cairo.thws
Center for Artificial Intelligence
Technical University of
Applied Sciences
Würzburg-Schweinfurt

Magda Gregorová
magda.gregorova@thws.de

Meausure-theory view of probability
handwavy and informal

December 1, 2025

Outline

cairo.thws
Center for Artificial Intelligence
Technical University of Applied Sciences Würzburg-Schweinfurt

1. σ-algebra $\mathcal{S}$ is collection of measurable sets (closed under complements and countable unions)
2. For practical purposes: think of $\mathcal{S}$ as "all reasonable subsets"
3. Lebesgue measure generalizes notions of length, area, volume
4. Probability measure is just a finite measure normalized to total mass 1
5. This abstraction allows us to talk about probability rigorously, but we'll soon move to more practical objects

Measures - assigning mass to sets

cairo.thws
Center for Artificial Intelligence
Technical University of Applied Sciences Würzburg-Schweinfurt

Measurable space $(S, \mathcal{S})$

- S - set (e.g., $\mathbb{R}^d$, discrete set, etc.)
- $\mathcal{S}$ - σ-algebra on S (collection of measurable subsets of S)
  - closed under complements and countable unions
  - contains $\emptyset$ and S

Measure μ on $(S, \mathcal{S})$ - function $\mu : \mathcal{S} \to [0, \infty]$

- $\mu(\emptyset) = 0$
- countable additivity: for disjoint $\{A_i : i \in I\} \subseteq \mathcal{S}$, $\mu(\bigcup_{i \in I} A_i) = \sum_{i \in I} \mu(A_i)$

Examples:

- counting measure: $\#(A)$ = number of elements in A
- Lebesgue measure on $\mathbb{R}^d$: $\lambda(A)$ = volume of A

1. Probability measure is just a normalized measure - total mass equals 1
2. The abstract probability space $(S, \mathcal{S}, \mathbb{P})$ is often denoted $(\Omega, \mathcal{F}, \mathbb{P})$ in literature
3. Random experiment: think coin flip, dice roll, or any random process
4. Outcome $s \in S$: specific result of the experiment (e.g., "heads", or specific image)
5. Event $A \in \mathcal{S}$: set of outcomes we're interested in (e.g., "at least 2 heads in 3 flips")
6. This level of abstraction separates the experiment from what we measure
7. But as we'll see next, we usually work with random variables that map outcomes to values we care about

Probability measure

Probability space $(S, \mathcal{S}, \mathbb{P})$

- S - sample space (possible outcomes of random experiment)
- $\mathcal{S}$ - σ-algebra on S (collection of events - sets of outcomes)
- $\mathbb{P} : \mathcal{S} \to [0,1]$ - probability measure
  - normalization: $\mathbb{P}(S) = 1$
  - countable additivity (same as before)

Interpretation:

- random experiment produces outcome $s \in S$
- event $A \in \mathcal{S}$ is set of outcomes
- $\mathbb{P}(A)$ is probability that outcome lies in A

Abstract formalism: $(S, \mathcal{S}, \mathbb{P})$ gives rigorous framework
In practice: we work with random variables and their distributions

1. Normalization is just dividing by total mass
2. This shows probability measures are special cases of finite measures
3. Connection to statistical physics and energy-based models
4. Partition function Z is the normalizing constant
5. When we transform measures, we transform probabilities - this is key for flows

Relationship: measures and probability

cairo.thws
Center for Artificial Intelligence
Technical University of Applied Sciences Würzburg-Schweinfurt

Any finite measure can be normalized:

- measure space $(S, \mathcal{S}, \mu)$ with $\mu(S) < \infty$
- define $\mathbb{P}(A) = \frac{\mu(A)}{\mu(S)}$ for $A \in \mathcal{S}$
- then $(S, \mathcal{S}, \mathbb{P})$ is probability space

Conversely: any probability measure can be scaled

- probability space $(S, \mathcal{S}, \mathbb{P})$
- for any $c > 0$, $\mu = c \cdot \mathbb{P}$ is finite measure with $\mu(S) = c$

