

Artificial Intelligence

Probabilities

Marc Toussaint
University of Stuttgart
Winter 2018/19

Motivation:

AI systems need to reason about what they know, or not know. Uncertainty may have so many sources: The environment might be stochastic, making it impossible to predict the future deterministically. The environment can only partially be observed, leading to uncertainty about the rest. This holds especially when the environment includes other agents or humans, the intentions of which are not directly observable. A system can only collect limited data, necessarily leading to uncertain models. We need a calculus for all this. And probabilities are the right calculus.

Actually, the trivial Bayes' rule in principle tells us how we have to process information: whenever we had prior uncertainty about something, then get new information, Bayes' rule tells us how to update our knowledge. This concept is so general that it includes large parts of Machine Learning, (Bayesian) Reinforcement Learning, Bayesian filtering (Kalman & particle filters), etc. The caveat of course is to compute or approximate such Bayesian information processing in practice.

In this lecture we introduce some basics of probabilities, many of which you've learned before in other courses. So the aim is also to recap and introduce the notation. What we introduce is essential for the later lectures on bandits, reinforcement learning, graphical models, and relational probabilistic models.

Outline

- *This set of slides is only for your reference. Only what is *-ed below and was explicitly discussed in the lecture is relevant for the exam.*
- Basic definitions*
 - Random variables*
 - joint, conditional, marginal distribution*
 - Bayes' theorem*
- Probability distributions:
 - Binomial* & Beta
 - Multinomial* & Dirichlet
 - Conjugate priors
 - Gauss* & Wichart
 - Student-t; Dirak; etc
 - Dirak & Particles
- Utilities, decision theory, entropy, KLD
- Monte Carlo*, Rejection & Importance Sampling

Probability Theory

- Why do we need probabilities?

Probability Theory

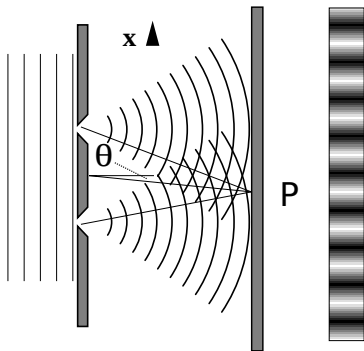
- Why do we need probabilities?
 - Obvious: to express inherent (*objective*) stochasticity of the world

Probability Theory

- Why do we need probabilities?
 - Obvious: to express inherent (*objective*) stochasticity of the world
- But beyond this: (also in a “deterministic world”):
 - lack of knowledge!
 - hidden (latent) variables
 - expressing *uncertainty*
 - expressing *information* (and lack of information)
 - **Subjective Probability**
- Probability Theory: an information calculus

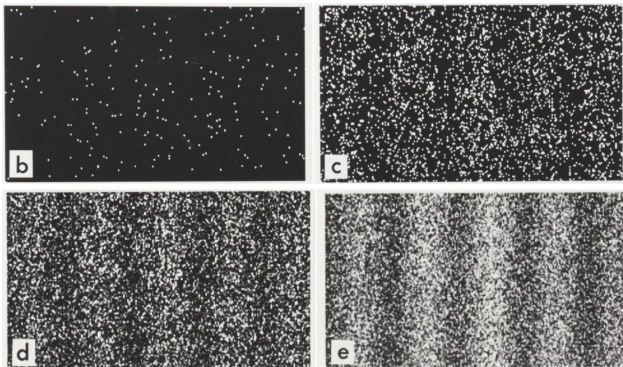
Objective Probability

The double slit experiment:

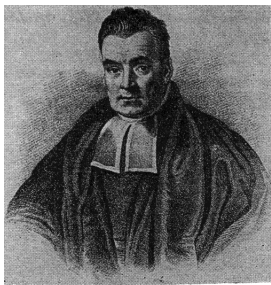


Objective Probability

The double slit experiment:



Thomas Bayes (1702-1761)



~~REV. T. BAYES~~

“Essay Towards Solving a Problem in the Doctrine of Chances”

