Representing Uncertainty in ML

LxMLS – 2023

Wilker Aziz
w.aziz@uva.nl

University of Amsterdam
Institute for Logic, Language and Computation

UTTER

Hello

I am an assistant professor at the Institute for Logic, Language and Computation (University of Amsterdam). You can check some of my work here https://probabll.github.io

Stuff I typically work on include

- machine learning
  approximate (Bayesian) inference, gradient estimation, normalising flows, latent variables models (e.g., VAEs),
- natural language processing
  translation, text classification, question answering, transparent and interpretable models

I teach advanced topics in DL (e.g., deep generative models, approximate inference) and NLP at UvA.

Some of my classes (in collaboration with UvA colleagues) can be found at https://uvadl2c.github.io and https://probabll.github.io/teaching/.

Outline

Dictionary definitions

Dictionary definitions

1. the quality of being indeterminate as to magnitude or value; the amount of variation in a numerical result that is consistent with observation;

1. Too specific. We can be uncertain about non-numerical things.

Dictionary definitions

1. the quality of being indeterminate as to magnitude or value; the amount of variation in a numerical result that is consistent with observation;
2. the state of not being definitely known or perfectly clear;

1. Too specific. We can be uncertain about non-numerical things.
2. Interesting: uncertainty as a property of the other (what we observe, contemplate or interact with). But you and I may be differently uncertain about the same thing (e.g., I don't know much about finance, you might).

Dictionary definitions

1. the quality of being indeterminate as to magnitude or value; the amount of variation in a numerical result that is consistent with observation;
2. the state of not being definitely known or perfectly clear;
3. the state or character of being uncertain in mind; a state of doubt

1. Too specific. We can be uncertain about non-numerical things.
2. Interesting: uncertainty as a property of the other (what we observe, contemplate or interact with). But you and I may be differently uncertain about the same thing (e.g., I don't know much about finance, you might).
3. My favourite: uncertainty as a property of the self (i.e., I am uncertain about stuff, so are you, on occasion we might agree but, generally, my uncertainty about stuff owes little to yours).

Dictionary definitions

1. the quality of being indeterminate as to magnitude or value; the amount of variation in a numerical result that is consistent with observation;
2. the state of not being definitely known or perfectly clear;
3. the state or character of being uncertain in mind; a state of doubt

Two views

Event-centric uncertainty concerns stochasticity inherent to that which we contemplate or interact with.

Agent-centric uncertainty concerns our own state of knowledge about what we contemplate or interact with (independently of those being stochastic themselves).

The agent-centric view can be argued to generalise the event-centric view ($\approx$ agents have access to the exact same information and agree to use it in the same way). (De Finetti, 1974; Lindley, 2013)

You and I observe a sequence of coin flips, we each represent our uncertainty about the next flip.

- Event-centric approach says our representations must be identical, else one or both of us is being irrational.
- Agent-centric approach says our representations need not be identical (e.g., we might possess different information about the flips and the physics of coins).

You and I read the first half of the lord of the rings, our uncertainty about the finale need not be the same.

- Event-centric: struggles with the fact that the finale isn't stochastic (we cannot have the author rewrite it).
- Agent-centric: one can always express their own uncertainty over something they lack information about.

Formal account

A representation of the state of knowledge of an agent. Most theories are built on two key frameworks:

- Possible worlds (Hintikka, 1957, 1961; Menzel, 2023): a set algebra used to represent knowledge and possibility.
- Plausibility measures (Friedman and Halpern, 1996): a tool to order propositions as to express a preference for those we are less uncertain about.

Book recommendation

- Overview of formal frameworks: Halpern (2017)

Possible worlds

Possible worlds

- You contemplate or interact with the world, but you do not know the actual state of the world.

- For example, if you are contemplating the result of rolling a six-sided die, you may represent a world (an outcome of the experiment) as a symbol w_k where k is the number of pips the die shows.

Possible worlds

- You contemplate or interact with the world, but you do not know the actual state of the world.
- You represent a world as a symbol (or collection of attributes) and you assume the actual world ω is one of a set Ω of possible worlds.

- For example, if you are contemplating the result of rolling a six-sided die, you may represent a world (an outcome of the experiment) as a symbol w_k where k is the number of pips the die shows.
- Assuming dice always land on exactly one of their 6 faces (never on an edge), you take $\Omega = \{w_1, w_2, w_3, w_4, w_5, w_6\}$ to represent all possible worlds.

Possible worlds

- You contemplate or interact with the world, but you do not know the actual state of the world.
- You represent a world as a symbol (or collection of attributes) and you assume the actual world ω is one of a set Ω of possible worlds.
- You use subsets of Ω to convey knowledge and possibility (or propositions).

- For example, if you are contemplating the result of rolling a six-sided die, you may represent a world (an outcome of the experiment) as a symbol w_k where k is the number of pips the die shows.
- Assuming dice always land on exactly one of their 6 faces (never on an edge), you take $\Omega = \{w_1, w_2, w_3, w_4, w_5, w_6\}$ to represent all possible worlds.
- If you speculate the actual world ω is an odd number you make the proposition that ω is in $A_{\text{odd}} = \{w \in \Omega : \mathrm{odd}(w)\} = \{w_1, w_3, w_5\}$.

Possible worlds

- You contemplate or interact with the world, but you do not know the actual state of the world.
- You represent a world as a symbol (or collection of attributes) and you assume the actual world ω is one of a set Ω of possible worlds.
- You use subsets of Ω to convey knowledge and possibility (or propositions).
- The set Σ of subsets of Ω you acknowledge as possible form a σ-algebra (i.e., closed under complementation and countable union/intersection).

- For example, if you are contemplating the result of rolling a six-sided die, you may represent a world (an outcome of the experiment) as a symbol w_k where k is the number of pips the die shows.
- Assuming dice always land on exactly one of their 6 faces (never on an edge), you take $\Omega = \{w_1, w_2, w_3, w_4, w_5, w_6\}$ to represent all possible worlds.
- If you speculate the actual world ω is an odd number you make the proposition that ω is in $A_{\text{odd}} = \{w \in \Omega : \mathrm{odd}(w)\} = \{w_1, w_3, w_5\}$.
- If you know the actual world ω is odd or prime and not one that can be decomposed as a sum of two other distinct worlds, then you know ω is in: $(\underbrace{\{1,3,5\}}_{A_{\text{odd}}} \cup \underbrace{\{2,3,5\}}_{A_{\text{prime}}}) \cap \underbrace{(\Omega \setminus \{3,5\})}_{A_{\text{not-sum-of-two-worlds}}} = \{1,2\}$

Possible worlds

- You contemplate or interact with the world, but you do not know the actual state of the world.
- You represent a world as a symbol (or collection of attributes) and you assume the actual world ω is one of a set Ω of possible worlds.
- You use subsets of Ω to convey knowledge and possibility (or propositions).
- The set Σ of subsets of Ω you acknowledge as possible form a σ-algebra (i.e., closed under complementation and countable union/intersection).

Plausibility measure

Plausibility measure

- Your uncertainty is qualitatively different for different propositions.

- You are at least as uncertain about $A_{\text{odd,prime}} = A_{\text{odd}} \cap A_{\text{prime}}$ as you are about A_{odd}, simply because the later includes the former.

Plausibility measure

- Your uncertainty is qualitatively different for different propositions.
- You attach an uncertainty qualifier to each proposition in Σ. A qualifier is anything you can order (e.g., fractions of a stick, numbers).

- You are at least as uncertain about $A_{\text{odd,prime}} = A_{\text{odd}} \cap A_{\text{prime}}$ as you are about A_{odd}, simply because the later includes the former.
- You may attach $2/6$ to $A_{\text{odd,prime}}$ and $3/6$ to A_{odd} and to A_{prime}. You may have information about prime numbers being particularly rare (e.g., your dice was built that way), then you choose a qualifier for A_{prime} that's lower than that of A_{odd}, and an even lower qualifier for $A_{\text{odd,prime}}$.

Plausibility measure

- Your uncertainty is qualitatively different for different propositions.
- You attach an uncertainty qualifier to each proposition in Σ. A qualifier is anything you can order (e.g., fractions of a stick, numbers).
- (Ω, Σ) is a *measurable space*, hence you may use a *plausibility measure* to give a more detailed representation of your ignorance about propositions.

- You are at least as uncertain about $A_{\text{odd,prime}} = A_{\text{odd}} \cap A_{\text{prime}}$ as you are about A_{odd}, simply because the later includes the former.
- You may attach $^2/_6$ to $A_{\text{odd,prime}}$ and $^3/_6$ to A_{odd} and to A_{prime}. You may have information about prime numbers being particularly rare (e.g., your dice was built that way), then you choose a qualifier for A_{prime} that's lower than that of A_{odd}, and an even lower qualifier for $A_{\text{odd,prime}}$.
- Plausibility (Pl) generalises certain aspects of a typical measure (e.g., it may be non-numerical). You can now convey $\text{Pl}(A_{\text{odd}}) \geq \text{Pl}(A_{\text{prime}}) \geq \text{Pl}(A_{odd,prime})$, and much more.

Plausibility measure

- Your uncertainty is qualitatively different for different propositions.
- You attach an uncertainty qualifier to each proposition in Σ. A qualifier is anything you can order (e.g., fractions of a stick, numbers).
- (Ω, Σ) is a *measurable space*, hence you may use a *plausibility measure* to give a more detailed representation of your ignorance about propositions.
- Plausibility measures include belief functions (Shafer, 1976), possibility measures (Dubois and Prade, 1990), ordinal ranking functions (Goldszmidt and Pearl, 1992), (non-numerical) preference orders (Friedman and Halpern, 1996), and, of course, probability (Kolmogorov, 1960).

- You are at least as uncertain about $A_{\text{odd,prime}} = A_{\text{odd}} \cap A_{\text{prime}}$ as you are about A_{odd}, simply because the later includes the former.
- You may attach $^2/_6$ to $A_{\text{odd,prime}}$ and $^3/_6$ to A_{odd} and to A_{prime}. You may have information about prime numbers being particularly rare (e.g., your dice was built that way), then you choose a qualifier for A_{prime} that's lower than that of A_{odd}, and an even lower qualifier for $A_{\text{odd,prime}}$.
- Plausibility (Pl) generalises certain aspects of a typical measure (e.g., it may be non-numerical). You can now convey $\text{Pl}(A_{\text{odd}}) \geq \text{Pl}(A_{\text{prime}}) \geq \text{Pl}(A_{odd,prime})$, and much more.
- Under certain documented assumptions (Friedman and Halpern, 1996), they enable something like a 'calculus of uncertainty' which formalises the procedures the agent must follow to incorporate additional information about the world and revise their uncertainty representation coherently (in axiomatic probability, this is known as *conditioning*).

Plausibility measure

- Your uncertainty is qualitatively different for different propositions.
- You attach an uncertainty qualifier to each proposition in Σ. A qualifier is anything you can order (e.g., fractions of a stick, numbers).
- (Ω, Σ) is a *measurable space*, hence you may use a *plausibility measure* to give a more detailed representation of your ignorance about propositions.
- Plausibility measures include belief functions (Shafer, 1976), possibility measures (Dubois and Prade, 1990), ordinal ranking functions (Goldszmidt and Pearl, 1992), (non-numerical) preference orders (Friedman and Halpern, 1996), and, of course, probability (Kolmogorov, 1960).

