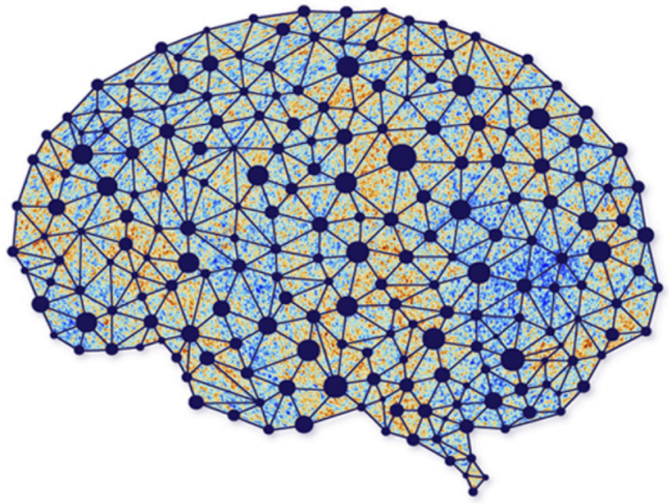
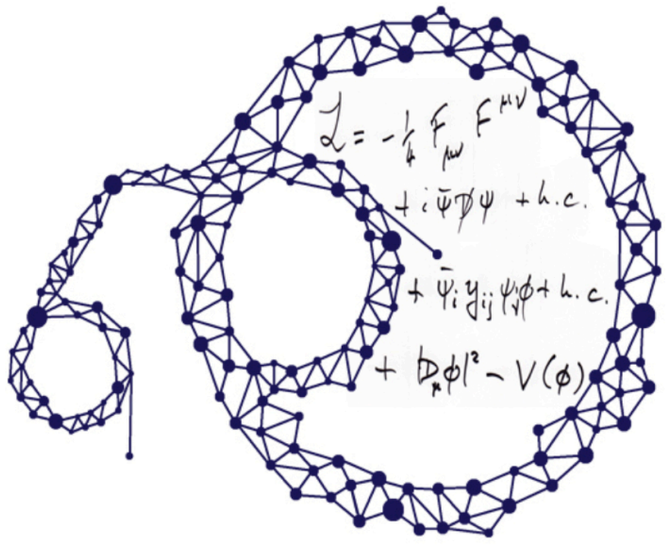
Physics 361 - Machine Learning in Physics

Lecture 2 – Background

Jan. 23rd 2025



AI
∩
Universe



Moritz Münchmeyer

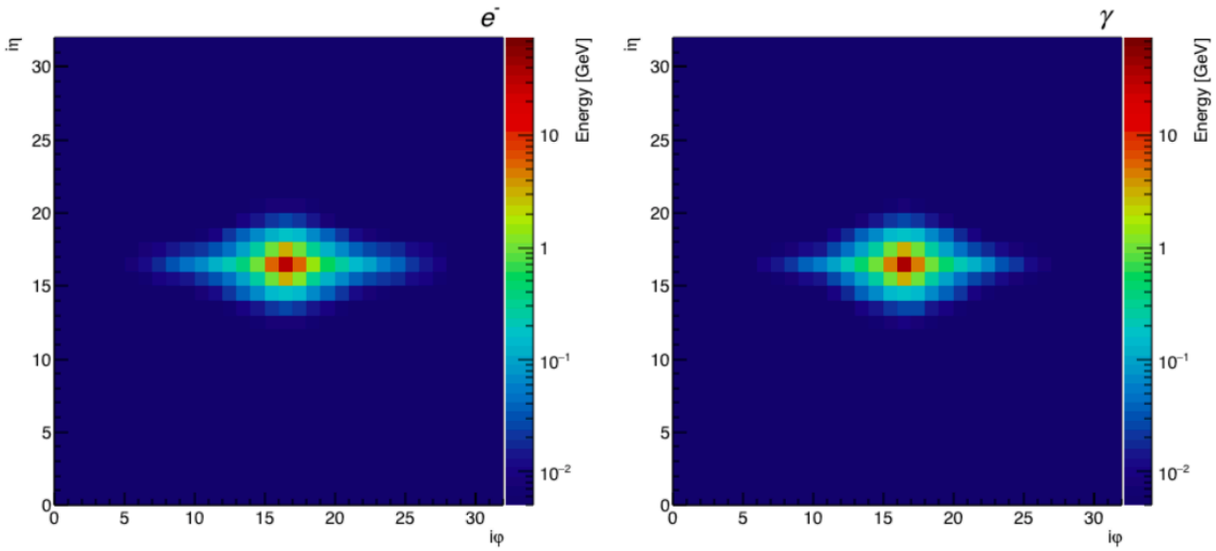
Some Examples of ML in the Physical Sciences (cont.)