- Addresses problem of *inverse probabilities*:
Knowing the conditional probability of B given A, what is the conditional probability of A given B?
- Example:
40% Bavarians speak dialect, only 1% of non-Bavarians speak (Bav.) dialect
Given a random German that speaks non-dialect, is he Bavarian?
(15% of Germans are Bavarian)

Inference

- “Inference” = Given some pieces of information (prior, observed variables) what is the implication (the implied information, the posterior) on a non-observed variable

- **Decision-Making and Learning as Inference:**
 - given pieces of information: about the world/game, collected data, assumed model class, *prior* over model parameters
 - make decisions about actions, classifier, model parameters, etc

Probability: Frequentist and Bayesian

- Frequentist probabilities are defined in the limit of an infinite number of trials

Example: “The probability of a particular coin landing heads up is 0.43”

- Bayesian (subjective) probabilities quantify degrees of belief

Example: “The probability of it raining tomorrow is 0.3”

– Not possible to repeat “tomorrow”

Basic definitions

Probabilities & Sets

- **Sample Space/domain** Ω , e.g. $\Omega = \{1, 2, 3, 4, 5, 6\}$

- **Probability** $P : A \subset \Omega \mapsto [0, 1]$

e.g., $P(\{1\}) = \frac{1}{6}$, $P(\{4\}) = \frac{1}{6}$, $P(\{2, 5\}) = \frac{1}{3}$,

- **Axioms:** $\forall A, B \subseteq \Omega$

- Nonnegativity $P(A) \geq 0$

- Additivity $P(A \cup B) = P(A) + P(B)$ if $A \cap B = \{\}$

- Normalization $P(\Omega) = 1$

- **Implications**

$$0 \leq P(A) \leq 1$$

$$P(\{\}) = 0$$

$$A \subseteq B \Rightarrow P(A) \leq P(B)$$

$$P(A \cup B) = P(A) + P(B) - P(A \cap B)$$

$$P(\Omega \setminus A) = 1 - P(A)$$

Probabilities & Random Variables

- For a random variable X with discrete domain $\text{dom}(X) = \Omega$ we write:

$$\forall_{x \in \Omega} : 0 \leq P(X=x) \leq 1$$

$$\sum_{x \in \Omega} P(X=x) = 1$$

Example: A dice can take values $\Omega = \{1, \dots, 6\}$.

X is the random variable of a dice throw.

$P(X=1) \in [0, 1]$ is the probability that X takes value 1.

- A bit more formally: a random variable is a map from a measurable space to the domain (sample space) and thereby introduces a probability measure on the domain

Probability Distributions

- $P(X=1) \in \mathbb{R}$ denotes a specific probability
 $P(X)$ denotes the probability distribution (function over Ω)

Probability Distributions

- $P(X=1) \in \mathbb{R}$ denotes a specific probability
 $P(X)$ denotes the probability distribution (function over Ω)

Example: A dice can take values $\Omega = \{1, 2, 3, 4, 5, 6\}$.

By $P(X)$ we describe the full distribution over possible values $\{1, \dots, 6\}$. These are 6 numbers that sum to one, usually stored in a *table*, e.g.: $[\frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}]$

- In implementations we typically represent distributions over discrete random variables as tables (arrays) of numbers
- Notation for summing over a RV:

In equation we often need to sum over RVs. We then write

$$\sum_X P(X) \dots$$

as shorthand for the explicit notation $\sum_{x \in \text{dom}(X)} P(X=x) \dots$

Joint distributions

Assume we have *two* random variables X and Y

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

x			P_{xy}	

y

- Definitions:

Joint: $P(X, Y)$

Marginal: $P(X) = \sum_Y P(X, Y)$

Conditional: $P(X|Y) = \frac{P(X, Y)}{P(Y)}$

The conditional is normalized: $\forall_Y : \sum_X P(X|Y) = 1$

- X is *independent* of Y iff: $P(X|Y) = P(X)$
(table thinking: all columns of $P(X|Y)$ are equal)

Joint distributions

joint: $P(X, Y)$

marginal: $P(X) = \sum_Y P(X, Y)$

conditional: $P(X|Y) = \frac{P(X, Y)}{P(Y)}$

- Implications of these definitions:

Product rule: $P(X, Y) = P(X|Y) P(Y) = P(Y|X) P(X)$

Bayes' Theorem: $P(X|Y) = \frac{P(Y|X) P(X)}{P(Y)}$

Bayes' Theorem

$$P(X|Y) = \frac{P(Y|X) P(X)}{P(Y)}$$

$$\text{posterior} = \frac{\text{likelihood} \cdot \text{prior}}{\text{normalization}}$$

Multiple RVs:

- Analogously for n random variables $X_{1:n}$ (stored as a rank n tensor)

Joint: $P(X_{1:n})$

Marginal: $P(X_1) = \sum_{X_{2:n}} P(X_{1:n}),$

Conditional: $P(X_1|X_{2:n}) = \frac{P(X_{1:n})}{P(X_{2:n})}$

- X is *conditionally independent* of Y given Z iff:

$$P(X|Y, Z) = P(X|Z)$$

- Product rule and Bayes' Theorem:

$$P(X_{1:n}) = \prod_{i=1}^n P(X_i|X_{i+1:n})$$

$$P(X_1|X_{2:n}) = \frac{P(X_2|X_1, X_{3:n}) P(X_1|X_{3:n})}{P(X_2|X_{3:n})}$$

$$P(X, Z, Y) = P(X|Y, Z) P(Y|Z) P(Z)$$

$$P(X|Y, Z) = \frac{P(Y|X, Z) P(X|Z)}{P(Y|Z)}$$

$$P(X, Y|Z) = \frac{P(X, Z|Y) P(Y)}{P(Z)}$$

Probability distributions

recommended reference: Bishop.: *Pattern Recognition and Machine Learning*

Bernoulli & Binomial

- We have a binary random variable $x \in \{0, 1\}$ (i.e. $\text{dom}(x) = \{0, 1\}$)
The *Bernoulli* distribution is parameterized by a single scalar μ ,

$$P(x=1 | \mu) = \mu, \quad P(x=0 | \mu) = 1 - \mu$$

$$\text{Bern}(x | \mu) = \mu^x (1 - \mu)^{1-x}$$

- We have a data set of random variables $D = \{x_1, \dots, x_n\}$, each $x_i \in \{0, 1\}$. If each $x_i \sim \text{Bern}(x_i | \mu)$ we have

$$P(D | \mu) = \prod_{i=1}^n \text{Bern}(x_i | \mu) = \prod_{i=1}^n \mu^{x_i} (1 - \mu)^{1-x_i}$$

$$\text{argmax}_{\mu} \log P(D | \mu) = \text{argmax}_{\mu} \sum_{i=1}^n x_i \log \mu + (1 - x_i) \log(1 - \mu) = \frac{1}{n} \sum_{i=1}^n x_i$$

- The *Binomial distribution* is the distribution over the count $m = \sum_{i=1}^n x_i$

$$\text{Bin}(m | n, \mu) = \binom{n}{m} \mu^m (1 - \mu)^{n-m}, \quad \binom{n}{m} = \frac{n!}{(n-m)! m!}$$

Beta

How to express uncertainty over a Bernoulli parameter μ

- The *Beta* distribution is over the interval $[0, 1]$, typically the parameter μ of a Bernoulli:

$$\text{Beta}(\mu | a, b) = \frac{1}{B(a, b)} \mu^{a-1} (1 - \mu)^{b-1}$$

with mean $\langle \mu \rangle = \frac{a}{a+b}$ and mode $\mu^* = \frac{a-1}{a+b-2}$ for $a, b > 1$

- The crucial point is:
 - Assume we are in a world with a “Bernoulli source” (e.g., binary bandit), but don’t know its parameter μ
 - Assume we have a *prior* distribution $P(\mu) = \text{Beta}(\mu | a, b)$
 - Assume we collected some data $D = \{x_1, \dots, x_n\}$, $x_i \in \{0, 1\}$, with counts $a_D = \sum_i x_i$ of $[x_i = 1]$ and $b_D = \sum_i (1 - x_i)$ of $[x_i = 0]$
 - The posterior is