Why this matters:

- energy-based models: unnormalized measures $\mu(A) = \int_A e^{-E(s)} ds$
- normalizing gives probability: $\mathbb{P}(A) = \frac{1}{Z} \int_A e^{-E(s)} ds$ where $Z = \int_S e^{-E(s)} ds$
- change of variables preserves this relationship

1. Measurability is technical requirement - ensures we can measure probabilities after transformation
2. For continuous functions between nice spaces (e.g., $\mathbb{R}^d$ with Borel sets), measurability is automatic
3. The value space T is what we actually care about - numbers, vectors, images, etc.
4. Abstract sample space S is often just formal device
5. In deep learning: we care about generated image x, not the random seed that produced it
6. Capital letter = random variable (random), lowercase = realization (fixed value)

Random variable - moving to value space

cairo.thws
Center for Artificial Intelligence
Technical University of Applied Sciences Würzburg-Schweinfurt

Random variable $X : S \to T$

- $(S, \mathcal{S}, \mathbb{P})$ - probability space (random experiment)
- $(T, \mathcal{T})$ - measurable space (value space, e.g., $\mathbb{R}^d$)
- X - measurable function: $X^{-1}(B) \in \mathcal{S}$ for all $B \in \mathcal{T}$

Interpretation:

- random experiment produces outcome $s \in S$
- we observe value $x = X(s) \in T$ - called realization of X
- X maps abstract outcomes to concrete values we care about

Notation convention:

- X - random variable (the function itself)
- x - realization (a specific value in T)

1. Concrete example to ground the abstraction
2. Note: multiple outcomes in S can give same value in T (e.g., (2,6) and (3,5) both give sum 8)
3. This is why we need the distribution P_X on T - it aggregates probabilities
4. Example: $P_X(\{7\}) = \frac{6}{36}$ because 6 outcomes map to sum of 7

Example: dice sum

cairo.thws
Center for Artificial Intelligence
Technical University of Applied Sciences Würzburg-Schweinfurt

Roll two dice and sum them

Setup:

- sample space: $S = \{(1,1),(1,2),\ldots,(6,6)\}$ (36 outcomes)
- probability: $\mathbb{P}(\{(i,j)\}) = \frac{1}{36}$ for each outcome
- random variable: $X(s_1, s_2) = s_1 + s_2$
- value space: $T = \{2,3,4,\ldots,12\}$

Realization:

- if experiment gives $(3,5) \in S$, then $x = X(3,5) = 8$
- $x = 8$ is a realization of random variable X

Key point:

- we care about the sum (value in T), not which dice showed what (outcome in S)
- this motivates working with distribution on T directly

1. The distribution P_X is completely determined by X and $\mathbb{P}$
2. Push-forward: probability "flows" from S to T through X
3. Once we have P_X, we can often forget about $(S, \mathcal{S}, \mathbb{P})$
4. In practice: we specify P_X directly without ever mentioning S
5. "X N(0,1)" directly specifies distribution on $\mathbb{R}$, no mention of underlying experiment

Distribution of random variable

cairo.thws
Center for Artificial Intelligence
Technical University of Applied Sciences Würzburg-Schweinfurt

Given: $(S, \mathcal{S}, \mathbb{P})$ **and random variable** $X : S \to T$

Distribution (law) of X**: probability measure** P_X **on** $(T, \mathcal{T})$

$$P_X(B) = \mathbb{P}(X \in B) = \mathbb{P}(\{s \in S : X(s) \in B\}), \quad B \in \mathcal{T}$$

Interpretation:

- $P_X(B)$ = probability that X takes value in set B
- P_X lives on value space T, not on sample space S
- P_X is the push-forward of $\mathbb{P}$ by X

Notation: $X \sim P_X$ means "X has distribution P_X"

Key point: P_X is a probability measure on T - we can work with it directly!

