good, we amplify its likelihood.)

3.21 Multinomial logistic regression

4 Exercises

4.1 Deriving gradient affine form

1. Given $f(w) = c^t w$, $\bigtriangledown f(w)$?

$$f'(x;u) = \lim_{h\to 0} \frac{f(w+hu) - f(w)}{h} = \lim_{h\to 0} \frac{c^T hu}{h} = c^T u$$

This shows

$$\bigtriangledown f(x) = c$$

2. Given $f(w) = w^T Aw$, $\bigtriangledown f(w)$?

$$\begin{aligned} f'(w;u) &= \lim_{h\to 0} \frac{f(w+hu) - f(w)}{h} \\ &= \lim_{h\to 0} \frac{(w+hu)^T A(w+hu) - w^T Aw}{h} \\ &= \lim_{h\to 0} \frac{w^T Aw + hw^T u + hu^T Aw + h^2 u^T Au - w^T Aw}{h} \\ &= u^T Aw + w^T Au \\ &= w^T A^T u + w^T Au \end{aligned}$$

This shows

$$\bigtriangledown f(x) = (w^T A^T + w^T A)^T = (A + A^T)w$$

3. Given $f(w) = \|Aw - y\|_2^2$, $\bigtriangledown f(w)$?

$$f(w) = \|Aw - y\|_2^2 = (Aw - y)^T (Aw - y)$$

$$\begin{aligned} f'(w;u) &= \lim_{h\to 0} \frac{(A(w+hu) - y)^T (A(w+hu) - y) - (Aw - y)^T (Aw - y)}{h} \\ &= \lim_{h\to 0} (A(w+hu) - y)^T Au + (Au)^T (A(w+hu) - y) \\ &= (Aw - y)^T Au + (Au)^T (Aw - y) \\ &= 2(Aw - y)^T Au \end{aligned}$$

This shows

$$\bigtriangledown f(x) = 2((Aw - y)^T A)^T = 2A^T (Aw - y) = 2A^T Aw - 2A^T y$$

4. Given $f(w) = \|Aw - y\|_2^2 + \lambda \|w\|_2^2$, express $f(w) = \|Bw - z\|_2^2$?

The gradient of the two need to be the same, using previous result, we need to solve

$$\begin{cases} A^TA + \lambda & = B^TB \\ A^Ty & = B^Tz \end{cases}$$

Note the equivalence between extending a matrix and addition, let

$$B = \begin{pmatrix} A \\ \sqrt{\lambda} I_{n\times n} \end{pmatrix} \text{ and } z = \begin{pmatrix} y \\ 0_{n\times 1} \end{pmatrix}$$

written in block-matrix form.

4.2 Recap: linear regression with square loss

Why OLS (vertical distance squared loss) estimator ($f(y)$):
OLS is a best unbiased linear estimator, when the true model is linear and $\mathbb{E}(\epsilon_i|X) = 0$, $Var(\epsilon_i|X) = \sigma^2$ and $Cov(\epsilon_i, \epsilon_j|X) = 0$.
Gaussian error $\implies$ OLS is best unbiased.
Best meaning lowest variance, unbiased meaning $\hat{E} = E$.
Also because of its closed form and simple: $\hat{\beta} = (X^TX)^{-1}Xy$ is linear to y.

4.3 Prediction vs inference

Prediction is asking given X, what is the best guess for Y.
R^2 is an important metric where it says how much variance in the given data is explained by your model.

$$R^2 = 1 - \frac{\mathbb{E}((y - \hat{y})^2)}{\mathbb{E}((y - \bar{y})^2)}$$

R^2 on training set ranges between $[0, 1]$, and $[-\inf, 1]$ OOS.
Inference is learning the data generation process: the underlying relationship between X and Y.
Standard error is an important metric here.
Hypothesis testing is often done in inference.
What happens if two features are highly correlated?
In a linear regression, you often see offseting coefficients, this causes standard error to go up. Correlated features can be mitigated by regularization, which also results in a biased estimator.