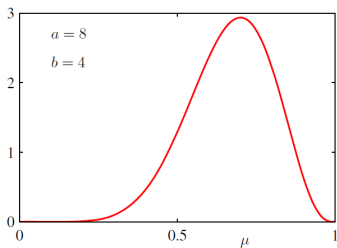
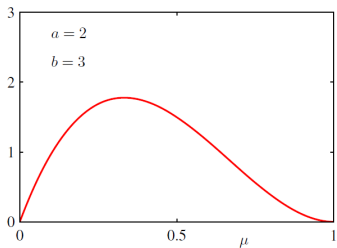
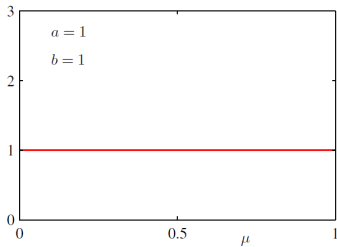
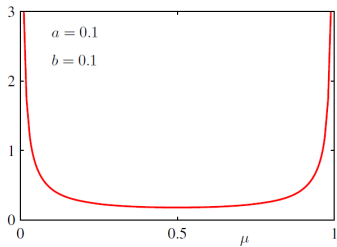
$$\begin{aligned} P(\mu | D) &= \frac{P(D | \mu)}{P(D)} P(\mu) \propto \text{Bin}(D | \mu) \text{Beta}(\mu | a, b) \\ &\propto \mu^{a_D} (1 - \mu)^{b_D} \mu^{a-1} (1 - \mu)^{b-1} = \mu^{a-1+a_D} (1 - \mu)^{b-1+b_D} \\ &= \text{Beta}(\mu | a + a_D, b + b_D) \end{aligned}$$

Beta

The prior is $\text{Beta}(\mu \mid a, b)$, the posterior is $\text{Beta}(\mu \mid a + a_D, b + b_D)$

- Conclusions:
 - The semantics of a and b are counts of $[x_i = 1]$ and $[x_i = 0]$, respectively
 - The Beta distribution is conjugate to the Bernoulli (explained later)
 - With the Beta distribution we can represent beliefs (state of knowledge) about uncertain $\mu \in [0, 1]$ and know how to update this belief given data

Beta



from Bishop

Multinomial

- We have an integer random variable $x \in \{1, \dots, K\}$
The probability of a single x can be parameterized by $\mu = (\mu_1, \dots, \mu_K)$:

$$P(x=k | \mu) = \mu_k$$

with the constraint $\sum_{k=1}^K \mu_k = 1$ (probabilities need to be normalized)

- We have a data set of random variables $D = \{x_1, \dots, x_n\}$, each $x_i \in \{1, \dots, K\}$. If each $x_i \sim P(x_i | \mu)$ we have

$$P(D | \mu) = \prod_{i=1}^n \mu_{x_i} = \prod_{i=1}^n \prod_{k=1}^K \mu_k^{[x_i=k]} = \prod_{k=1}^K \mu_k^{m_k}$$

where $m_k = \sum_{i=1}^n [x_i=k]$ is the count of $[x_i=k]$. The ML estimator is

$$\operatorname{argmax}_{\mu} \log P(D | \mu) = \frac{1}{n} (m_1, \dots, m_K)$$

- The *Multinomial distribution* is this distribution over the counts m_k

$$\operatorname{Mult}(m_1, \dots, m_K | n, \mu) \propto \prod_{k=1}^K \mu_k^{m_k}$$

Dirichlet

How to express uncertainty over a Multinomial parameter μ

- The *Dirichlet* distribution is over the K -simplex, that is, over $\mu_1, \dots, \mu_K \in [0, 1]$ subject to the constraint $\sum_{k=1}^K \mu_k = 1$:

$$\text{Dir}(\mu | \alpha) \propto \prod_{k=1}^K \mu_k^{\alpha_k - 1}$$

It is parameterized by $\alpha = (\alpha_1, \dots, \alpha_K)$, has mean $\langle \mu_i \rangle = \frac{\alpha_i}{\sum_j \alpha_j}$ and mode $\mu_i^* = \frac{\alpha_i - 1}{\sum_j \alpha_j - K}$ for $\alpha_i > 1$.