Classifying Events and Objects

- Examples:
  - LHC particle collisions. ML has a long history in particle physics, reaching back several decades.

Arxiv: 1807.11916
End-to-End Physics Event Classification with CMS Open Data

(Here and below I select papers somewhat at random, there are MANY other good papers in each domain)



(a) Average electron shower (b) Average photon shower

Figure 1: e/γ showers averaged over 50k showers. The e shower is slightly more spread out in ϕ-in addition to being slightly asymmetric-due to bremsstrahlung effects.

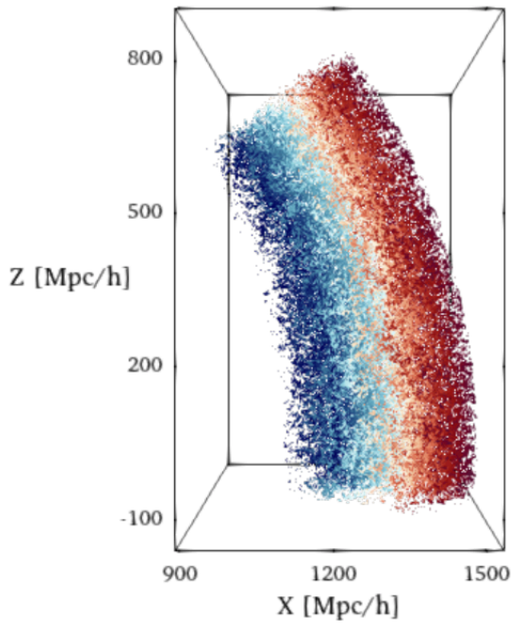
- Ice cube particle shower classification. E.g. 2209.03042
- Galaxy type classification. In the past, different galaxy types were classified by researchers by eye. Not possible with millions of galaxies.

Measuring physical parameters

- It is often not clear how to measure a parameter from a collection of data.
- If we have reliable simulations, we can train a neural network to perform the measurement, using supervised learning.
- Example: Measuring cosmological parameters (age of the universe, amount of dark matter etc.) from a galaxy survey

SimBig project 2211.00723

Galaxy data



CNN

Parameter Measurements

- Main challenge: Reliability of training data.

Simulation-based Inference

- When we measure parameters, we also need error bars (or better the full posterior).

- **Simulation-based inference is the process of finding parameters of a simulator from observations, probabilistically**.

- In “traditional” data analysis in physics we often make analytical assumptions of the statistics of an observable, most commonly that it is Gaussian distributed.

- With machine learning one can **learn the probability distribution of observables from simulations.** In a Bayesian analysis, the likelihood or the posterior can be learned from simulations.

- This is usually done using a **Neural Density Estimator, such as a Normalizing flow**.