Probability

The most well known plausibility measure, has been motivated from various angles, the most prominent are:

- Objectivist: a notion of long-run stable frequency of repeatable events
- Subjectivist: a personal quantification of belief, it owes nothing to sample frequency (though it may coincide with it whenever that makes sense to an agent), it is constrained only by the axioms of probability theory and not by any interpretation (as chance or frequency). (Ramsey, 1931; De Finetti, 1974)

- Objectivist view is coherent with definition (ii) *the state of not being definitely known or perfectly clear;*
- Subjectivist view is coherent with definition (iii) *the state or character of being uncertain in mind; a state of doubt*

Both views share the same formal device (i.e., probability measures). The different interpretations are relevant when discussing strategies to use data to inform our representation of uncertainty.

Historical overview of different interpretations (Hacking, 1975)

Probability measure

A function $\Pr : \Sigma \to [0, 1]$ such that

- $\Pr(\emptyset) = 0$
- $\Pr(\Omega) = 1$
- and $\underbrace{\Pr(\cup_i A_i) = \sum_i \Pr(A_i)}_{\text{additivity}}$ for pairwise disjoint events A_i

The significance of (countable) additivity is tremendous.

It can be shown (Radon–Nikodym theorem) that to identify a probability measure (i.e., over an event space Σ) it is sufficient to identify a probability density function (which assigns a non-negative density to each *outcome*, rather than event) and a base measure.

It is much easier to work with probability density functions than with probability measures directly, esp when we intend to *predict* these objects (e.g., using NNs) from available information. Pr is a function from $\Sigma \to [0, 1]$ s.t. countable additivity, a pdf is a function from $\mathbb{R}$ to $\mathbb{R}_{>0}$ whose integral converges (to 1 if properly normalised).

Statistics

Gives us procedures we can use to fix the "free parameters" of our favourite framework of uncertainty representation (e.g., the probability measure) as to be coherent with relevant knowledge and evidence.

- Frequentist statistics: deeply rooted in the objectivist interpretation; procedures are based on repeatedly sampling data and typically formulated as optimisation problems (e.g., maximum likelihood estimation).
- Bayesian statistics: deeply rooted in subjectivist interpretation; procedures are based on probability calculus thus formulated as probabilistic inference problems (e.g., conditioning, marginalisation, expectation).

In ML

We represent our uncertainty about something by identifying a probability measure over a space of propositions (events) about the variables of interest; we typically specify a family of such measures and prioritise the members that are more consistent with observational data (that's the role of statistics).

In the rest of the class, we will fit our models using a Frequentist procedure ($\approx$ MLE).

If we use Frequentist procedures (which we often do, at least in DL), we are tempted to think that our representation will comply with the objectivist interpretation of probability. Unfortunately, this isn't really the case, but that is a topic for another chat :)

Summary

- Uncertainty isn't variance, or entropy, or probability, or statistics, ...
- Uncertainty is a state of limited knowledge and it can be represented from the perspective of the phenomena under study or of the agents studying those.
- Possible worlds (a representation of what is possible) and plausibility measures (an expression of preferences) give a family of mathematical tools for uncertainty representation.
- Probability is the dominant tool in ML, partly due to its intuitive foundations (and interpretations), partly due to us having developed statistics for it, thus enabling data-driven procedures to represent uncertainty optimally (in some sense of the word).

Outline

1. Uncertainty
2. Probabilistic Models
3. Modelling Random Experiments
4. Modelling Observed Random Variables
5. Tools for prescribing distributions
   - Univariate
   - Multivariate

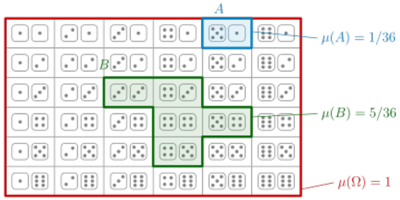
Probabilistic Models

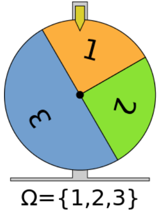
A probabilistic model *prescribes* the probability measure of a random experiment.

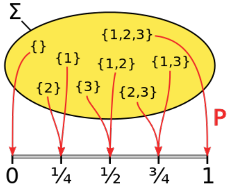
Random experiment: a sample space Ω, an event space $\Sigma = \mathcal{P}(\Omega)$, and a probability measure $\mathbb{P} : \Sigma \to [0, 1]$ where $\mathbb{P}(\emptyset) = 0$, $\mathbb{P}(\Omega) = 1$ and $\mathbb{P}(\cup_{i\in I} E_i) = \sum_{i\in I} \mathbb{P}(E_i)$ for collections $\{E_i\}$ of pairwise disjoint events.

Probabilistic Models

A probabilistic model *prescribes* the probability measure of a random experiment.







Figures from Wikimedia (CC-ASA-4.0)

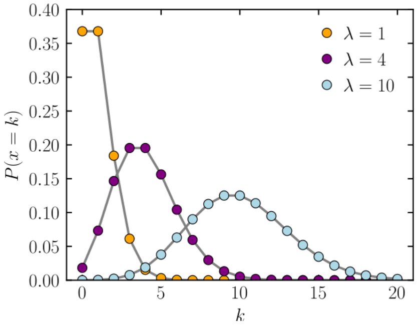
Random experiment: a sample space Ω, an event space $\Sigma = \mathcal{P}(\Omega)$, and a probability measure $\mathbb{P} : \Sigma \to [0,1]$ where $\mathbb{P}(\emptyset) = 0$, $\mathbb{P}(\Omega) = 1$ and $\mathbb{P}(\cup_{i \in I} E_i) = \sum_{i \in I} \mathbb{P}(E_i)$ for collections $\{E_i\}$ of pairwise disjoint events.

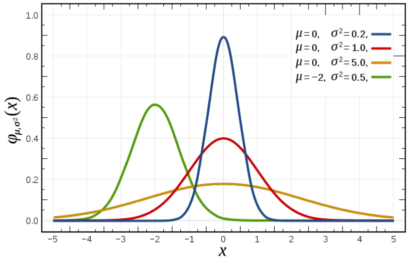
There are so many ways in which we can prescribe a probability measure:

- We may specify the probability of events in the event space, one at a time. This is at least tedious, intricate (we cannot violate countable additivity), sometimes impossible!

Probabilistic Models

A probabilistic model *prescribes* the probability measure of a random experiment.





Figures from Wikimedia (CC-ASA-4.0)

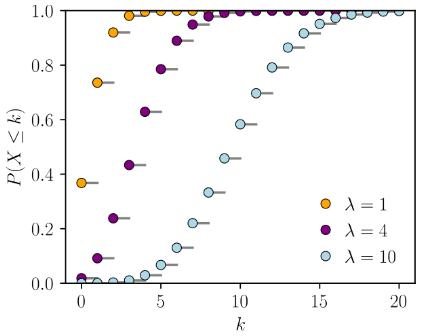
Random experiment: a sample space Ω, an event space $\Sigma = \mathcal{P}(\Omega)$, and a probability measure $\mathbb{P} : \Sigma \to [0, 1]$ where $\mathbb{P}(\emptyset) = 0$, $\mathbb{P}(\Omega) = 1$ and $\mathbb{P}(\cup_{i \in I} E_i) = \sum_{i \in I} \mathbb{P}(E_i)$ for collections $\{E_i\}$ of pairwise disjoint events.

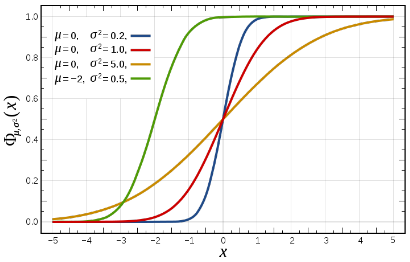
There are so many ways in which we can prescribe a probability measure:

- We may specify the probability of events in the event space, one at a time. This is at least tedious, intricate (we cannot violate countable additivity), sometimes impossible!
- We may instead specify a probability mass or density function (pmf or pdf) for outcomes of a random variable $X : \Omega \to \mathcal{X} \subseteq \mathbb{R}$.
  The rv and its pdf in turn identify a probability measure.

Probabilistic Models

A probabilistic model *prescribes* the probability measure of a random experiment.





Figures from Wikimedia (CC-ASA-4.0)

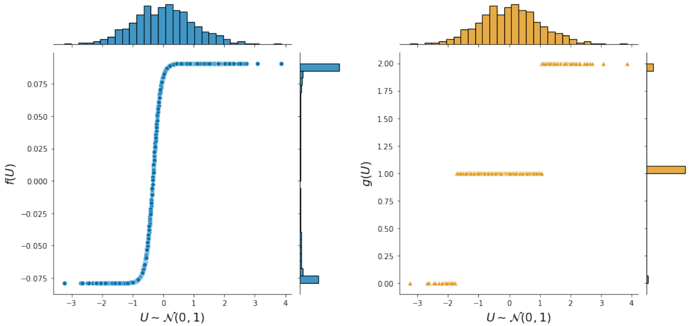
Random experiment: a sample space Ω, an event space $\Sigma = \mathcal{P}(\Omega)$, and a probability measure $\mathbb{P} : \Sigma \to [0,1]$ where $\mathbb{P}(\emptyset) = 0$, $\mathbb{P}(\Omega) = 1$ and $\mathbb{P}(\cup_{i \in I} E_i) = \sum_{i \in I} \mathbb{P}(E_i)$ for collections $\{E_i\}$ of pairwise disjoint events.

There are so many ways in which we can prescribe a probability measure:

- We may specify the probability of events in the event space, one at a time. This is at least tedious, intricate (we cannot violate countable additivity), sometimes impossible!
- We may instead specify a probability mass or density function (pmf or pdf) for outcomes of a random variable $X : \Omega \to \mathcal{X} \subseteq \mathbb{R}$. The rv and its pdf in turn identify a probability measure.
- We may instead specify the cumulative distribution function (cdf) of an rv, the cdf in turn identifies a pdf, the rv and its pdf identify a probability measure.

Probabilistic Models

A probabilistic model *prescribes* the probability measure of a random experiment.



$f(u) = 0.1\tanh(4.2u + 1.4)$ and $g(u) = \begin{cases} 0 & \text{if } f(u) < -0.5 \\ 2 & \text{if } f(u) > 0.5 \\ 1 & \text{otherwise} \end{cases}$

Figures from Wikimedia (CC-ASA-4.0)

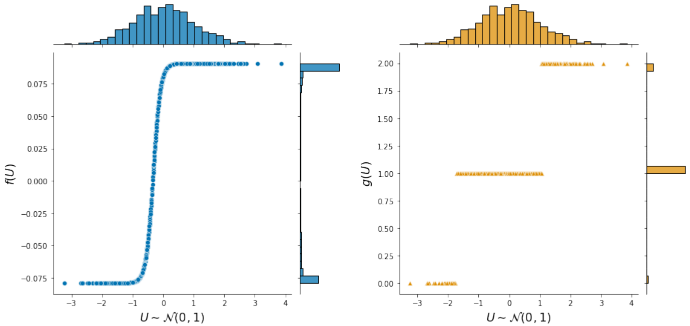
Random experiment: a sample space Ω, an event space $\Sigma = \mathcal{P}(\Omega)$, and a probability measure $\mathbb{P} : \Sigma \to [0,1]$ where $\mathbb{P}(\emptyset) = 0$, $\mathbb{P}(\Omega) = 1$ and $\mathbb{P}(\cup_{i\in I} E_i) = \sum_{i\in I} \mathbb{P}(E_i)$ for collections $\{E_i\}$ of pairwise disjoint events.