- The crucial point is:
 - Assume we are in a world with a “Multinomial source” (e.g., an integer bandit), but don’t know its parameter μ
 - Assume we have a *prior* distribution $P(\mu) = \text{Dir}(\mu | \alpha)$
 - Assume we collected some data $D = \{x_1, \dots, x_n\}$, $x_i \in \{1, \dots, K\}$, with counts $m_k = \sum_i [x_i = k]$
 - The posterior is

$$P(\mu | D) = \frac{P(D | \mu)}{P(D)} P(\mu) \propto \text{Mult}(D | \mu) \text{Dir}(\mu | a, b)$$

$$\propto \prod_{k=1}^K \mu_k^{m_k} \prod_{k=1}^K \mu_k^{\alpha_k - 1} = \prod_{k=1}^K \mu_k^{\alpha_k - 1 + m_k}$$

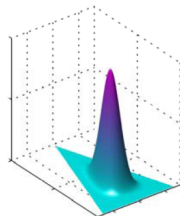
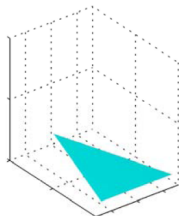
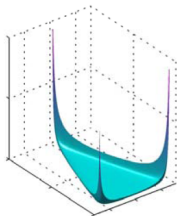
Dirichlet

The prior is $\text{Dir}(\mu \mid \alpha)$, the posterior is $\text{Dir}(\mu \mid \alpha + m)$

- Conclusions:
 - The semantics of α is the counts of $[x_i = k]$
 - The Dirichlet distribution is conjugate to the Multinomial
 - With the Dirichlet distribution we can represent beliefs (state of knowledge) about uncertain μ of an integer random variable and know how to update this belief given data

Dirichlet

Illustrations for $\alpha = (0.1, 0.1, 0.1)$, $\alpha = (1, 1, 1)$ and $\alpha = (10, 10, 10)$:



from Bishop

Motivation for Beta & Dirichlet distributions

- Bandits:
 - If we have binary [integer] bandits, the Beta [Dirichlet] distribution is a way to represent and update beliefs
 - The belief space becomes discrete: The parameter α of the prior is continuous, but the posterior updates live on a discrete “grid” (adding counts to α)
 - We can in principle do belief planning using this
- Reinforcement Learning:
 - Assume we know that the world is a finite-state MDP, but do not know its transition probability $P(s' | s, a)$. For each (s, a) , $P(s' | s, a)$ is a distribution over the integer s'
 - Having a separate Dirichlet distribution for each (s, a) is a way to represent our belief about the world, that is, our belief about $P(s' | s, a)$
 - We can in principle do belief planning using this \rightarrow *Bayesian Reinforcement Learning*
- Dirichlet distributions are also used to model texts (word distributions in text), images, or mixture distributions in general

Conjugate priors

- Assume you have data $D = \{x_1, \dots, x_n\}$ with likelihood

$$P(D | \theta)$$

that depends on an uncertain parameter θ

Assume you have a prior $P(\theta)$

- The prior $P(\theta)$ is **conjugate** to the likelihood $P(D | \theta)$ iff the posterior

$$P(\theta | D) \propto P(D | \theta) P(\theta)$$

is in the *same distribution class* as the prior $P(\theta)$

- Having a conjugate prior is very convenient, because then you know how to update the belief given data

Conjugate priors

likelihood	conjugate
Binomial $\text{Bin}(D \mu)$	Beta $\text{Beta}(\mu a, b)$
Multinomial $\text{Mult}(D \mu)$	Dirichlet $\text{Dir}(\mu \alpha)$
Gauss $\mathcal{N}(x \mu, \Sigma)$	Gauss $\mathcal{N}(\mu \mu_0, A)$
1D Gauss $\mathcal{N}(x \mu, \lambda^{-1})$	Gamma $\text{Gam}(\lambda a, b)$
n D Gauss $\mathcal{N}(x \mu, \Lambda^{-1})$	Wishart $\text{Wish}(\Lambda W, \nu)$
n D Gauss $\mathcal{N}(x \mu, \Lambda^{-1})$	Gauss-Wishart $\mathcal{N}(\mu \mu_0, (\beta\Lambda)^{-1}) \text{Wish}(\Lambda W, \nu)$