- See e.g. arxiv:1911.01429 The frontier of simulation-based inference

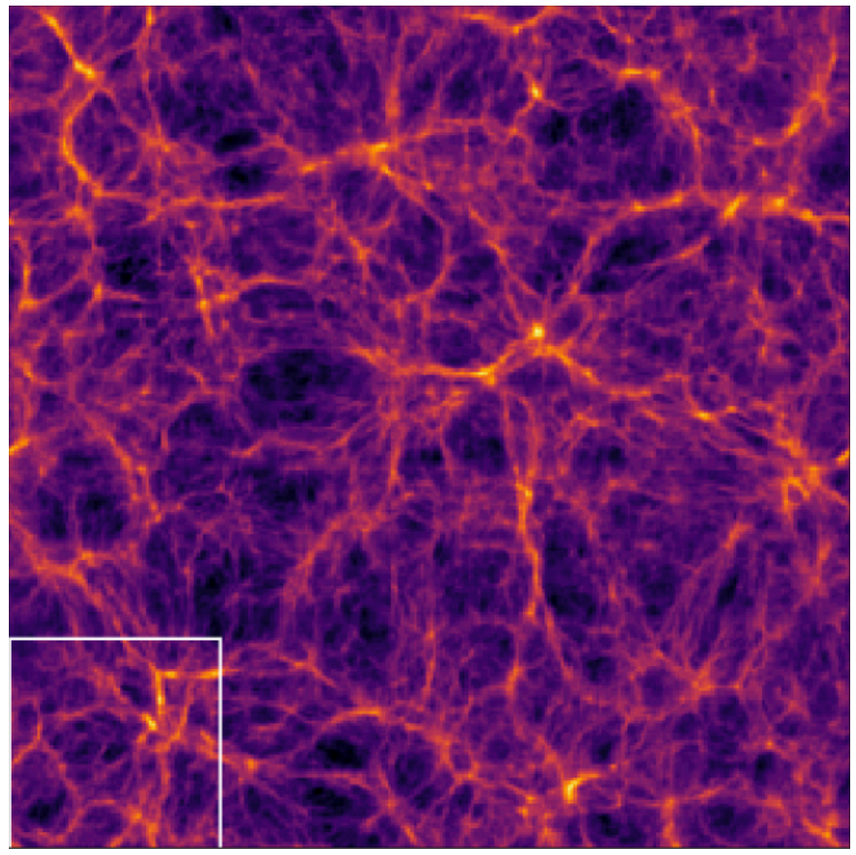
Generating Simulations / Emulators

- Neural networks can be used as **surrogate models to replace computationally expensive simulations**. These are often called Emulators.

- Once trained on data or simulations, an emulator can make new “simulations” much faster.

  - Example from my own research:

    - Generating 3d simulations of the matter distribution of the universe using a diffusion model. (Arxiv: 2311.05217)

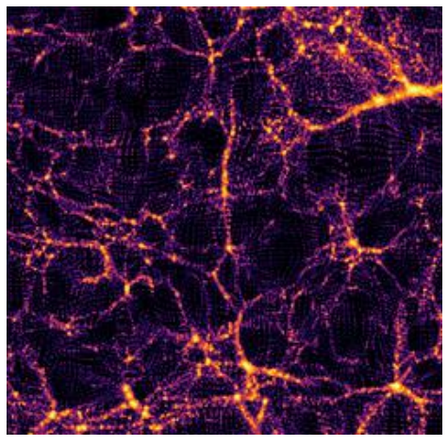


- Machine Learning is often used to **speed up classical methods**.

Auto-differentiation without ML

- To train neural networks, computational techniques were developed that can train models with **billions of free parameters**. This is done with auto-differentiation libraries such as
  - **PyTorch**
  - **JAX**
  - **Tensorflow**

- This software is useful in physics **even if you don't use any machine learning.**
- Physicists **re-write their codes in auto differentiable language**, which allows efficient optimization with respect to any parameters. Some examples from my field:
  - CosmoJax, a differentiable cosmology library
  - Differentiable cosmology simulations, e.g. pmwd