There are so many ways in which we can prescribe a probability measure:

- We may specify the probability of events in the event space, one at a time. This is at least tedious, intricate (we cannot violate countable additivity), sometimes impossible!
- We may instead specify a probability mass or density function (pmf or pdf) for outcomes of a random variable $X : \Omega \to \mathcal{X} \subseteq \mathbb{R}$. The rv and its pdf in turn identify a probability measure.
- We may instead specify the cumulative distribution function (cdf) of an rv, the cdf in turn identifies a pdf, the rv and its pdf identify a probability measure.
- We may specify a simulator (e.g., a function from `rand()` to outcomes of an rv), the simulator identifies an inverse cdf, which in turn identifies a cdf, which in turn ...

Probabilistic Models

A probabilistic model *prescribes* the probability measure of a random experiment.



$$f(u) = 0.1\tanh(4.2u + 1.4) \text{ and } g(u) = \begin{cases} 0 & \text{if } f(u) < -0.5 \\ 2 & \text{if } f(u) > 0.5 \\ 1 & \text{otherwise} \end{cases}$$

Figures from Wikimedia (CC-ASA-4.0)

Random experiment: a sample space Ω, an event space $\Sigma = \mathcal{P}(\Omega)$, and a probability measure $\mathbb{P} : \Sigma \to [0,1]$ where $\mathbb{P}(\emptyset) = 0$, $\mathbb{P}(\Omega) = 1$ and $\mathbb{P}(\cup_{i\in I} E_i) = \sum_{i\in I} \mathbb{P}(E_i)$ for collections $\{E_i\}$ of pairwise disjoint events.

There are so many ways in which we can prescribe a probability measure:

- We may specify the probability of events in the event space, one at a time. This is at least tedious, intricate (we cannot violate countable additivity), sometimes impossible!
- We may instead specify a probability mass or density function (pmf or pdf) for outcomes of a random variable $X : \Omega \to \mathcal{X} \subseteq \mathbb{R}$. The rv and its pdf in turn identify a probability measure.
- We may instead specify the cumulative distribution function (cdf) of an rv, the cdf in turn identifies a pdf, the rv and its pdf identify a probability measure.
- We may specify a simulator (e.g., a function from `rand()` to outcomes of an rv), the simulator identifies an inverse cdf, which in turn identifies a cdf, which in turn ...

Probabilistic Modelling and Reasoning

Probabilistic modelling concerns the specification of a joint distribution over random variables of interest.

Probabilistic reasoning concerns fixing a subset of these random variables to some observations and inferring marginal and conditional distributions by application of probability calculus.

The latter is also known as *probabilistic inference*.

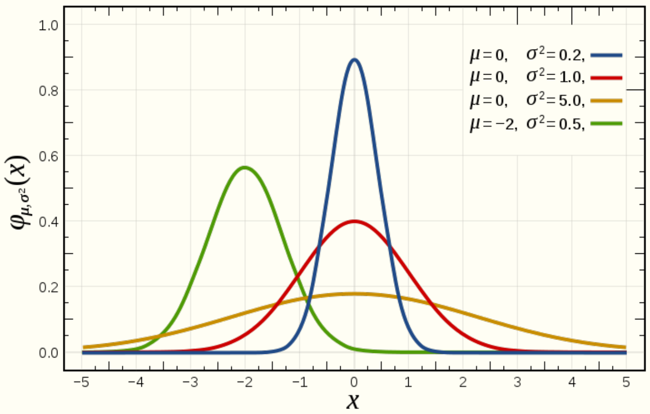
Notation. Capital letters for rvs (e.g., X, Y), lowercase letters for assignments (e.g., $X = x$, $Y = y$), calligraphic letters for range of rvs (e.g., $\mathcal{X}$, $\mathcal{Y}$). I use p_X for the pdf of X and F_X for its cdf. When needed I show the dependency of the probability density on a parameter θ as follows: $p_X(x|\theta)$.

Probability calculus recap. Chain rule $p_{XY}(x, y) = p_X(x)p_{Y|X}(y|x) = p_Y(y)p_{X|Y}(x|y)$. Conditional probability $p_{Y|X}(y|x) = \frac{p_{XY}(x,y)}{p_X(x)}$. Marginalisation $p_X(x) = \int_{\mathcal{Y}} p_{XY}(x, y)\,\mathrm{d}y$.

If you would like to learn *all* about probabilistic graphical models (PGMs), check the excellent book by Koller and Friedman (2009). I'd recommend Part I (on representation of distributions) to *anyone*.

Learning

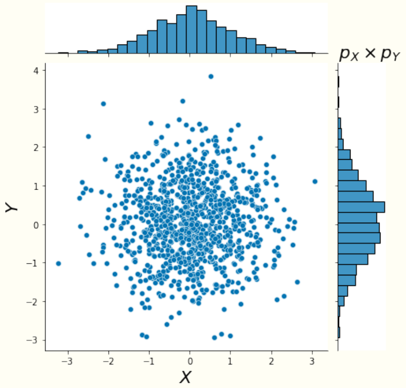
Oftentimes, we begin specifying a probability distribution by choosing a class of distributions (e.g., Normal, Exponential, Categorical).



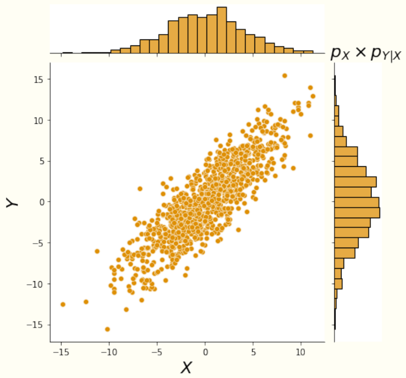
Learning

Oftentimes, we begin specifying a probability distribution by choosing a class of distributions (e.g., Normal, Exponential, Categorical).

For multivariate data, we may need to choose a factorisation of the joint distribution.



$\mathcal{N}(x|u_1, s_1^2)\mathcal{N}(y|u_2, s_2^2)$



$\mathcal{N}(x|u_1, s_1^2)\mathcal{N}(y|\mu(x), \sigma(x)^2)$

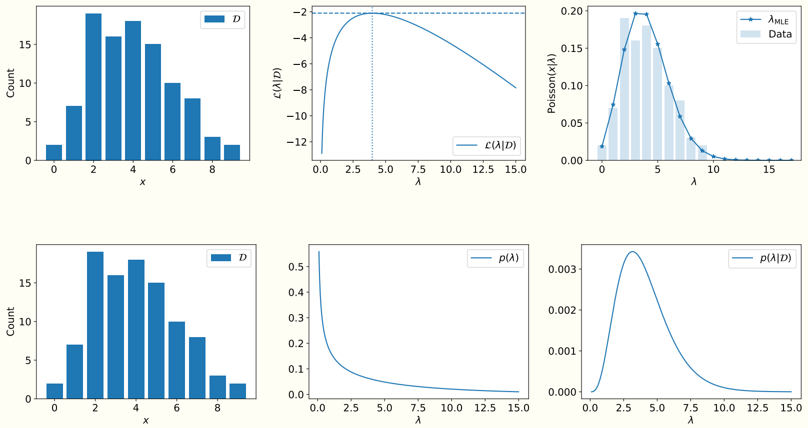
Learning

Oftentimes, we begin specifying a probability distribution by choosing a class of distributions (e.g., Normal, Exponential, Categorical).

For multivariate data, we may need to choose a factorisation of the joint distribution.

Model fitting

- Maximum likelihood estimation (MLE) singles out a member of the class (e.g., $\mathcal{N}(2,1)$, $\mathrm{Exponential}(10)$, $\mathrm{Cat}(0.1, 0.2, 0.7)$).
- Bayesian estimation conditions on available evidence (data and model assumptions) to update prior beliefs (via probability calculus).



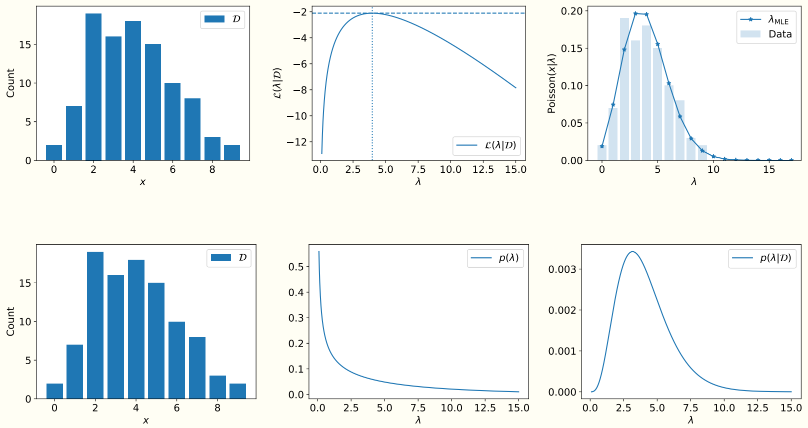
Learning

Oftentimes, we begin specifying a probability distribution by choosing a class of distributions (e.g., Normal, Exponential, Categorical).

For multivariate data, we may need to choose a factorisation of the joint distribution.

Model fitting

- Maximum likelihood estimation (MLE) singles out a member of the class (e.g., $\mathcal{N}(2,1)$, $\mathrm{Exponential}(10)$, $\mathrm{Cat}(0.1, 0.2, 0.7)$).
- Bayesian estimation conditions on available evidence (data and model assumptions) to update prior beliefs (via probability calculus).



Reasons for appreciating probabilistic models

Probabilistic models allows to incorporate assumptions through

- the choice of distribution
- dependencies among random variables
- the way that distributions uses side information
- stipulate unobserved data and their properties

They return a distribution over outcomes which can be used to

- generate data
- account for unobserved data
- provide explanation and suggest improvements
- inform decision makers

Outline

Modelling random experiments

We treat *data* as outcomes of experiments involving random variables.

A *model* of the data prescribes a distribution for those random variables.
Ideally, one that is faithful to statistical properties of our observations.
Some applications:

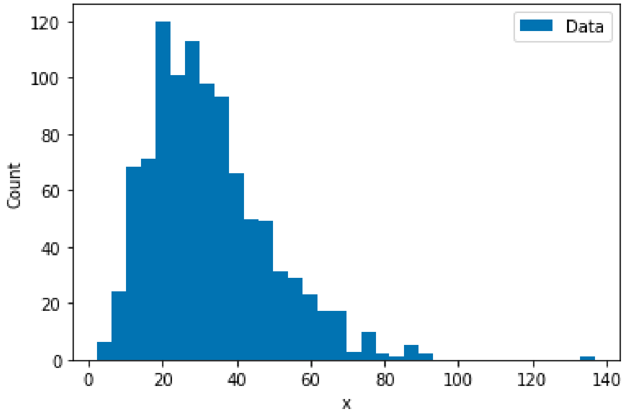
- reveal structure hidden in existing data;
- support decisions about existing and future data.

The main subject of statistical interest is data (as opposed to tasks).
Think of a task as a potential application of a (good) model of the data.

Modelling data does not imply solving a predictive task.

For example, a generative classifier is built upon a joint pdf $p_Y(y)p_{X|Y}(x|y)$ over labels $y \in \mathcal{Y}$ and inputs $x \in \mathcal{X}$. Making a specific prediction for a novel input x_* is a decision problem, oftentimes handled independently of model specification and learning. A common decision rule for classification is $y_* = \arg\max_y p_{Y|X}(y|x_*)$.