1. This is the key conceptual shift for students
2. Abstract probability space is formal machinery, but not where we actually work
3. In generative modeling: we always work directly with distributions on $\mathbb{R}^d$
4. The triplet $(\Omega, \mathscr{F}, \mathbb{P})$ is rarely mentioned in ML papers
5. Examples: "$z \sim p_{\text{prior}}$", "$x \sim p_{\text{data}}$" - these directly specify distributions
6. We're working with push-forward measures, but we specify them directly
7. Next section: we'll see different ways to represent these distributions (measure, CDF, density)

Working directly with distributions

cairo.thws
Center for Artificial Intelligence
Technical University of Applied Sciences Würzburg-Schweinfurt

Two perspectives on probability:

Formal perspective:

- start with abstract probability space $(S, \mathscr{S}, \mathbb{P})$
- define random variable $X : S \to T$
- derive distribution P_X on value space T

Practical perspective (what we actually do):

- directly specify distribution P_X on value space T
- $(S, \mathscr{S}, \mathbb{P})$ is implicit, often $S = T$ and $X =$ identity
- notation: "$X \sim \mathcal{N}(0, I)$" directly defines Gaussian on $\mathbb{R}^d$

From now on: we work with distributions on value spaces $T = \mathbb{R}^d$

1. This is a key conceptual point that students often miss
2. The measure, CDF, and density (when it exists) all describe the same distribution
3. Other representations exist: characteristic function $\phi_X(t) = E[e^{itX}]$, moment generating function $M_X(t) = E[e^{tX}]$, quantile function, etc.
4. We focus on these three because: (1) measure is fundamental, (2) CDF always exists, (3) density is what we compute with in ML
5. Measure is most general, CDF always exists, density only sometimes exists
6. In ML we mostly work with densities, but need to understand when they exist
7. Component-wise for CDF in $\mathbb{R}^d$: $F_X(x_1, \ldots, x_d) = P_X((-\infty, x_1] \times \cdots \times (-\infty, x_d])$

Three primary representations of a distribution

Given random variable X with values in $\mathbb{R}^d$:

1. Probability measure P_X - operates on sets
- $P_X(A)$ - probability that $X \in A$

2. Cumulative distribution function (CDF) F_X - operates on points
- $F_X(x) = P_X((-\infty, x])$, always exists

3. Probability density function (PDF) p_X - operates on points
- $P_X(A) = \int_A p_X(x)\,dx$ when it exists

Three views of the same distribution
(other representations exist: characteristic function, MGF, etc.)

1. Riemann integration: only over intervals, uses length/area/volume
2. Measure integration: over any measurable set, uses any measure
3. The measure μ determines how we "weigh" different parts of the space
4. This is a generalization that will let us work with probability measures

Interlude: integration with respect to a measure

cairo.thws
Center for Artificial Intelligence
Technical University of Applied Sciences Würzburg-Schweinfurt

Why we need this: to connect measures with densities (functions)

Riemann integration (what you know):

$$\int_a^b f(x)\,dx = \text{area under curve}$$

Generalization - integration w.r.t. measure μ:

$$\int_A f\,d\mu$$

- A - any measurable set
- μ - measure (weighs different parts of space)

Intuition: weighted sum of f over set A, weighted by measure μ

1. Lebesgue measure is the "natural" measure on Euclidean space
2. For nice functions on nice sets: Riemann = Lebesgue integration
3. But Lebesgue integration works for much wider class of functions and sets
4. The notation dx is shorthand for $d\lambda(x)$ (Lebesgue measure)
5. Counting measure: discrete sums become integrals!
6. Example with probability: $\int_{\mathbb{R}^d} f \, dP_X = E[f(X)]$ works for both discrete and continuous X
7. This framework unifies discrete and continuous cases - no need for separate notation

Interlude: Lebesgue measure and notation

cairo.thws
Center for Artificial Intelligence
Technical University of Applied Sciences Würzburg-Schweinfurt

Lebesgue measure λ on $\mathbb{R}^d$

- generalizes length/area/volume
- $\lambda([a,b]) = b - a$ (length), $\lambda(A)$ = volume of A
- integration: $\int_A f \, d\lambda = \int_A f(x)\, dx$ (familiar!)