Distributions over continuous domain

Distributions over continuous domain

- Let x be a continuous RV. The **probability density function (pdf)** $p(x) \in [0, \infty)$ defines the probability

$$P(a \leq x \leq b) = \int_a^b p(x) dx \in [0, 1]$$

The (cumulative) probability distribution

$F(y) = P(x \leq y) = \int_{-\infty}^y dx p(x) \in [0, 1]$ is the cumulative integral with $\lim_{y \rightarrow \infty} F(y) = 1$

(In discrete domain: *probability distribution* and *probability mass function* $P(x) \in [0, 1]$ are used synonymously.)

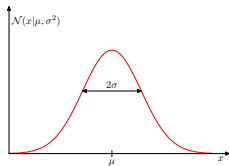
- Two basic examples:

Gaussian: $\mathcal{N}(x | \mu, \Sigma) = \frac{1}{|2\pi\Sigma|^{1/2}} e^{-\frac{1}{2}(x-\mu)^\top \Sigma^{-1} (x-\mu)}$

Dirac or δ (“point particle”) $\delta(x) = 0$ except at $x = 0$, $\int \delta(x) dx = 1$

$\delta(x) = \frac{\partial}{\partial x} H(x)$ where $H(x) = [x \geq 0]$ = Heaviside step function

Gaussian distribution



- 1-dim: $\mathcal{N}(x | \mu, \sigma^2) = \frac{1}{|2\pi\sigma^2|^{1/2}} e^{-\frac{1}{2}(x-\mu)^2/\sigma^2}$
- n -dim Gaussian in *normal form*:

$$\mathcal{N}(x | \mu, \Sigma) = \frac{1}{|2\pi\Sigma|^{1/2}} \exp\left\{-\frac{1}{2}(x - \mu)^\top \Sigma^{-1} (x - \mu)\right\}$$

with **mean** μ and **covariance** matrix Σ . In *canonical form*:

$$\mathcal{N}[x | a, A] = \frac{\exp\{-\frac{1}{2}a^\top A^{-1}a\}}{|2\pi A^{-1}|^{1/2}} \exp\left\{-\frac{1}{2}x^\top A x + x^\top a\right\} \quad (1)$$

with **precision** matrix $A = \Sigma^{-1}$ and coefficient $a = \Sigma^{-1}\mu$ (and mean $\mu = A^{-1}a$).

Note: $|2\pi\Sigma| = \det(2\pi\Sigma) = (2\pi)^n \det(\Sigma)$

- Gaussian identities: see

Gaussian identities

Symmetry: $\mathcal{N}(x | a, A) = \mathcal{N}(a | x, A) = \mathcal{N}(x - a | 0, A)$

Product:

$$\mathcal{N}(x | a, A) \mathcal{N}(x | b, B) = \mathcal{N}[x | A^{-1}a + B^{-1}b, A^{-1} + B^{-1}] \mathcal{N}(a | b, A + B)$$

$$\mathcal{N}[x | a, A] \mathcal{N}[x | b, B] = \mathcal{N}[x | a + b, A + B] \mathcal{N}(A^{-1}a | B^{-1}b, A^{-1} + B^{-1})$$

“Propagation”:

$$\int_y \mathcal{N}(x | a + Fy, A) \mathcal{N}(y | b, B) dy = \mathcal{N}(x | a + Fb, A + FBF^T)$$

Transformation:

$$\mathcal{N}(Fx + f | a, A) = \frac{1}{|F|} \mathcal{N}(x | F^{-1}(a - f), F^{-1}AF^{-T})$$

Marginal & conditional:

$$\mathcal{N}\left(\begin{array}{c} x \\ y \end{array} \middle| \begin{array}{c} a \\ b \end{array}, \begin{array}{cc} A & C \\ C^T & B \end{array}\right) = \mathcal{N}(x | a, A) \cdot \mathcal{N}(y | b + C^T A^{-1}(x - a), B - C^T A^{-1}C)$$