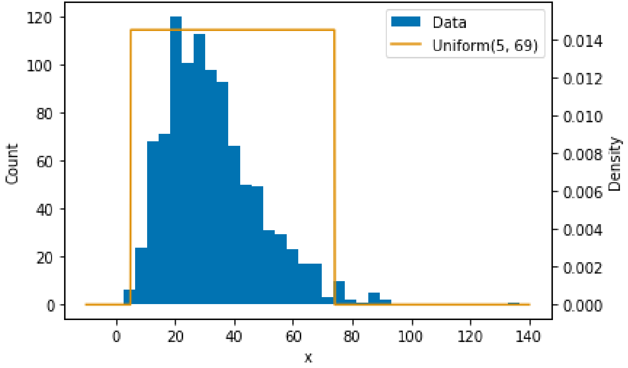
Faithfulness



Consider the data in the example

- the measurements are continuous and positive
- the sample mean is close to 32
- the sample stddev is close to 16
- they concentrate around a single value (unimodal)
- they stretch to the right (skew)

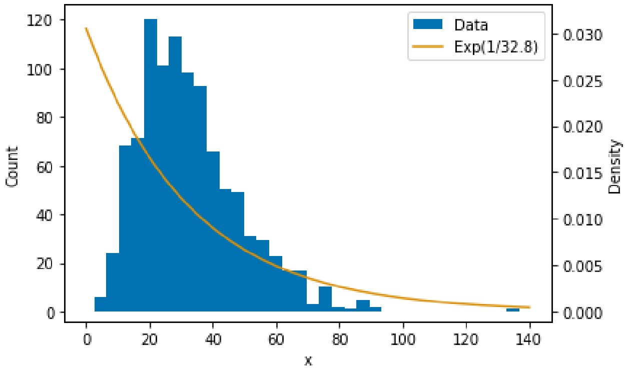
Faithfulness



Consider the data in the example

- the measurements are continuous and positive
- the sample mean is close to 32
- the sample stddev is close to 16
- they concentrate around a single value (unimodal)
- they stretch to the right (skew)

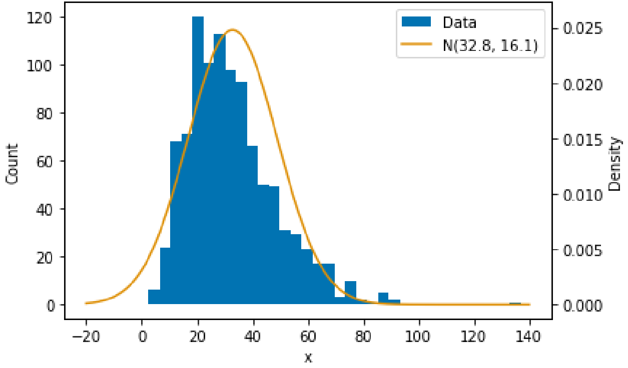
Faithfulness



Consider the data in the example

- the measurements are continuous and positive
- the sample mean is close to 32
- the sample stddev is close to 16
- they concentrate around a single value (unimodal)
- they stretch to the right (skew)

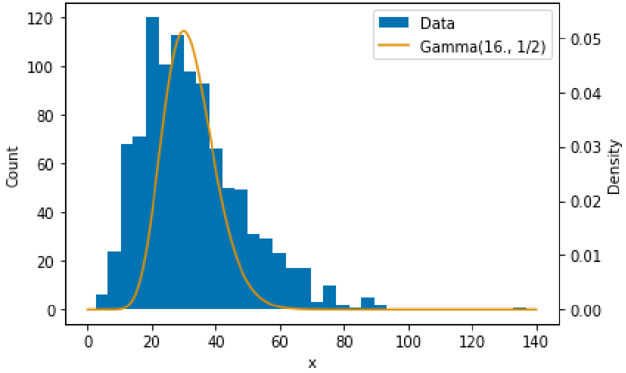
Faithfulness



Consider the data in the example

- the measurements are continuous and positive
- the sample mean is close to 32
- the sample stddev is close to 16
- they concentrate around a single value (unimodal)
- they stretch to the right (skew)

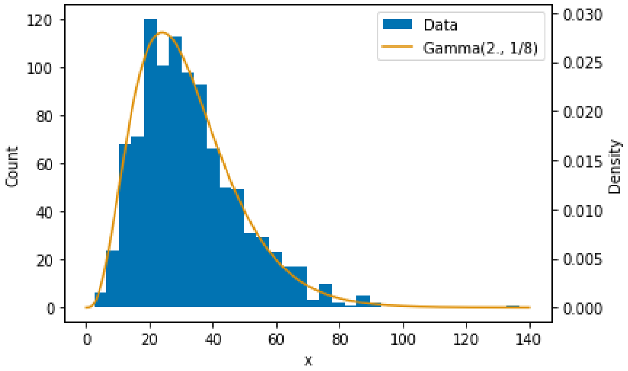
Faithfulness



Consider the data in the example

- the measurements are continuous and positive
- the sample mean is close to 32
- the sample stddev is close to 16
- they concentrate around a single value (unimodal)
- they stretch to the right (skew)

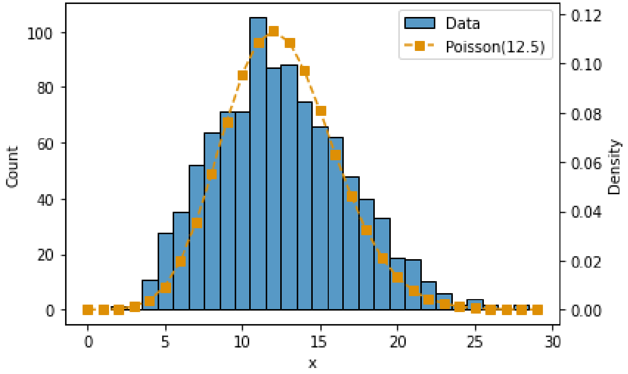
Faithfulness



Consider the data in the example

- the measurements are continuous and positive
- the sample mean is close to 32
- the sample stddev is close to 16
- they concentrate around a single value (unimodal)
- they stretch to the right (skew)

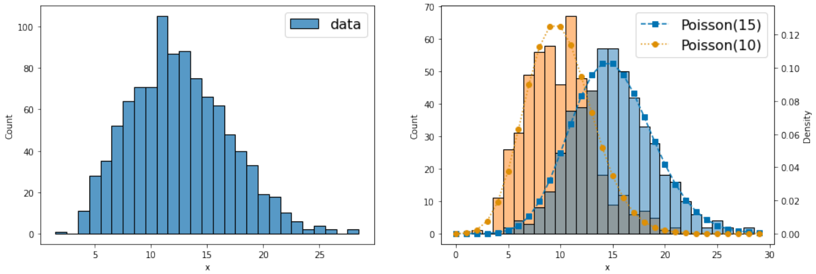
Hidden structure



Here the measurements are natural numbers, the sample mean is close to 12.5 and the median is 12.

A Poisson distribution can capture the mean, but not the spread (recall that the Poisson mean and variance are equal).

Hidden structure

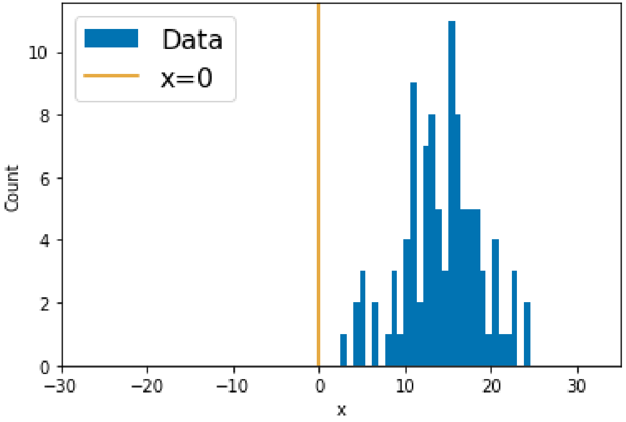


A different probabilistic model may posit the presence of two groups mixed in a single population.

Here the measurements are natural numbers, the sample mean is close to 12.5 and the median is 12.

A Poisson distribution can capture the mean, but not the spread (recall that the Poisson mean and variance are equal).

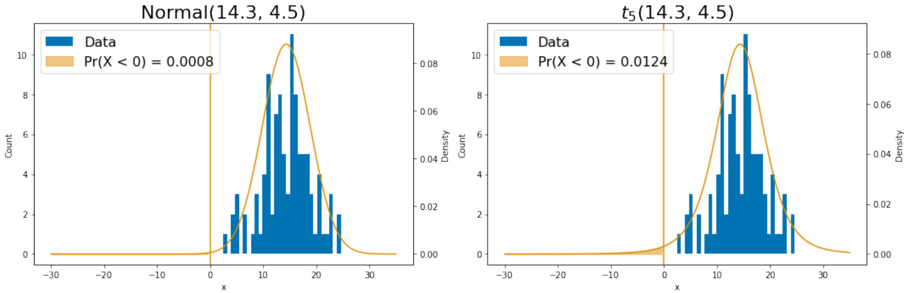
Decisions about future data



Here we observe continuous measurements from a sensor in a car. Data come in in batches of 100 measurements.

Suppose that if 1% (or more) of the readings drop below 0, the driver is at risk.

Decisions about future data



The heavier tails of the Student's t reserve much more probability for unseen data.

Here we observe continuous measurements from a sensor in a car. Data come in in batches of 100 measurements.

Suppose that if 1% (or more) of the readings drop below 0, the driver is at risk.

Outline

Modelling observed random variables

Our goal is to learn a distribution over a set of **observed** random variables.

Observed random variables are the result of random experiments that have already happened: e.g., sentences in a collection of news articles, number of stars in a product review.

Modelling observed random variables

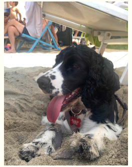
Our goal is to learn a distribution over a set of **observed** random variables.

Observed random variables are the result of random experiments that have already happened: e.g., sentences in a collection of news articles, number of stars in a product review.

Typical use in ML: *conditional models.*

▷ *We are given some variables (inputs) and we are interested in making predictions about other variables (outputs)*

- such inputs are also called *predictors* (or *covariates*)
- with some probability, *predicted by the model*, an output takes on a certain *outcome* in a sample space

Predictor	**Outcome**	**Sample space**
Why did they bother recording this???	$\star$	$\{\star, \star\star, \star\star\star, \star\star\star\star, \star\star\star\star\star\}$
Source: geen standaard MT: no standard	compare('no step')=0.5	$[0, 1]$
he proposed a famous solution to an inverse probability problem in the 18th century	https://en.wikipedia.org/wiki/Thomas_Bayes	$\mathcal{W}_{\text{en}}$
	*Pepper loves the beach!*	Σ^*_{en}
That's not possible!	*Dat is niet mogelijk!*	Σ^*_{nl}

Modelling observed random variables

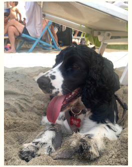
Our goal is to learn a distribution over a set of **observed** random variables.

Observed random variables are the result of random experiments that have already happened: e.g., sentences in a collection of news articles, number of stars in a product review.