Counting measure # on discrete/countable sets

- $\#(A)$ = number of elements in A
- integration: $\int_A f \, d\# = \sum_{x \in A} f(x)$ (discrete sum!)

Key insight: integral notation unifies discrete and continuous

$$\int_A f \, d\mu \quad \begin{cases} = \int_A f(x)\, dx & \text{continuous (Lebesgue)} \\ = \sum_{x \in A} f(x) & \text{discrete (counting)} \end{cases}$$

1. We work with measurable sets - for practical purposes, "all reasonable subsets"
2. Integration w.r.t. probability measure uses the framework from the interlude
3. The normalization property connects to what we saw in Section 1: any finite measure can be normalized
4. The notation $\int_A dP_X$ means $\int_A 1\, dP_X$ - we're integrating the constant function 1 over set A, which gives the total probability mass in A
5. Measure is the fundamental object, but we often use CDF or density for computation
6. for continuous distributions: $P_X(\{x_0\}) = 0$ (defining property: no point masses)
7. Intuition: $P_X(\{x_0\}) = \int_{\{x_0\}} dP_X = 0$ because the set has zero measure
8. This is why we can't talk about "probability at a point" for continuous distributions - only density

Probability measure - the fundamental object

cairo.thws
Center for Artificial Intelligence
Technical University of Applied Sciences Würzburg-Schweinfurt

Probability measure P_X on $\mathbb{R}^d$

- assigns probability to measurable sets: $P_X(A) = \mathbb{P}(X \in A)$
- as integration: $P_X(A) = \int_A dP_X$

Key properties:

- normalization: $P_X(\mathbb{R}^d) = \int_{\mathbb{R}^d} dP_X = 1$
- for continuous distributions: $P_X(\{x_0\}) = \int_{\{x_0\}} dP_X = 0$ (single points have zero probability)

Properties:

- most fundamental representation
- works for discrete, continuous, and mixed distributions

1. CDF is unique representation - different distributions have different CDFs
2. Definition uses measure integration: integral of constant function 1 over the interval
3. For multidimensional case, CDF is defined component-wise
4. CDF can have jumps (discrete distributions) or be smooth (continuous distributions)
5. Derivative of CDF gives density (when density exists): $p_X(x) = \frac{dF_X}{dx}$
6. In practice: we rarely work with CDF directly in ML, but it's theoretically important

Cumulative distribution function (CDF)

CDF $F_X : \mathbb{R} \to [0,1]$

Definition:

$$F_X(x) = P_X(X \leq x) = P_X((-\infty, x]) = \int_{(-\infty,x]} dP_X$$

Key properties:

- always exists (for any distribution)
- $F_X(-\infty) = 0$, $F_X(\infty) = 1$
- non-decreasing, right-continuous
- recovers probabilities: $P_X((a,b]) = F_X(b) - F_X(a)$

Note: for $\mathbb{R}^d$: $F_X(x_1, \ldots, x_d) = P_X((-\infty, x_1] \times \cdots \times (-\infty, x_d])$

1. Density measures probability per unit volume
2. Radon-Nikodym derivative: formal name for this limit
3. This is why density can be > 1 (it's a rate, not a probability)
4. Absolute continuity ensures this limit exists

Probability density function (PDF)

cairo.thws
Center for Artificial Intelligence
Technical University of Applied Sciences Würzburg-Schweinfurt

Setup:

- $P_X(A)$ gives probability over sets
- for continuous distributions: $P_X(\{x\}) = 0$ at points

What happens as we shrink a set around x?
The ratio of probability to volume as region shrinks:

$$p_X(x) = \lim_{\epsilon \to 0} \frac{P_X(B_\epsilon(x))}{\lambda(B_\epsilon(x))} = \frac{dP_X}{d\lambda}(x) \qquad \text{density = Radon-Nikodym derivative}$$

where λ is Lebesgue measure (volume)

Key property: $P_X(A) = \int_A p_X(x)\,dx$

Note: density does not always exist (requires absolute continuity)