More Gaussian identities: see

<http://ipvs.informatik.uni-stuttgart.de/mlr/marc/notes/gaussians.pdf>

Gaussian prior and posterior

- Assume we have data $D = \{x_1, \dots, x_n\}$, each $x_i \in \mathbb{R}^n$, with likelihood

$$P(D | \mu, \Sigma) = \prod_i \mathcal{N}(x_i | \mu, \Sigma)$$

$$\operatorname{argmax}_{\mu} P(D | \mu, \Sigma) = \frac{1}{n} \sum_{i=1}^n x_i$$

$$\operatorname{argmax}_{\Sigma} P(D | \mu, \Sigma) = \frac{1}{n} \sum_{i=1}^n (x_i - \mu)(x_i - \mu)^\top$$

- Assume we are initially uncertain about μ (but know Σ). We can express this uncertainty using again a Gaussian $\mathcal{N}[\mu | a, A]$. Given data we have

$$\begin{aligned} P(\mu | D) &\propto P(D | \mu, \Sigma) P(\mu) = \prod_i \mathcal{N}(x_i | \mu, \Sigma) \mathcal{N}[\mu | a, A] \\ &= \prod_i \mathcal{N}[\mu | \Sigma^{-1} x_i, \Sigma^{-1}] \mathcal{N}[\mu | a, A] \propto \mathcal{N}[\mu | \Sigma^{-1} \sum_i x_i, n\Sigma^{-1} + A] \end{aligned}$$

Note: in the limit $A \rightarrow 0$ (uninformative prior) this becomes

$$P(\mu | D) = \mathcal{N}\left(\mu \mid \frac{1}{n} \sum_i x_i, \frac{1}{n} \Sigma\right)$$

Motivation for Gaussian distributions

- Gaussian Bandits
- Control theory, Stochastic Optimal Control
- State estimation, sensor processing, Gaussian filtering (Kalman filtering)
- Machine Learning
- etc

Particle Approximation of a Distribution

- We approximate a distribution $p(x)$ over a continuous domain \mathbb{R}^n
- A particle distribution $q(x)$ is a weighed set $\mathcal{S} = \{(x^i, w^i)\}_{i=1}^N$ of N particles
 - each particle has a “location” $x^i \in \mathbb{R}^n$ and a weight $w^i \in \mathbb{R}$
 - weights are normalized, $\sum_i w^i = 1$

$$q(x) := \sum_{i=1}^N w^i \delta(x - x^i)$$

where $\delta(x - x^i)$ is the δ -distribution.

- Given weighted particles, we can estimate for any (smooth) f :

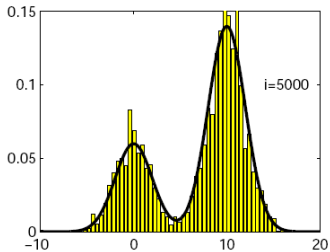
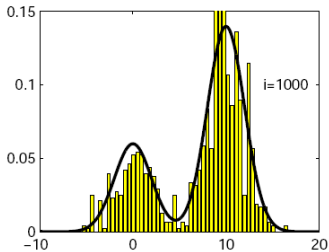
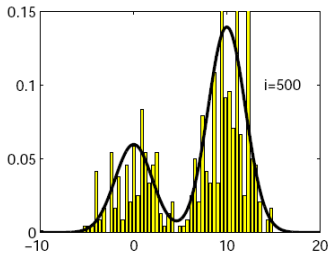
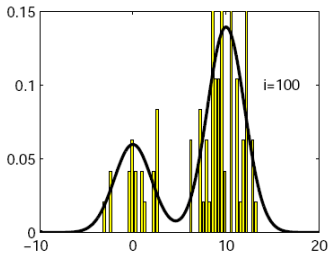
$$\langle f(x) \rangle_p = \int_x f(x)p(x)dx \approx \sum_{i=1}^N w^i f(x^i)$$

See *An Introduction to MCMC for Machine Learning*

www.cs.ubc.ca/~nando/papers/mlintro.pdf

Particle Approximation of a Distribution

Histogram of a particle representation:



Motivation for particle distributions

- Numeric representation of “difficult” distributions
 - Very general and versatile
 - But often needs many samples
- Distributions over games (action sequences), sample based planning, MCTS
- State estimation, particle filters
- etc