Typical use in ML: *conditional models.*

▷ *We are given some variables (inputs) and we are interested in making predictions about other variables (outputs)*

- such inputs are also called *predictors* (or *covariates*)
- with some probability, *predicted by the model*, an output takes on a certain *outcome* in a sample space

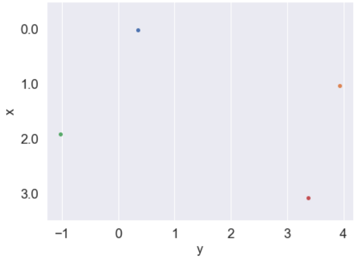
Predictor	**Outcome**	**Sample space**
Why did they bother recording this???	$\star$	$\{\star, \star\star, \star\star\star, \star\star\star\star, \star\star\star\star\star\}$
Source: geen standaard MT: no standard	compare('no step')=0.5	$[0, 1]$
he proposed a famous solution to an inverse probability problem in the 18th century	https://en.wikipedia.org/wiki/Thomas_Bayes	$\mathcal{W}_{\text{en}}$
	*Pepper loves the beach!*	Σ^*_{en}
That's not possible!	*Dat is niet mogelijk!*	Σ^*_{nl}

Choosing a model family

1. data type: countable (binary, categorical, ordinal), uncountable, univariate or multivariate, combinatorial, etc.
2. match properties of the data and distribution: overdispersion, skewness, heavy tails

Choosing a model family

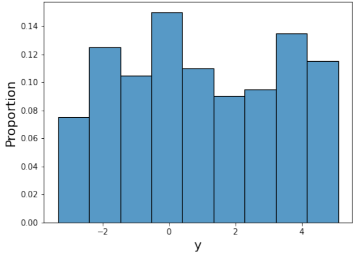
1. data type: countable (binary, categorical, ordinal), uncountable, univariate or multivariate, combinatorial, etc.
2. match properties of the data and distribution: overdispersion, skewness, heavy tails

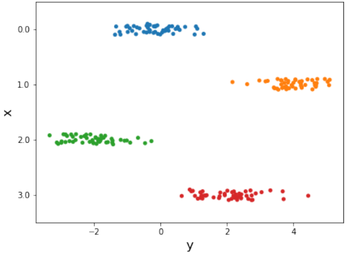


- Don't be mislead by data sparsity, collect more measurements and you will observe uncertainty (Plank, 2022).

Choosing a model family

1. data type: countable (binary, categorical, ordinal), uncountable, univariate or multivariate, combinatorial, etc.
2. match properties of the data and distribution: overdispersion, skewness, heavy tails

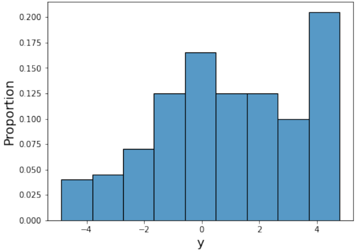


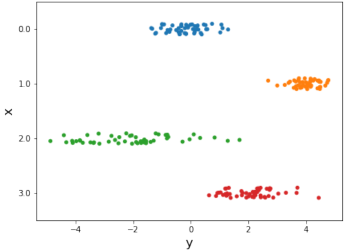


- Don't be mislead by data sparsity, collect more measurements and you will observe uncertainty (Plank, 2022).
- Don't be mislead by the marginal if you intend to model conditionally

Choosing a model family

1. data type: countable (binary, categorical, ordinal), uncountable, univariate or multivariate, combinatorial, etc.
2. match properties of the data and distribution: overdispersion, skewness, heavy tails

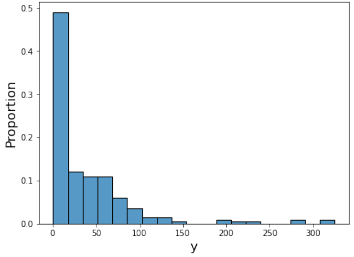


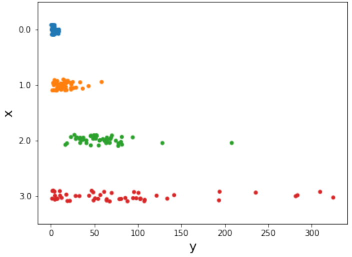


- Don't be mislead by data sparsity, collect more measurements and you will observe uncertainty (Plank, 2022).
- Don't be mislead by the marginal if you intend to model conditionally
- Do you expect variance to differ?

Choosing a model family

1. data type: countable (binary, categorical, ordinal), uncountable, univariate or multivariate, combinatorial, etc.
2. match properties of the data and distribution: overdispersion, skewness, heavy tails

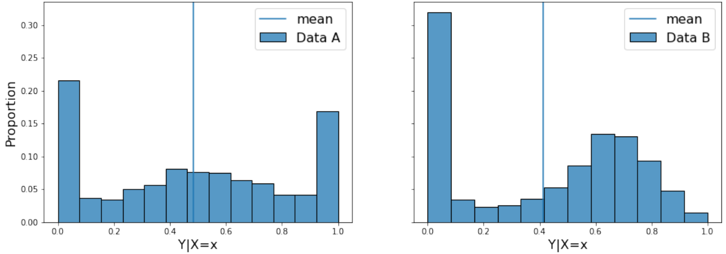




- Don't be mislead by data sparsity, collect more measurements and you will observe uncertainty (Plank, 2022).
- Don't be mislead by the marginal if you intend to model conditionally
- Do you expect variance to differ?
- Do you expect skew?

Choosing a model family

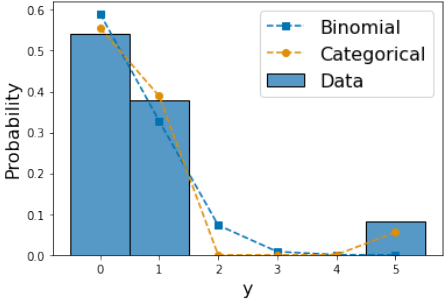
1. data type: countable (binary, categorical, ordinal), uncountable, univariate or multivariate, combinatorial, etc.
2. match properties of the data and distribution: overdispersion, skewness, heavy tails



- Don't be mislead by data sparsity, collect more measurements and you will observe uncertainty (Plank, 2022).
- Don't be mislead by the marginal if you intend to model conditionally
- Do you expect variance to differ?
- Do you expect skew?
- Bounded support? Multimodality? Asymmetry?

Choosing a model family

1. data type: countable (binary, categorical, ordinal), uncountable, univariate or multivariate, combinatorial, etc.
2. match properties of the data and distribution: overdispersion, skewness, heavy tails



- Don't be mislead by data sparsity, collect more measurements and you will observe uncertainty (Plank, 2022).
- Don't be mislead by the marginal if you intend to model conditionally
- Do you expect variance to differ?
- Do you expect skew?
- Bounded support? Multimodality? Asymmetry?
- Unsure about categorical or ordinal treatment?

Choosing a model family

1. data type: countable (binary, categorical, ordinal), uncountable, univariate or multivariate, combinatorial, etc.
2. match properties of the data and distribution: overdispersion, skewness, heavy tails

- Don't be mislead by data sparsity, collect more measurements and you will observe uncertainty (Plank, 2022).
- Don't be mislead by the marginal if you intend to model conditionally
- Do you expect variance to differ?
- Do you expect skew?
- Bounded support? Multimodality? Asymmetry?
- Unsure about categorical or ordinal treatment?

Statistical models parameterised by NNs

Once a family of distributions is in place, we let a neural network predict a member of the family, which is does by mapping from available information (e.g., x).

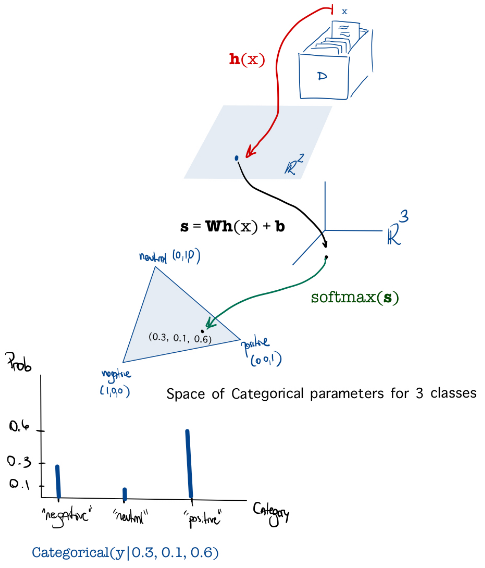
$$Y|x \sim \mathrm{Cat}(f(x;\theta)) \qquad \text{or} \qquad Y|x \sim \mathcal{N}(\mu(x;\theta), \sigma(x;\theta)^2)$$

we then proceed to estimate parameters θ of the NNs

NNs compute the parameters of the statistical model. We estimate NN parameters.

Example - Text classifier

Before DL was popular, we would identifying informative features $h(x)$ of the available predictor x. We would then map these features to the parameter of a Categorical distribution (e.g., via a log-linear model): $Y|X = x \sim \mathrm{Cat}(\mathrm{softmax}(Wh(x) + b))$.



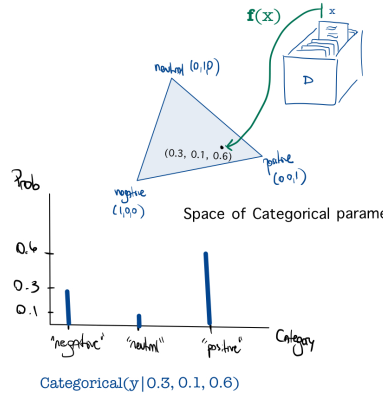
Example - Text classifier

Before DL was popular, we would identifying informative features $h(x)$ of the available predictor x. We would then map these features to the parameter of a Categorical distribution (e.g., via a log-linear model): $Y|X = x \sim \mathrm{Cat}(\mathrm{softmax}(Wh(x) + b))$.

Nowadays, we tend to condition on **everything available to us** by learning how to map from **arbitrarily complex** data to the parameters of our distributions. We do so with NNs: $Y|X = x \sim \mathrm{Cat}(f(x; \theta))$.

There's a lot of research on how to design $f(\cdot; \theta)$ and estimate θ effectively.

In $Y|X = x \sim \mathrm{Cat}(f(x; \theta))$, $f(\cdot; \theta)$ is a NN architecture with parameters θ, it maps any covariate x, say a long review in English, to the parameters of the Categorical distribution that *by assumption* govern the conditional response variable.



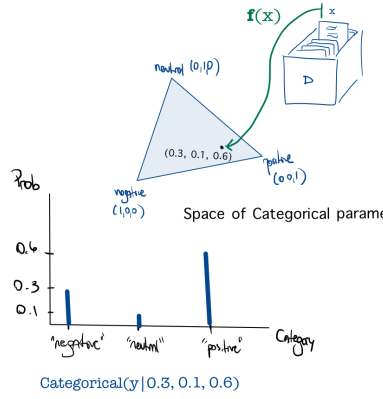
Example - Text classifier

Before DL was popular, we would identifying informative features $h(x)$ of the available predictor x. We would then map these features to the parameter of a Categorical distribution (e.g., via a log-linear model): $Y|X = x \sim \text{Cat}(\text{softmax}(Wh(x) + b))$.

Nowadays, we tend to condition on **everything available to us** by learning how to map from **arbitrarily complex** data to the parameters of our distributions. We do so with NNs: $Y|X = x \sim \text{Cat}(f(x;\theta))$.

There's a lot of research on how to design $f(\cdot;\theta)$ and estimate θ effectively.

In $Y|X = x \sim \text{Cat}(f(x;\theta))$, $f(\cdot;\theta)$ is a NN architecture with parameters θ, it maps any covariate x, say a long review in English, to the parameters of the Categorical distribution that *by assumption* govern the conditional response variable.



Maximum likelihood estimation

We have a probability model of a random variable Y, and this model may condition on available covariates X. This model has parameters θ and assigns probability mass/density $p(y|x,\theta)$ to an observation.

I may omit the subscripts from the pdfs whenever I find it unambiguous. That is, I write $p(y|x,\theta)$ instead of $p_{Y|X}(y|x,\theta)$.