1. Measure is most fundamental but assigns probabilities to sets, not points
2. CDF always exists and can be evaluated at points
3. PDF only exists for absolutely continuous distributions
4. In ML: we work with PDFs (assume absolute continuity)
5. PDF is most convenient: can evaluate at points and optimize

Relationship between the three

Comparison of the three representations:

	Measure P_X	**CDF** F_X	**PDF** p_X
Always exists?	yes	yes	no
Domain	sets	points	points
Range	$[0,1]$	$[0,1]$	$[0,\infty)$

How to convert between them:

From	**To**	**Formula**
P_X	F_X	$F_X(x) = P_X((-\infty, x])$
P_X	p_X	$p_X = \frac{dP_X}{d\lambda}$ (if abs. cont.)
p_X	P_X	$P_X(A) = \int_A p_X \, d\lambda$
p_X	F_X	$F_X(x) = \int_{-\infty}^{x} p_X(t) \, dt$
F_X	p_X	$p_X(x) = \frac{dF_X}{dx}$ (if smooth)

1. This is the fundamental question in generative modeling
2. We transform a simple distribution to get a complex one
3. The transformation g is typically a neural network
4. We work directly with distributions on value spaces (as established in Section 2)
5. No need to refer back to abstract probability space $(S, \mathcal{S}, \mathbb{P})$

Transformations of random variables

Setup:

- X - random variable with values in T (e.g., $\mathbb{R}^d$)
- P_X - distribution of X on T
- $g : T \to U$ - measurable function (transformation)
- $Y = g(X)$ - transformed random variable with values in U

Question: If we know P_X, what is the distribution P_Y of $Y = g(X)$?

Examples in ML:

- $X \sim \mathcal{N}(0, I)$ (simple), g is neural network, $Y = g(X)$ (complex model distribution)
- data transformation: X is data, g encodes to latent space

1. Push-forward is how distributions transform under functions
2. The pre-image $g^{-1}(C)$ is the set of points in T that map to C
3. Diagram shows: regular grid in T becomes warped in U, but probability is preserved
4. This is well-defined because g is measurable
5. In ML: g is neural network, P_X is simple distribution (e.g., Gaussian), P_Y is complex model distribution
6. We're working directly with distributions P_X and P_Y, not going back to abstract probability space

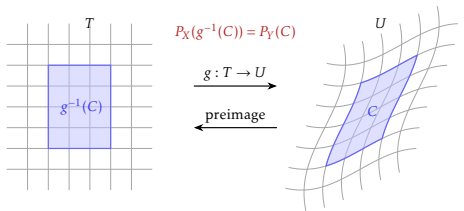
Push-forward measure

Answer: distribution of $Y = g(X)$ is the push-forward of P_X by g

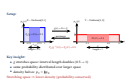
Definition: P_Y on U defined by

$$P_Y(C) = P_X(g^{-1}(C)) \quad \text{for } C \subseteq U$$

where $g^{-1}(C) = \{x \in T : g(x) \in C\}$ is the pre-image



Intuition: probability of Y landing in C equals probability of X being in pre-image

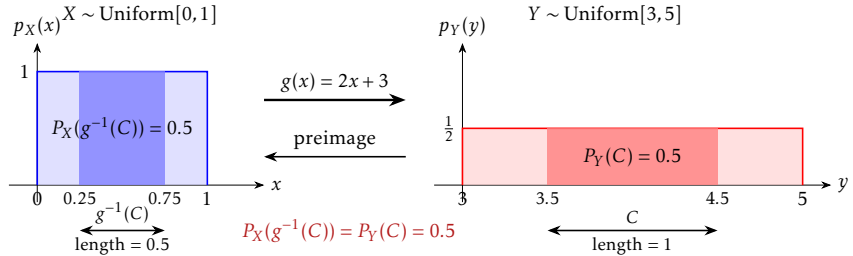


1. This is a simple concrete example showing push-forward
2. The preimage calculation: solve $a \leq 2x + 3 \leq b$ for x
3. Uniform on $[0,1]$: $P_X([a,b]) = b - a$ when $[a,b] \subseteq [0,1]$
4. After transformation: still uniform, but on $[3,5]$ instead of $[0,1]$
5. The scaling factor 2 will become important when we look at densities (Jacobian)
6. This example shows: push-forward preserves the "shape" but changes location and scale