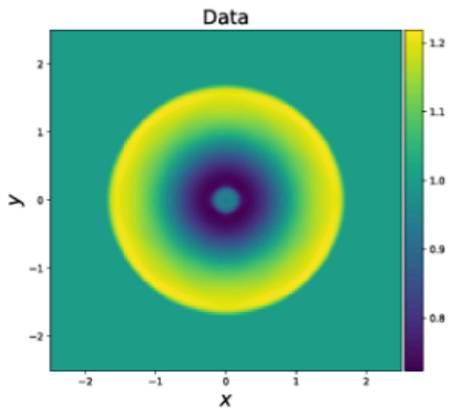
Clustering and Anomaly detection

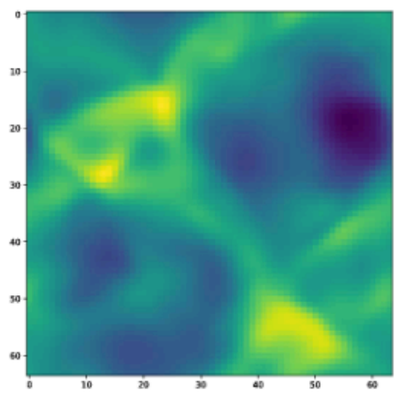
- **How can we organize a large data set of events or objects into classes of similar objects? Clustering and dimensionality reduction algorithms.**
  - Classic k-means is still very useful! E.g. stellar populations.
  - Clustering can also happen in the “latent space” of a generative model.
  - Data visualization, e.g. t-SNE
- **How can we find something “new” without knowing what to look for? Anomaly detection!**
  - Humans are pretty good at anomaly detection by eye, but data sets are too large to be inspected that way and the anomaly may only be visible in the right data representation.
  - Anomalies have been found in archival data, long after the data was taken (example: Fast Radio Bursts). Perhaps there is something exciting hidden in existing data.
  - Unsupervised learning can be used to classify existing events or objects. If an object is not close to any known class, it is flagged as an anomaly.

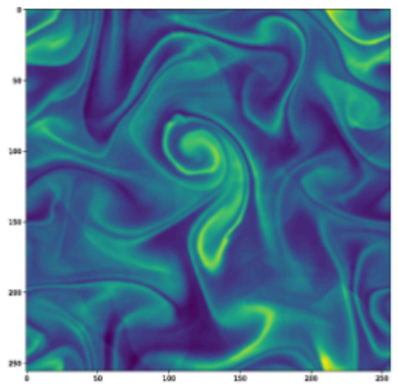
Solving PDEs and Inverse Problems

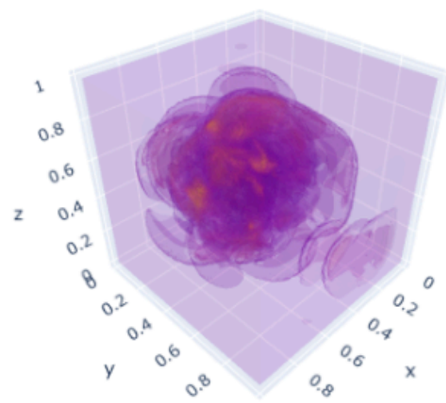
- Many problems in physics amount to **solving a complicated set of partial differential equations (PDE).** There are various ways to use NN for that.

  - Examples (from the PDEBench data set):









- In **Inverse Problems**, one wants to find the input data that produced a specific output. That can mean removing noise or undoing a non-linear evolution. Often they are ill-conditioned and need to be regularized.

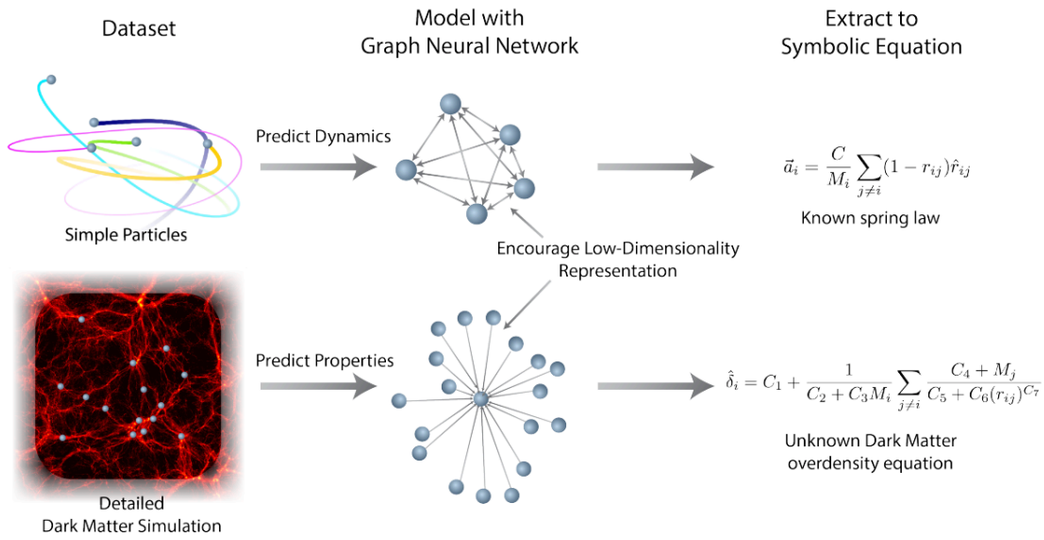
- Neural Networks are being trained to solve such problems approximately.