Maximum likelihood estimation

We have a probability model of a random variable Y, and this model may condition on available covariates X. This model has parameters θ and assigns probability mass/density $p(y|x,\theta)$ to an observation.

Given a dataset $\mathcal{D} = \{(x^{(1)}, y^{(1)}), \ldots, (x^{(N)}, y^{(N)})\}$ of i.i.d. observations,

I may omit the subscripts from the pdfs whenever I find it unambiguous. That is, I write $p(y|x,\theta)$ instead of $p_{Y|X}(y|x,\theta)$.

Maximum likelihood estimation

We have a probability model of a random variable Y, and this model may condition on available covariates X. This model has parameters θ and assigns probability mass/density $p(y|x, \theta)$ to an observation.

Given a dataset $\mathcal{D} = \{(x^{(1)}, y^{(1)}), \ldots, (x^{(N)}, y^{(N)})\}$ of i.i.d. observations, the log-likelihood function gives us a criterion for parameter estimation

$$\mathcal{L}_{\mathcal{D}}(\theta) =$$

I may omit the subscripts from the pdfs whenever I find it unambiguous. That is, I write $p(y|x, \theta)$ instead of $p_{Y|X}(y|x, \theta)$.

Maximum likelihood estimation

We have a probability model of a random variable Y, and this model may condition on available covariates X. This model has parameters θ and assigns probability mass/density $p(y|x,\theta)$ to an observation.

Given a dataset $\mathcal{D} = \{(x^{(1)}, y^{(1)}), \ldots, (x^{(N)}, y^{(N)})\}$ of i.i.d. observations, the log-likelihood function gives us a criterion for parameter estimation

$$\mathcal{L}_{\mathcal{D}}(\theta) = \log \prod_{s=1}^{N} p(y^{(s)}|x^{(s)}, \theta) =$$

I may omit the subscripts from the pdfs whenever I find it unambiguous. That is, I write $p(y|x,\theta)$ instead of $p_{Y|X}(y|x,\theta)$.

Maximum likelihood estimation

We have a probability model of a random variable Y, and this model may condition on available covariates X. This model has parameters θ and assigns probability mass/density $p(y|x,\theta)$ to an observation.

Given a dataset $\mathcal{D} = \{(x^{(1)}, y^{(1)}), \ldots, (x^{(N)}, y^{(N)})\}$ of i.i.d. observations, the log-likelihood function gives us a criterion for parameter estimation

$$\mathcal{L}_{\mathcal{D}}(\theta) = \log \prod_{s=1}^{N} p(y^{(s)}|x^{(s)}, \theta) = \sum_{s=1}^{N} \log p(y^{(s)}|x^{(s)}, \theta)$$

I may omit the subscripts from the pdfs whenever I find it unambiguous. That is, I write $p(y|x,\theta)$ instead of $p_{Y|X}(y|x,\theta)$.

Maximum likelihood estimation

We have a probability model of a random variable Y, and this model may condition on available covariates X. This model has parameters θ and assigns probability mass/density $p(y|x,\theta)$ to an observation.

Given a dataset $\mathcal{D} = \{(x^{(1)}, y^{(1)}), \ldots, (x^{(N)}, y^{(N)})\}$ of i.i.d. observations, the log-likelihood function gives us a criterion for parameter estimation

$$\mathcal{L}_{\mathcal{D}}(\theta) = \log \prod_{s=1}^{N} p(y^{(s)}|x^{(s)}, \theta) = \sum_{s=1}^{N} \log p(y^{(s)}|x^{(s)}, \theta)$$

Stochastic gradient-based optimisation gives us a local optimum of the log-likelihood function: iterative updates to θ in the direction $\nabla_\theta \mathcal{L}_{\mathcal{D}}(\theta)$.

I may omit the subscripts from the pdfs whenever I find it unambiguous. That is, I write $p(y|x,\theta)$ instead of $p_{Y|X}(y|x,\theta)$.

Summary – a recipe for supervised learning

Maximum likelihood estimation

- tells you which loss to optimise
  (i.e. negative log-likelihood)

Automatic differentiation (*backprop*)

- "give me a tractable forward pass and I will give you gradients"

Stochastic optimisation powered by backprop

- general purpose gradient-based optimisers

Our main job is to pick an appropriate family of distributions.

Outline

Prescribing distributions

We will now discuss various ways to prescribe distributions using deep learning. For each technique, we will keep an eye on two things:

- our ability to assess the probability mass/density of a given outcome
- our ability to sample outcomes from the corresponding distribution

We begin with the univariate case and then discuss the multivariate case.

- assessment: useful for learning (e.g., via approximate MLE).
- sampling: useful for making predictions.

Outline

Overview

- enumeration
- known parametric form
- transform a known random source
- data augmentation and marginalisation

Throughout we parameterise the conditional distribution of a random variable Y given $X = x$.

We use neural network architectures which we denote with a shorthand notation:

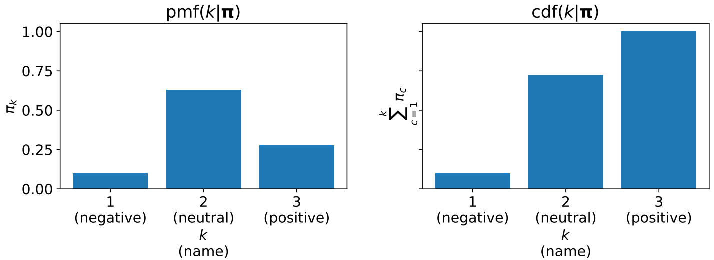
$$\text{layer}_O(\text{inputs}; \text{parameters})$$

e.g., $\text{linear}_K(\mathbf{h}; \theta_{\text{out}})$ uses parameters θ_{out} to map some input vector $\mathbf{h}$ to K outputs linearly and deterministically.

We omit the implementation details hoping the layer's name is suggestive enough (e.g., linear is implemented as $\mathbf{Wh} + \mathbf{b}$ with $\theta_{\text{out}} = \{\mathbf{W}, \mathbf{b}\}$).

Enumerate masses

Predict the probability mass of each and every one of K outcomes.



- mass assessment: evaluate K masses (e.g., an NN forward pass, t_Δ) and look the relevant one up
- sampling: linear in K (via inverse cdf or Gumbel trick)

Sentiment classifier

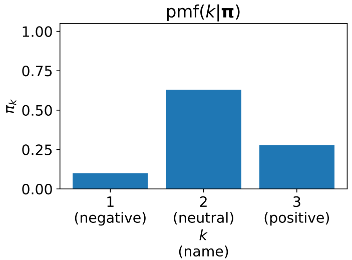
- Input: a piece of text x
- Output: a distribution over 3 possible sentiment labels (negative, neutral, positive).

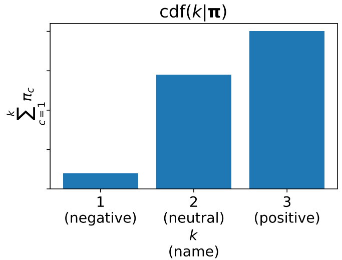
Examples

- encode the text $\mathbf{h} = \text{encode}_D(x; \theta_{\text{enc}})$
- predict K scores $\mathbf{s} = (s_1, \ldots, s_K)^\top$
  - $\mathbf{s} = \text{linear}_K(\mathbf{h}; \theta_{\text{out}})$
- map them to the probability simplex: $\boldsymbol{\pi} = \text{t}_\Delta(\mathbf{s})$
  - softmax (see (Niculae and Blondel, 2017) and (Niculae et al., 2023, §3.2 and 3.3) for an origin story)
  - or sparsemax (Martins and Astudillo, 2016)
  - or entmax$_\alpha$ (Peters et al., 2019)
  - etc.

Enumerate masses

Predict the probability mass of each and every one of K outcomes.





- mass assessment: evaluate K masses (e.g., an NN forward pass, t_Δ) and look the relevant one up
- sampling: linear in K (via inverse cdf or Gumbel trick)

How about countably infinite spaces?

Sentiment classifier

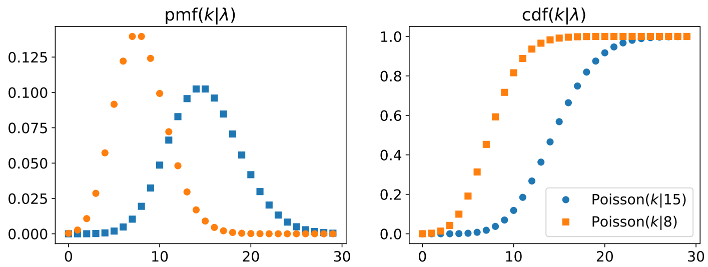
- Input: a piece of text x
- Output: a distribution over 3 possible sentiment labels (negative, neutral, positive).

Examples

- encode the text $\mathbf{h} = \text{encode}_D(x; \theta_{\text{enc}})$
- predict K scores $\mathbf{s} = (s_1, \ldots, s_K)^\top$
  - $\mathbf{s} = \text{linear}_K(\mathbf{h}; \theta_{\text{out}})$
- map them to the probability simplex: $\boldsymbol{\pi} = \text{t}_\Delta(\mathbf{s})$
  - softmax (see (Niculae and Blondel, 2017) and (Niculae et al., 2023, §3.2 and 3.3) for an origin story)
  - or sparsemax (Martins and Astudillo, 2016)
  - or entmax$_\alpha$ (Peters et al., 2019)
  - etc.

Known pmf

Predict the parameter(s) of a known pmf.



- mass assessment: evaluate the parameter(s) (e.g., NN forward pass) and the pmf (a few operations)
- sampling is typically possible (via inverse cdf method, or some specialised algorithm)

Heard count

- Input: an image $\mathbf{x} \in \mathbb{R}^{H \times W \times C}$ of a field
- Output: a Poisson distribution over the number of cows in the field. A Poisson is identified by its rate parameter (a strictly positive scalar), which we predicted given $\mathbf{x}$ as follows

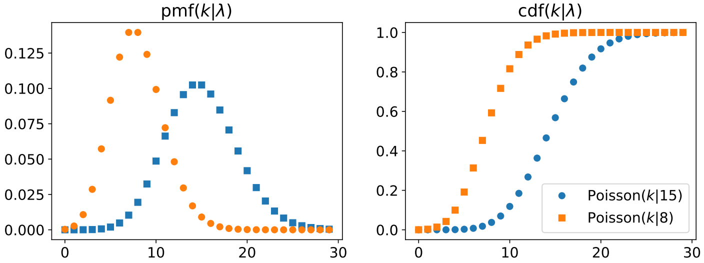
$$\begin{aligned} \mathbf{h} &= \mathrm{encode}_D(\mathbf{x}; \theta_{\text{enc}}) \\ \lambda &= \mathrm{softplus}(\mathrm{linear}_1(\mathbf{h}; \theta_{\text{rate}})) \end{aligned} \qquad (1)$$

Probability mass of an outcome: $\mathrm{Poisson}(k|\lambda) = \frac{\lambda^k \exp(-k)}{k!}$

Inverse cdf method: if $\mathrm{icdf}(\cdot|\lambda)$ is the inverse of the cdf of the rv Y, then transforming a uniform sample $p \sim \mathcal{U}(0,1)$ via $\mathrm{icdf}(p|\lambda)$ yields an rv with the same distribution as Y.

Known pmf

Predict the parameter(s) of a known pmf.



- mass assessment: evaluate the parameter(s) (e.g., NN forward pass) and the pmf (a few operations)
- sampling is typically possible (via inverse cdf method, or some specialised algorithm)

How about uncountable sample spaces?