Example: uniform distribution and affine transformation

cairo.thws
Center for Artificial Intelligence
Technical University of Applied Sciences Würzburg-Schweinfurt

Setup: $X \sim \text{Uniform}[0,1]$, transformation $g(x) = 2x + 3$, find distribution of $Y = g(X)$



Key insight:

- g stretches space: interval length doubles $(0.5 \to 1)$
- same probability distributed over larger space
- density halves: $p_Y = \frac{1}{2} p_X$

Stretching space $\Rightarrow$ lower density (probability conserved)

1. The uniform example was special - constant density, linear transformation
2. In ML: we use complex distributions (Gaussian, data distributions) and complex transformations (neural networks)
3. Need to understand how densities transform in general
4. This is the change of variables formula - fundamental for normalizing flows, diffusion models, etc.

From push-forward to change of variables

cairo.thws
Center for Artificial Intelligence
Technical University of Applied Sciences Würzburg-Schweinfurt

What we saw: uniform distribution + affine transformation

- $X \sim \text{Uniform}[0,1]$, $g(x) = 2x + 3$, $Y = g(X)$
- stretching by factor 2 $\Rightarrow$ density halves
- easy to compute: $p_Y = \frac{1}{2}$

General question:
Given arbitrary distribution p_X and transformation g, how do we compute p_Y?

Challenge:

- not just uniform distributions
- not just affine transformations
- need general formula relating p_Y to p_X and g

Goal: derive the change of variables formula with Jacobian determinant

1. Key step: specialize push-forward to the set $(-\infty, y]$ which gives CDF
2. Increasing function: preimage preserves order, so $g^{-1}((-\infty, y]) = (-\infty, g^{-1}(y)]$
3. Decreasing function: preimage reverses order, so $g^{-1}((-\infty, y]) = [g^{-1}(y), \infty)$
4. This gives us exact formulas relating CDFs
5. Next: differentiate to get density formula

Step 1: From push-forward to CDF

Recall: push-forward at measure level

$$P_Y(C) = P_X(g^{-1}(C))$$

Choose special set: $C = (-\infty, y]$, then $P_Y((-\infty, y]) = F_Y(y)$ - CDF!

What is $g^{-1}((-\infty, y])$?

- If g increasing: $g^{-1}((-\infty, y]) = (-\infty, g^{-1}(y)]$

$$\Rightarrow F_Y(y) = F_X(g^{-1}(y))$$

- If g decreasing: $g^{-1}((-\infty, y]) = [g^{-1}(y), \infty)$

$$\Rightarrow F_Y(y) = 1 - F_X(g^{-1}(y))$$

1. Chain rule for differentiation
2. PDF is derivative of CDF
3. Increasing case: derivative is positive
4. Decreasing case: minus sign from derivative of (1-F), but derivative of g^{-1} is negative, so they cancel to positive
5. Absolute value handles both cases: density must be positive!
6. This explains why we need absolute value in change of variables formula

Step 2: Differentiate to get density

Case 1: g increasing

$$p_Y(y) = \frac{dF_Y}{dy}(y) = \frac{d}{dy}F_X(g^{-1}(y)) = p_X(g^{-1}(y)) \cdot \frac{dg^{-1}}{dy}(y)$$

Note: $\frac{dg^{-1}}{dy} > 0$ for increasing g

Case 2: g decreasing

$$p_Y(y) = \frac{dF_Y}{dy}(y) = \frac{d}{dy}[1 - F_X(g^{-1}(y))] = -p_X(g^{-1}(y)) \cdot \frac{dg^{-1}}{dy}(y)$$

Note: $\frac{dg^{-1}}{dy} < 0$ for decreasing g, so $-\frac{dg^{-1}}{dy} > 0$

Change of varible formula: $$p_Y(y) = p_X(g^{-1}(y)) \left| \frac{dg^{-1}}{dy}(y) \right|$$

1. This derivation uses change of variables formula from calculus directly
2. More direct than the CDF approach, but requires knowing change of variables
3. Absolute value appears because we integrate over sets (not oriented intervals)
4. If integrals are equal for all sets C, integrands must be equal
5. Same formula as before - two derivations confirm the result!