Symbolic methods

- Theoretical insight in physics come in the form of symbolic expressions. Naturally, combining machine learning and symbolic expressions is an exciting direction.

- Machine learning can be used to improve **symbolic regression**, the process of finding mathematical expressions that describe data.

  - Example: 2006.11287



- Machine learning can come up with **novel proofs and novel solutions**. A large-language model can make “educated guesses” (proposed solutions) that are then verified with a systematic evaluator. e.g. https://www.nature.com/articles/s41586-023-06924-6

Unit 1: Background

Unit 1 : Background

1.1 Probability Theory Background

Sources: e.g. deeplearningbook.org

Probability theory and Machine Learning

- **Data analysis in physics (and most domains) is always probabilistic**:
  - Inherent stochasticity in the system being modeled.
  - Incomplete observability.
  - Incomplete modeling.
- **Machine learning is inherently probabilistic**, and algorithms are written down using the notation of probability theory (e.g. expectation values).
- There are various forms of **probabilistic machine learning** that we will encounter, e.g.
  - Generative models represent PDFs
  - Machine learning of PDFs with normalizing flows, in Simulation-based Inference
  - Machine learning with probabilistic weights (Bayesian Neural Networks)
  - Machine learning can speed up more traditional statistical inference techniques such as MCMC.
- We will thus **frequently need concepts from probability theory** in this course.

Random variables

- A random variable x is sampled from
  - Probability density function (PDF) $p(x)$ ← lower case
    (continous case)
  - Probability mass function (PMF) $P(x)$ ← upper case
    (discrete case)

- We often have vector valued random vars. $\vec{x}$
- For individual samples we write $x_1, x_2, x_3, \ldots$
- $x \sim P(x)$ means that x is sampled from $P(x)$
- sometimes people write X : random variable
  x : sample of it

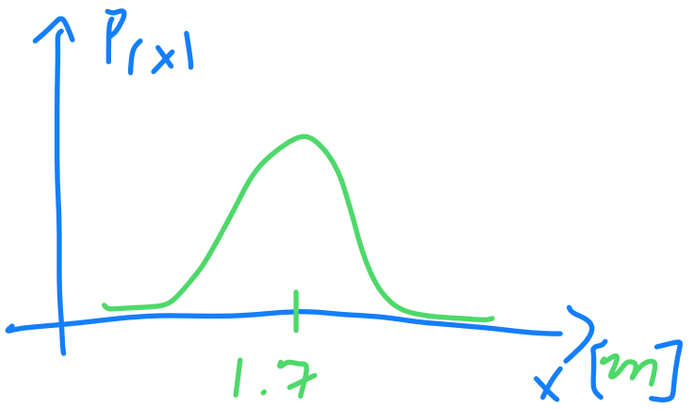
For a continuous random variable

- $p(x) \geq 0$, but $p(x)$ can be > 1
- $\int p(x)\, dx = 1$ normalization
- To get a probability of an interval

$$P_{[a,b]} = \int_a^b p(x)\, dx$$

$$0 \leq P_{ob} \leq 1$$

$P(x$ = height of a person)



For a discrete random variable

$P(x$ = Person wears glasses)

$\begin{cases} x = \text{no gl.}\ P(x) = 0.8 \\ x = \text{gl.}\ P(x) = 0.2 \end{cases}$