Heard count

- Input: an image $\mathbf{x} \in \mathbb{R}^{H \times W \times C}$ of a field
- Output: a Poisson distribution over the number of cows in the field. A Poisson is identified by its rate parameter (a strictly positive scalar), which we predicted given $\mathbf{x}$ as follows

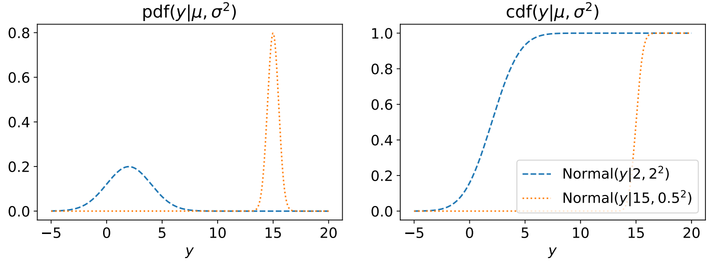
$$\begin{aligned} \mathbf{h} &= \mathrm{encode}_D(\mathbf{x}; \theta_{\text{enc}}) \\ \lambda &= \mathrm{softplus}(\mathrm{linear}_1(\mathbf{h}; \theta_{\text{rate}})) \end{aligned} \tag{2}$$

Probability mass of an outcome: $\mathrm{Poisson}(k|\lambda) = \frac{\lambda^k \exp(-k)}{k!}$

Inverse cdf method: if icdf$(\cdot|\lambda)$ is the inverse of the cdf of the rv Y, then transforming a uniform sample $p \sim \mathcal{U}(0,1)$ via icdf$(p|\lambda)$ yields an rv with the same distribution as Y.

Known pdf

Predict the parameter(s) of a known pdf.



- density assessment: evaluate the parameter(s) (e.g., NN forward pass) and the pdf (a few operations, assuming analytical form or an efficient numerical algorithm)
- sampling is typically possible (via inverse cdf method, or some specialised algorithm)

Temperature

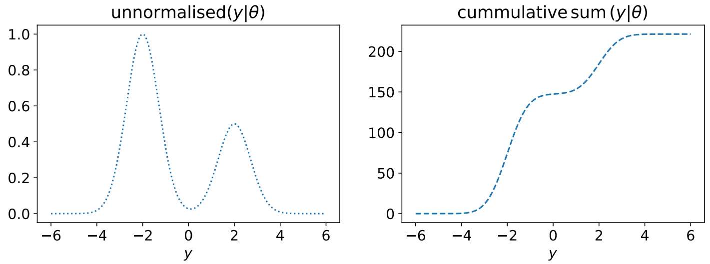
- Input: an image $\mathbf{x} \in \mathbb{R}^{H \times W \times C}$ of a car engine
- Output: a Normal distribution over temperature values in Celsius. A Normal is identified by a location in $\mathbb{R}$ and a scale in $\mathbb{R}_{>0}$, which we predict given $\mathbf{x}$ as follows

$$\begin{aligned} \mathbf{h} &= \mathrm{encode}_D(\mathbf{x}; \theta_{\text{enc}}) \\ \mu &= \mathrm{linear}_1(\mathbf{h}; \theta_{\text{loc}}) \\ \sigma &= \mathrm{softplus}(\mathrm{linear}_1(\mathbf{h}; \theta_{\text{scale}})) \end{aligned} \tag{3}$$

Probability density of an outcome: $\mathcal{N}(y|\mu, \sigma^2) = \dfrac{\exp\left(\frac{-(y-\mu)^2}{2\sigma^2}\right)}{\sigma\sqrt{2\pi}}$

Unnormalised pdf/pmf

Predict a non-negative score for a *given* outcome (as opposed to each and every outcome). This procedure must identify a pdf up to an unknown normalisation constant.



- density assessment: difficult, numerical integration is possible (but inefficient in high dimensions)
- sampling: only possible approximately (e.g., via MCMC, Langevin).

We need to output non-negative values all over the domain, and need the integral across the domain to converge to a real number (that is, finite).

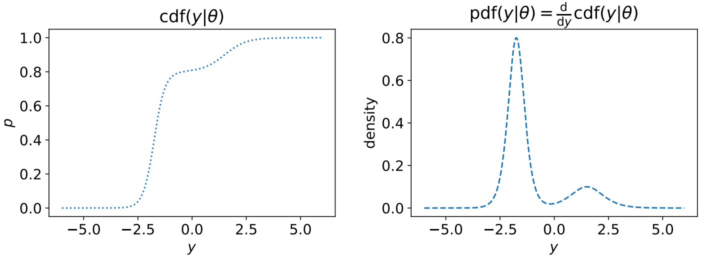
Here is an example, engineered to meet the requirements: a conic combination of exponentiated-square basis functions.

$$\begin{aligned}
\mathbf{h} &= \mathrm{encode}_D(\mathbf{x}; \theta_{\text{enc}}) \\
\mathbf{u} &= \mathrm{linear}_C(\mathbf{x}; \theta_{\text{hid}}) \\
\mathbf{v} &= \exp(-\mathbf{u} \odot \mathbf{u}) && \leftarrow C \text{ non-negative numbers} \\
s &= \mathrm{linear}_1(\mathbf{v}; \exp(\theta_{\text{out}})) && \leftarrow \text{ non-negative parameters}
\end{aligned} \tag{4}$$

For countably infinite or uncountable sample spaces, it may be difficult to guarantee that the integral of an arbitrary non-negative function converges.

CDF

Predict the cumulative probability p for an outcome y:
that is $p = \int_{a \leq y} \mathrm{pdf}(a|\theta)\mathrm{d}a$



- density assessment: requires differentiation (which can be automated!)
- sampling: difficult (requires inverting the cdf)

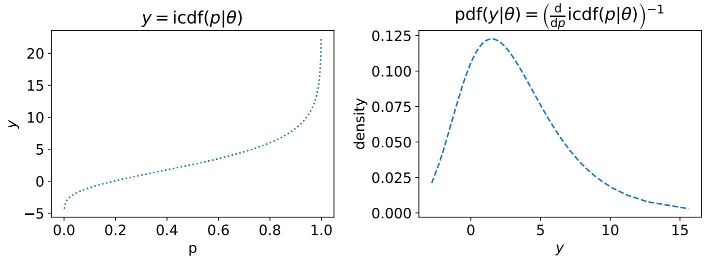
We can constrain an NN to specify a non-decreasing function by having non-negative weights in linear layers (biases are unconstrained), and non-decreasing activation functions (e.g., tanh, relu, softplus, sigmoid, etc.).

To constrain the output to $[0, 1]$ we may use a sigmoid output, or a convex combination of C sigmoid outputs.

See, for example, (Huang et al., 2018; De Cao et al., 2020)

Percentile function - inverse cdf

Predict the outcome y for a given percentile p:



- density assessment: possible in some cases (via inverse function theorem)
- sampling: easy by design

Now the output is an outcome (if $\mathcal{Y} = \mathbb{R}$, the output is unconstrained), the domain of the NN is $[0, 1]$, and the function must be non-decreasing (as the cdf case).

For y whose inverse p we know (for example, those we sampled via $p \sim \mathcal{U}(0, 1)$ and $y = \mathrm{icdf}(p|\theta)$) we can assess the density using autodiff.

Sampler - bijection

Parameterise a bijective transformation of a known and convenient base random variable

- Assessment and sampling can be made simple, in some cases only one of the two is simple;
- Parameterising one such model takes some skill (to achieve efficient computations)

Suppose access to an invertible function (a bijection): $x = h^{-1}(y)$, then

$$p_Y(y) = p_X(h^{-1}(y))|\det J_{h^{-1}}(y)| \tag{5}$$

Start from a known p_X, for example a Gaussian, and obtain a novel p_Y, more complex than a Gaussian.

It's possible to assess the density of some outcome y by mapping it to the corresponding $x = h^{-1}(y)$, assessing its density and the Jacobian.

It's possible to draw samples, by drawing from the simple p_X and then mapping to the corresponding $y = h(x)$.

See Normalising Flows (Rezende and Mohamed, 2015; Kingma et al., 2016; Papamakarios et al., 2019)

Sampler - general case

Parameterise an arbitrary transformation of a base random variable

- Sampling: trivial by design (it costs a forward pass through an NN we choose)
- Assessment: intractable in general! For an outcome $y \in \mathbb{R}^O$, a base density $p_X(x)$ on $\mathbb{R}^I$

$$p_Y(y) = \int_{\mathcal{C}} p_X(x)\mathrm{d}x \tag{6}$$

$$\mathcal{C} = \{x : f(x;\theta) = \mathbf{y}\} \tag{7}$$

$\mathcal{C}$ is the set of all points in $\mathbb{R}^I$ that f maps to exactly y

Here f is some deep neural network with I inputs and O outputs (e.g., a feed forward neural network)

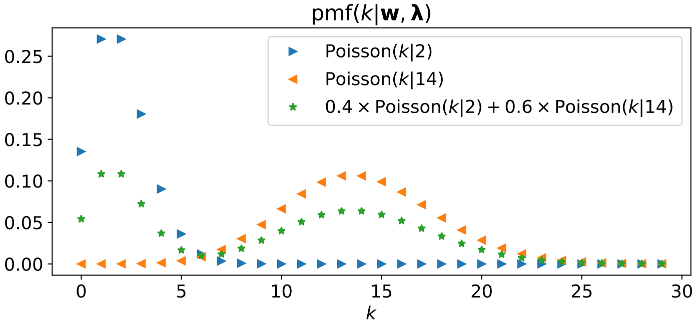
$$x \sim \mathcal{N}(0, I) \tag{8}$$

$$y = f(x;\theta) \tag{9}$$

See Generative Adversarial Network (GAN; Goodfellow et al., 2014) and implicit distributions (Huszár, 2017).

Marginalise a latent variable - mixture model

Predict the parameters of C known pdfs and the coefficients $\mathbf{w}$ of a finite mixture: $p(y|x,\theta) = \sum_{z=1}^{C} w_z p(y|x,\theta_z)$ with $\mathbf{w} \in \Delta_{C-1}$.

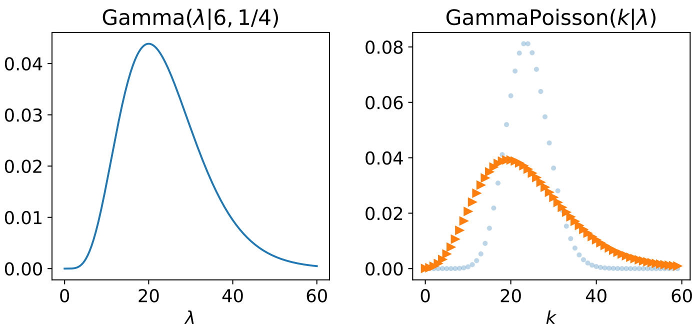


- assessment: linear in C
- sampling: ancestral sampling (draw a component in linear time, then draw from that component)

This simple idea can be used to create very flexible distributions. See, for example, Farinhas et al. (2022), where the components are defined in different subsets of Δ.

Marginalise a latent variable - compounding

Predict a distribution for the parameter(s) of a known parametric family: e.g., $\int_{\mathbb{R}_{>0}} \mathrm{Gamma}(\lambda|\alpha,\beta)\mathrm{Poisson}(y|\lambda)\mathrm{d}\lambda$.



- assessment: typically intractable (some exceptions in the exponential family), a common solution is a variational lowerbound
- sampling: ancestral sampling

Where the Gamma's shape α and rate β are predicted by an NN.