Alternative derivation: direct from push-forward

Recall: push-forward with densities (assuming they exist)

$$P_Y(C) = \int_C p_Y(y)\,dy = \int_{g^{-1}(C)} p_X(x)\,dx = P_X(g^{-1}(C))$$

Change integration variable: substitute $x = g^{-1}(y)$, then $dx = \frac{dg^{-1}}{dy}(y)\,dy$

$$\int_C p_Y(y)\,dy = \int_{g^{-1}(C)} p_X(x)\,dx = \int_C p_X(g^{-1}(y)) \left|\frac{dg^{-1}}{dy}(y)\right| dy$$

Since this holds for all C, integrands must be equal:

Change of varible formula: $$p_Y(y) = p_X(g^{-1}(y)) \left|\frac{dg^{-1}}{dy}(y)\right|$$

1. Both examples use same transformation, different distributions
2. Uniform: confirms intuitive result from Section 4
3. Gaussian: mean shifts by 3, variance scales by $2^2 = 4$
4. The derivative $\frac{1}{2}$ accounts for the stretching by factor 2
5. General principle: affine transformation is easy to compute

Examples: applying the formula

cairo.thws
Center for Artificial Intelligence
Technical University of Applied Sciences Würzburg-Schweinfurt

Setup: transformation $g(x) = 2x + 3$, so $g^{-1}(y) = \frac{y-3}{2}$ and $\frac{dg^{-1}}{dy} = \frac{1}{2}$

Example 1: Uniform

$$X \sim \text{Uniform}[0,1] : p_X(x) = 1$$

$$p_Y(y) = 1 \cdot \frac{1}{2} = \frac{1}{2} \quad \Rightarrow \quad Y \sim \text{Uniform}[3,5]$$

Example 2: Gaussian

$$X \sim \mathcal{N}(0,1) : p_X(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}$$

$$p_Y(y) = \frac{1}{\sqrt{2\pi}} e^{-(y-3)^2/8} \cdot \frac{1}{2} \quad \Rightarrow \quad Y \sim \mathcal{N}(3,4)$$

1. Jacobian matrix: natural generalization of derivative to multiple dimensions
2. Determinant of Jacobian: measures how volumes scale under transformation
3. Next: geometric interpretation of what the Jacobian determinant means

Generalization to $\mathbb{R}^d$

In 1D: derivative $g'(x)$ measures how g stretches/compresses locally

In $\mathbb{R}^d$: Jacobian matrix $J_g(x) = \frac{\partial g}{\partial x}(x)$ generalizes the derivative

- $J_g(x)$ is $d \times d$ matrix of all partial derivatives
- $\det J_g(x)$ measures local volume scaling factor

Change of integration variables in $\mathbb{R}^d$:

$$dx = \left|\det J_{g^{-1}}(y)\right| dy$$

Change of varible formula: $p_Y(y) = p_X(g^{-1}(y))\left|\det J_{g^{-1}}(y)\right|$

1. Jacobian is the best linear approximation of g near x
2. In 1D: $g(x+\Delta x) \approx g(x) + g'(x)\Delta x$ (tangent line)
3. Linear maps scale volumes by their determinant
4. This justifies the change of variables formula

Jacobian as linear approximation

Near point x, transformation g is approximately linear:

$$g(x+\Delta x) \approx g(x) + J_g(x) \cdot \Delta x$$

where $J_g(x) = \frac{\partial g}{\partial x}(x)$ is the Jacobian matrix

Volume scaling:

- linear maps scale volumes by determinant
- $J_g(x)$ is linear approximation at x
- therefore: $|\det J_g(x)|$ = local volume scaling factor