- $P(x) \leq 1$
- $\sum_x P(x) = 1$ normalization

Joint, conditional, marginal

- The *joint probability* of two random variables x and y is written as $P(x, y)$
- The *marginal probability* is ↑ e.g. height and weight of a person

$$p(x) = \int p(x, y)\, dy \quad \text{(continous)}$$

$$P(x) = \sum_y P(x, y) \quad \text{(discrete)}$$

- The *conditional probability* is

$$P(y|x) = \frac{P(x, y)}{P(x)}$$

„P of y given x"

Chain rule of conditional probability

given many random vars $x^{(1)}, x^{(2)} \ldots$ we have

$$P(\mathrm{x}^{(1)}, \ldots, \mathrm{x}^{(n)}) = P(\mathrm{x}^{(1)}) \Pi_{i=2}^{n} P(\mathrm{x}^{(i)} \mid \mathrm{x}^{(1)}, \ldots, \mathrm{x}^{(i-1)})$$

e.g: $P(a, b, c) = P(a \mid b, c)\, P(b \mid c)\, P(c)$

Independence of random variables

$$p(x, y) = p(x)\, p(y) \quad \forall\, x, y$$

e.g. not the case for a person's weight and height

Expectation, Variance, Covariance

The expectation value of a funct. $f(x)$ of a random variable x is given by

"expectation value" — what fct of x do we want exp. val from — $|\psi(x)|^2$ in QM

$$\mathbb{E}_{x\sim p}[f(x)] = \int p(x)\, f(x)\, dx \qquad \text{cont.}$$

x is sampled from p(x)

$$\mathbb{E}_{x\sim p}[f(x)] = \sum_x p(x)\, f(x) \qquad \text{discrete}$$

In physics we like to write

$$\mathbb{E}[f(x)] \quad \text{as} \quad \langle f(x)\rangle$$

for example the mean: $\langle x\rangle = \int p(x)\, x\, dx$

Expectation values are Linear

$$\langle \alpha f(x) + \beta g(x) \rangle = \alpha \langle f(x) \rangle + \beta \langle g(x) \rangle$$

The Variance is defined as

$$\begin{aligned} \text{Var}(f(x)) &= \langle (f(x) - \langle f(x) \rangle)^2 \rangle \\ &= \langle f^2 - 2f\langle f \rangle + \langle f \rangle^2 \rangle \\ &= \langle f^2 \rangle - 2\langle f \rangle^2 + \langle f \rangle^2 \\ &= \langle f^2 \rangle - \langle f \rangle^2 \end{aligned}$$

same with $\mathbb{E}$ notation:

$$\begin{aligned} \text{Var}(X) &= \mathrm{E}\left[(X - \mathrm{E}[X])^2\right] \\ &= \mathrm{E}\left[X^2 - 2X\,\mathrm{E}[X] + \mathrm{E}[X]^2\right] \\ &= \mathrm{E}\left[X^2\right] - 2\,\mathrm{E}[X]\,\mathrm{E}[X] + \mathrm{E}[X]^2 \\ &= \mathrm{E}\left[X^2\right] - 2\,\mathrm{E}[X]^2 + \mathrm{E}[X]^2 \\ &= \mathrm{E}\left[X^2\right] - \mathrm{E}[X]^2 \end{aligned}$$

The standard deviation is

$$\sigma = \sqrt{\text{Var}}$$

• The Covariance is

$$Cov[f(x), g(y)] = \langle (f(x) - \langle f(x) \rangle)(g(y) - \langle g(y) \rangle) \rangle$$

expectation value brackets

$f(x)$: function of random var X, e.g. height

$g(y)$: function of random var y, e.g. weight

• For a random vector $\vec{x}$ we have the

Covariance matrix $Cov(\vec{x})_{ij} = Cov(x_i, x_j)$

• and the Correlation matrix is

$$Corr(\vec{x})_{ij} = \frac{Cov(\vec{x})_{ij}}{\sqrt{Cov(\vec{x})_{ii}\, Cov(\vec{x})_{jj}}}$$