Variation: have λ be predicted by an NN that takes x, α, β as inputs.

- assessment: intractable
- sampling: ancestral sampling

See variational auto-encoders (Rezende et al., 2014; Kingma and Welling, 2014). Various applications in NLP (some of my own work is in this space) and in CV.

Outline

Multivariate

For the fixed-dimension case, we may have access to multivariate generalisations of known pdfs (e.g., MVN).

In general (fixed-dimension or not), we can exploit a *factorisation* with or without conditional independence assumptions.

Then, we predict the factors:

- direct, or cdf, or sampler/simulator
- unnormalised

Directed

Decompose an outcome into parts $y = (w_1, \ldots, w_N)$, for example, a sentence is a sequence of tokens, an image is a sequence of pixels. We fix an order (e.g., left-to-right, row-wise or column-wise, etc).

Factorise $p_{Y|X}(y|x)$ using univariate conditionals.

Examples of factorisation

1. full conditional independence $p_{Y|X}(y|x) \overset{\text{ind.}}{=} \prod_{n=1}^{N} p_{W|X}(w_n|x)$
2. Markov model $p_{Y|X}(y|x) \overset{\text{ind.}}{=} \prod_{n=1}^{N} p_{W|XH}(w_n|x, w_{n-k+1:n-1})$
3. chain rule $p_{Y|X}(y|x) = \prod_{n=1}^{N} p_{W|XH}(w_n|x, w_{<n})$
   aka *autoregressive factorisation*

Given a flexible-enough family for the conditionals, (3) can identify *any* probability measure, in principle.

See Germain et al. (MADE; 2015) and any decoder-only or encoder-decoder model (Mikolov et al., 2010; Van den Oord et al., 2016; Oord et al., 2016; Vaswani et al., 2017)

Undirected

It's possible to factorise $p_{Y|X}(y|x)$ using unnormalised factors.

For example, a first-order conditional random field $p_{Y|X}(y|x) \propto \prod_{n=1}^{N} \Phi(x, y_{n-1}, y_n)$ uses factors like $\Phi(x, y_{n-1}, y_n) > 0$. The familiar constraints apply: non-negativity, finite normalisation constant.

Density/mass assessment, sampling may be possible in some cases (eg, first-order CRF) but are intractable in general.

With rather flexible NNs, we can parameterise an unnormalised model without an explicit factorisation: $p_{Y|X}(y|x, \theta) \propto \mathrm{NN}(x, y; \theta)$. See, for example, energy-based models (EBMs).

For a good class on EBMs, check Module 5 on https://uvadl2c.github.io

Honourable mentions

- Sparse continuous distributions (Martins et al., 2022)
- Score matching (implicit generative models) (Vincent, 2011; Song and Ermon, 2019; Song et al., 2020)
- Diffusion processes (Sohl-Dickstein et al., 2015; Kingma et al., 2021)

For a good class on score matching and diffusion, check Module 5 on https://uvadl2c.github.io

Summary

There are various ways to prescribe distributions both univariate and multivariate.

- Predict parameters for known pdfs and cdfs: we predict some finite (typically small) number of parameters, and evaluate the mass/density of an outcome using a known function.
- For more flexibility we construct novel pdfs or cdfs by predicting unnormalised densities or parameterising flows and simulators.
- For multivariate and structured data we typically exploit a factorisation into simpler distributions (NNs are particularly good at representing complex conditioning contexts).

There are various tradeoffs: is mass/density assessment tractable? can we sample? do we need backward passes? do we need to approximate normalisation constants?

What Next?

There's a lot more to what I said today.

- Big open problems involving parameter estimation, probabilistic inference, and model criticism.
- Creative applications for uncertainty representation (e.g., out-of-domain or error detection, controllable generation).
- Challenging data that may require novel tools.
- Augment our uncertainty representation to include uncertainty about parameters and model family.

If you are excited about LLMs, they too predict a representation of uncertainty: about the response given the prompt.

They exploit an autoregressive factorisation of the conditional distribution, their factors are simple Categorical distributions over a vocabulary of tokens.

Decision making, probabilistic inference, disentanglement learning, representation of epistemic uncertainty, and statistical evaluation are big challenges in that space. See for example our work on decision making (Eikema and Aziz, 2020, 2022) and statistical evaluation (Barkhof and Aziz, 2022; Baan et al., 2022; Giulianelli et al., 2023).

References I

Joris Baan, Wilker Aziz, Barbara Plank, and Raquel Fernandez. Stop measuring calibration when humans disagree. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1892–1915, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. URL https://aclanthology.org/2022.emnlp-main.124.

Claartje Barkhof and Wilker Aziz. Statistical model criticism of variational auto-encoders. *arXiv preprint arXiv:2204.03030*, 2022.

Nicola De Cao, Wilker Aziz, and Ivan Titov. Block neural autoregressive flow. In Ryan P. Adams and Vibhav Gogate, editors, *UAI*, volume 115 of *Proceedings of machine learning research*, pages 1263–1273, Tel Aviv, Israel, July 2020. PMLR. URL http://proceedings.mlr.press/v115/de-cao20a.html. tex.pdf: http://proceedings.mlr.press/v115/de-cao20a/de-cao20a.pdf.

References II

Bruno De Finetti. *Theory of probability: a critical introductory treatment*. 1974.

D. Dubois and H. Prade. *An Introduction to Possibilistic and Fuzzy Logics*, page 742–761. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1990. ISBN 1558601252.

Bryan Eikema and Wilker Aziz. Is MAP Decoding All You Need? The Inadequacy of the Mode in Neural Machine Translation. *arXiv:2005.10283 [cs]*, May 2020. URL http://arxiv.org/abs/2005.10283. arXiv: 2005.10283.

References III

Bryan Eikema and Wilker Aziz. Sampling-based approximations to minimum Bayes risk decoding for neural machine translation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10978–10993, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. URL https://aclanthology.org/2022.emnlp-main.754.

António Farinhas, Wilker Aziz, Vlad Niculae, and Andre Martins. Sparse communication via mixed distributions. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=WAid50QschI.

Nir Friedman and Joseph Y. Halpern. Plausibility measures and default reasoning. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence - Volume 2*, AAAI'96, page 1297–1304. AAAI Press, 1996. ISBN 026251091X.

References IV

Mathieu Germain, Karol Gregor, Iain Murray, and Hugo Larochelle. Made: Masked autoencoder for distribution estimation. In *International conference on machine learning*, pages 881–889. PMLR, 2015.

Mario Giulianelli, Joris Baan, Wilker Aziz, Raquel Fernández, and Barbara Plank. What comes next? evaluating uncertainty in neural text generators against human production variability. *arXiv preprint arXiv:2305.11707*, 2023.

Moisés Goldszmidt and Judea Pearl. Rank-based systems: A simple approach to belief revision, belief update, and reasoning about evidence and actions. In *Proceedings of the Third International Conference on Principles of Knowledge Representation and Reasoning*, KR'92, page 661–672, San Francisco, CA, USA, 1992. Morgan Kaufmann Publishers Inc. ISBN 1558602623.

References V

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

Ian Hacking. *The emergence of probability: A philosophical study of early ideas about probability, induction and statistical inference*. Cambridge University Press, 1975.

Joseph Y Halpern. *Reasoning about uncertainty*. MIT press, 2017.

Jaakko Hintikka. *Modality as referential multiplicity*. Filosofisen Yhdistyksen vuosikirja, 1957.

Jaakko Hintikka. Modality and quantification. *Theoria*, 27(3):119–128, 1961.

References VI

Chin-Wei Huang, David Krueger, Alexandre Lacoste, and Aaron Courville. Neural Autoregressive Flows. In *International Conference on Machine Learning*, pages 2078–2087. PMLR, July 2018. URL http://proceedings.mlr.press/v80/huang18d.html. ISSN: 2640-3498.

Ferenc Huszár. Variational inference using implicit distributions. *arXiv preprint arXiv:1702.08235*, 2017.

Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. *Advances in neural information processing systems*, 34:21696–21707, 2021.

References VII

Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. In Yoshua Bengio and Yann LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. URL http://arxiv.org/abs/1312.6114.

Durk P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved Variational Inference with Inverse Autoregressive Flow. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 4743–4751. Curran Associates, Inc., 2016. URL http://papers.nips.cc/paper/6581-improved-variational-inference-with-inverse-autoregres pdf.

References VIII

Daphne Koller and Nir Friedman. *Probabilistic Graphical Models*. MIT Press, 2009.

Andrey N. Kolmogorov. *Foundations of the Theory of Probability*. Chelsea Pub Co, 2 edition, June 1960.

Dennis V Lindley. *Understanding uncertainty*. John Wiley & Sons, 2013.

Andre Martins and Ramon Astudillo. From Softmax to Sparsemax: A Sparse Model of Attention and Multi-Label Classification. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1614–1623, New York, New York, USA, June 2016. PMLR. URL http://proceedings.mlr.press/v48/martins16.html.

References IX

André FT Martins, Marcos Treviso, António Farinhas, Pedro MQ Aguiar, Mário AT Figueiredo, Mathieu Blondel, and Vlad Niculae. Sparse continuous distributions and fenchel-young losses. *The Journal of Machine Learning Research*, 23(1):11728–11801, 2022.

Christopher Menzel. Possible Worlds. In Edward N. Zalta and Uri Nodelman, editors, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Summer 2023 edition, 2023.

Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. Recurrent neural network based language model. In *Interspeech*, volume 2, pages 1045–1048. Makuhari, 2010.

Vlad Niculae and Mathieu Blondel. A regularized framework for sparse and structured neural attention. *Advances in neural information processing systems*, 30, 2017.

References X

Vlad Niculae, Caio F Corro, Nikita Nangia, Tsvetomila Mihaylova, and André FT Martins. Discrete latent structure in neural networks. *arXiv preprint arXiv:2301.07473*, 2023.

Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing Flows for Probabilistic Modeling and Inference. *arXiv:1912.02762 [cs, stat]*, December 2019. URL http://arxiv.org/abs/1912.02762. arXiv: 1912.02762.

References XI

Ben Peters, Vlad Niculae, and André F. T. Martins. Sparse Sequence-to-Sequence Models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1504–1519, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1146. URL https://www.aclweb.org/anthology/P19-1146.

Barbara Plank. The “problem” of human label variation: On ground truth in data, modeling and evaluation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10671–10682, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. URL https://aclanthology.org/2022.emnlp-main.731.

Frank Plumpton Ramsey. *The foundations of mathematics and other logical essays*. K. Paul, Trench, Trubner & Company, Limited, 1931.

References XII

Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International conference on machine learning*, pages 1530–1538. PMLR, 2015.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic Backpropagation and Approximate Inference in Deep Generative Models. In Eric P. Xing and Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1278–1286, Bejing, China, June 2014. PMLR. URL http://proceedings.mlr.press/v32/rezende14.html. Issue: 2.

Glenn Shafer. *A Mathematical Theory of Evidence*. Princeton University Press, 1976. ISBN 9780691100425. URL http://www.jstor.org/stable/j.ctv10vm1qb.

References XIII

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR, 2015.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

Yang Song, Sahaj Garg, Jiaxin Shi, and Stefano Ermon. Sliced score matching: A scalable approach to density and score estimation. In *Uncertainty in Artificial Intelligence*, pages 574–584. PMLR, 2020.

Aaron Van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. Conditional image generation with pixelcnn decoders. *Advances in neural information processing systems*, 29, 2016.

References XIV

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674, 2011.