Consequence for small region around x:

$$\text{volume}(g(B_\epsilon(x))) \approx |\det J_g(x)| \cdot \text{volume}(B_\epsilon(x))$$

1. This is where all the theory pays off
2. Simple distribution: easy to sample from, easy to evaluate
3. Neural network: flexible enough to model complex data
4. Change of variables: connects the two via tractable density
5. Next: see how each method uses this formula

Connection to generative models

cairo.thws
Center for Artificial Intelligence
Technical University of Applied Sciences Würzburg-Schweinfurt

What we derived:

$$p_Y(y) = p_X(g^{-1}(y)) \left| \det J_{g^{-1}}(y) \right|$$

The generative modeling setup:

- Start: simple distribution p_X (e.g., Gaussian $\mathcal{N}(0, I)$)
- Transform: via learnable function g_θ (neural network)
- Goal: complex distribution p_Y that matches data distribution p_{data}

Key advantage: change of variables formula lets us compute p_Y explicitly!

Three approaches: normalizing flows, diffusion models, flow matching

1. Normalizing flows: direct application of change of variables
2. The name "flow": transformation flows probability from prior to data
3. Examples: RealNVP, Glow, MAF, IAF
4. Challenge: designing architectures where g^{-1} and $\det J$ are tractable
5. Popular: coupling layers, autoregressive flows

Normalizing flows

Setup:

- $X \sim p_{prior}$ (simple, e.g., $\mathcal{N}(0, I)$)
- $Y = g_\theta(X)$ where g_θ is invertible neural network
- Want: $p_Y \approx p_{data}$

Key property: can compute density explicitly

$$p_Y(y;\theta) = p_{prior}(g_\theta^{-1}(y)) \left| \det J_{g_\theta^{-1}}(y) \right|$$

Training: maximize log-likelihood on data

$$\max_\theta \sum_{y \in \text{data}} \log p_Y(y;\theta)$$

Requires:

- g_θ invertible
- Jacobian determinant computable

1. Diffusion: more indirect use of change of variables
2. Forward process defined by known transformations (adding Gaussian noise)
3. Reverse process learned to approximate inverse
4. Examples: DDPM, Score-based models
5. Very successful in practice (Stable Diffusion, DALL-E 2, etc.)

Diffusion models

cairo.thws
Center for Artificial Intelligence
Technical University of Applied Sciences Würzburg-Schweinfurt

Setup:

- Forward process: gradually add noise $X_0 \to X_1 \to \cdots \to X_T$
- Each step: $X_{t+1} = g_t(X_t) + \text{noise}$
- Reverse process: learn to denoise $X_T \to \cdots \to X_1 \to X_0$

Role of change of variables:

- Forward: transformations with known densities
- Reverse: approximate inverse transformations
- Change of variables connects forward and reverse densities

Key difference from flows:

- Don't need invertible network
- Don't need to compute Jacobian determinant
- Trade exact likelihood for flexibility

1. Flow matching: continuous-time version of change of variables
2. Instead of discrete transformations, continuous flow
3. Continuity equation: how density changes along flow
4. Examples: Conditional Flow Matching, Rectified Flow
5. Very recent and active area of research
6. Unifies many concepts from flows and diffusion

Flow matching

cairo.thws
Center for Artificial Intelligence
Technical University of Applied Sciences Würzburg-Schweinfurt

Setup:

- Continuous-time flow: $\frac{dx_t}{dt} = v_t(x_t)$ where v_t is velocity field
- Connects p_0 (simple) to p_1 (data) via continuous path

Change of variables in continuous time:

Instantaneous change of variables + continuity equation:

$$\frac{\partial p_t}{\partial t} + \nabla \cdot (v_t p_t) = 0$$

Related to Jacobian: $\frac{\partial \log p_t}{\partial t} = -\nabla \cdot v_t$

Key idea:

- Learn velocity field v_t
- Flow transports probability from p_0 to p_1
- Change of variables formula in differential form