$\in (-1, 1)$

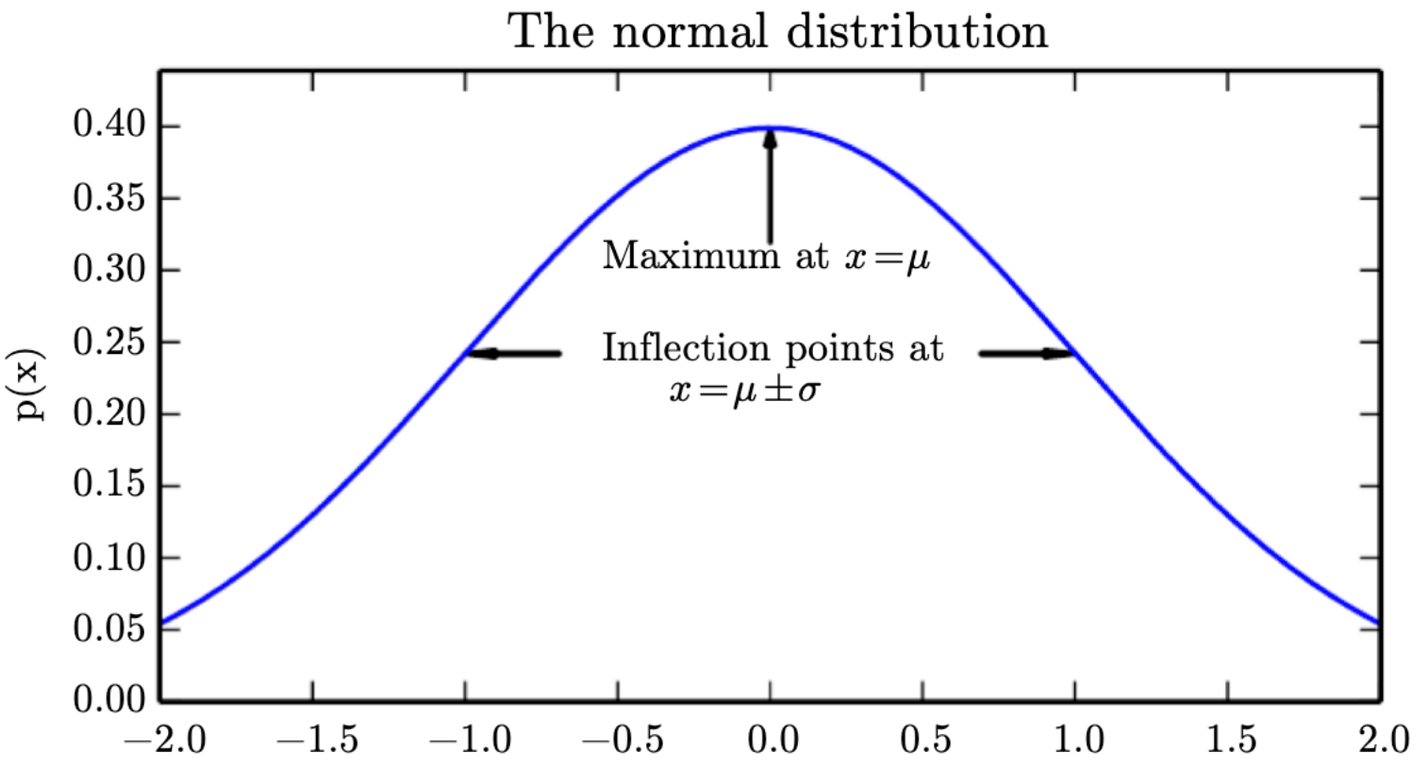
e.g. $Corr(\text{height}, \text{weight}) \sim 0.7$

Gaussian Distribution = „normal distribution"

in 1 dim.: $\mathcal{N}(x; \mu, \sigma^2) = \sqrt{\frac{1}{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right)$

mean

σ^2: variance
σ: standard dev.



deeplearningbook.org

in N dim.: $\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sqrt{\frac{1}{(2\pi)^n \det(\boldsymbol{\Sigma})}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right)$

covariance matrix

Bayes theorem

$$P(x,y) = P(y,x)$$

$$P(x|y)\,P(y) = P(y|x)\,P(x)$$

$$\Rightarrow \quad \boxed{P(x|y) = \frac{P(y|x)\,P(x)}{P(y)}}$$

Bayes theorem

In particular given $P(x|y)$ we can calculate $P(y|x)$ and vice versa.

Change of variables

If we have a random vector $\vec{x}$ and define a new random vector $\vec{y}$ by

$$\vec{y} = g(\vec{x})$$

↖ deterministic, invertible, continuous, differentiable

In 1 dim:

$$p_y(y) = p_x(g^{-1}(y)) \left| \frac{\partial x}{\partial y} \right| \qquad p_x(x) = p_y(g(x)) \left| \frac{\partial g(x)}{\partial x} \right|$$

In n-dim:

$$p_x(\boldsymbol{x}) = p_y(g(\boldsymbol{x})) \left| \det \left(\frac{\partial g(\boldsymbol{x})}{\partial \boldsymbol{x}} \right) \right|$$

Jacobian determinant

Unit 1 : Background

1.2 Classical Statistics and Data Analysis Background

Sources:
- Cowan - Statistical data analysis
- also mostly covered in deeplearningbook.org

Estimators

- A (point) estimator makes a prediction for some quantity of interest λ.
  We write $\hat{\lambda}$, where "hat" means estimator.

  E.g. estimator of the mean height $\bar{h}$

  $$\hat{\bar{h}} = \frac{1}{N} \sum_i h_i$$

- Given a dataset $\{x^{obs}\}$ drawn from/sampled a PDF $P(x)$, an estimator is some function

  $$\hat{\lambda} = \mathcal{E}[\{x^{obs}\}]$$

  ↖ some function, "guessed" or derived

- An optimal estimator is
  - unbiased: $\langle \hat{\lambda} \rangle = \lambda$ "correct on average"
  - minimum variance: $\mathrm{Var}[\hat{\lambda}]$ is as small as possible (smallest error)

Some common estimators

notation: $\bar{X}$
"bar" means mean

(sample) mean: $\hat{\bar{x}} = \frac{1}{n} \sum_{i=1}^{n} x_i^{obs}$

variance: $\hat{V} = \frac{1}{n-1} \sum_{n=1}^{n} (x_1 - \hat{\bar{x}})^2$

covariance: $\hat{Cov} = \frac{1}{n-1} \sum_{i=1}^{n} (x_1 - \hat{\bar{x}})(y_i - \hat{\bar{y}})$

These estimators have a variance, e.g.

$V[\hat{\bar{x}}] = \frac{\sigma^2}{n}$ where σ^2 is the variance of x

Likelihood, Posterior, Prior

- The Likelihood is the probability of measuring data $\vec{d}$ given a model M with parameters $\vec{\lambda}$.

$$\mathcal{L}(\vec{d} \mid M, \vec{\lambda})$$

often not explicitly written

often used as the Loss function in machine Learning.

- The Posterior is the probability of parameters $\vec{\lambda}$ given observed data $\vec{d}$.

$$P(\vec{\lambda}, M \mid \vec{d})$$

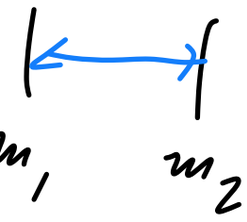
e.g M: standard model
$\vec{\lambda}$: Higgs mass
$\vec{d}$: collision data

- We can use Bayes theorem to get P from L

$$P(\vec{\lambda} \mid \vec{d}) = \frac{\mathcal{L}(\vec{d} \mid \vec{\lambda})\, P(\vec{\lambda})}{P(\vec{d})}$$

- The prior $P(\vec{\lambda})$ is the probability of $\vec{\lambda}$ before we perform the measurement.

E.g. • flat in some interval



• Gaussian around some prior measurement.

Prior is important if the data is not very constraining.

- Finally we have the evidence

$P(\vec{d})$ is the probability of the data under model M for ANY parameters $\vec{\lambda}$.

$$P(\vec{d}) = \int \mathcal{L}(d|\vec{\lambda})\, P(\vec{\lambda})\, d\vec{\lambda}$$

Often used in model comparison

Course logistics

- **Reading for this lecture:**
  - **For example:** Deeplearningbook.org chapter 3, parts of chapter 5.

- **Problem set**: No problem set in the first week