Utilities & Decision Theory

- Given a space of events Ω (e.g., outcomes of a trial, a game, etc) the utility is a function

$$U : \Omega \rightarrow \mathbb{R}$$

- The utility represents preferences as a single scalar – which is not always obvious (cf. multi-objective optimization)
- *Decision Theory* making decisions (that determine $p(x)$) that maximize expected utility

$$E\{U\}_p = \int_x U(x) p(x)$$

- Concave utility functions imply risk aversion (and convex, risk-taking)

Entropy

- The neg-log ($-\log p(x)$) of a distribution reflects something like “error”:
 - neg-log of a Gaussian \leftrightarrow squared error
 - neg-log likelihood \leftrightarrow prediction error
- The ($-\log p(x)$) is the “optimal” coding length you should assign to a symbol x . This will minimize the expected length of an encoding

$$H(p) = \int_x p(x)[- \log p(x)]$$

- The **entropy** $H(p) = \mathbb{E}_{p(x)}\{-\log p(x)\}$ of a distribution p is a measure of uncertainty, or lack-of-information, we have about x

Kullback-Leibler divergence

- Assume you use a “wrong” distribution $q(x)$ to decide on the coding length of symbols drawn from $p(x)$. The expected length of a encoding is

$$\int_x p(x)[- \log q(x)] \geq H(p)$$

- The difference

$$D(p \parallel q) = \int_x p(x) \log \frac{p(x)}{q(x)} \geq 0$$

is called Kullback-Leibler divergence

Proof of inequality, using the Jensen inequality:

$$- \int_x p(x) \log \frac{q(x)}{p(x)} \geq - \log \int_x p(x) \frac{q(x)}{p(x)} = 0$$

Monte Carlo methods

Monte Carlo methods

- Generally, a Monte Carlo method is a method to generate a set of (potentially weighted) samples that approximate a distribution $p(x)$. In the unweighted case, the samples should be i.i.d. $x_i \sim p(x)$
In the general (also weighted) case, we want particles that allow to estimate expectations of anything that depends on x , e.g. $f(x)$:

$$\lim_{N \rightarrow \infty} \langle f(x) \rangle_q = \lim_{N \rightarrow \infty} \sum_{i=1}^N w^i f(x^i) = \int_x f(x) p(x) dx = \langle f(x) \rangle_p$$

In this view, Monte Carlo methods approximate an integral.

- Motivation: $p(x)$ itself is too complicated to express analytically or compute $\langle f(x) \rangle_p$ directly
- Example: What is the probability that a solitaire would come out successful? (Original story by Stan Ulam.) Instead of trying to analytically compute this, generate many random solitaires and count.
- Naming: The method developed in the 40ies, where computers became faster. Fermi, Ulam and von Neumann initiated the idea. von Neumann called it "Monte Carlo" as a code name.

Rejection Sampling

- How can we generate i.i.d. samples $x_i \sim p(x)$?
- Assumptions:
 - We can sample $x \sim q(x)$ from a simpler distribution $q(x)$ (e.g., uniform), called **proposal distribution**
 - We can numerically evaluate $p(x)$ for a specific x (even if we don't have an analytic expression of $p(x)$)
 - There exists M such that $\forall_x : p(x) \leq Mq(x)$ (which implies q has larger or equal support as p)
- Rejection Sampling:
 - Sample a candidate $x \sim q(x)$
 - With probability $\frac{p(x)}{Mq(x)}$ accept x and add to \mathcal{S} ; otherwise reject
 - Repeat until $|\mathcal{S}| = n$
- This generates an unweighted sample set \mathcal{S} to approximate $p(x)$

Importance sampling

- Assumptions:
 - We can sample $x \sim q(x)$ from a simpler distribution $q(x)$ (e.g., uniform)
 - We can numerically evaluate $p(x)$ for a specific x (even if we don't have an analytic expression of $p(x)$)
- Importance Sampling:
 - Sample a candidate $x \sim q(x)$
 - Add the weighted sample $(x, \frac{p(x)}{q(x)})$ to \mathcal{S}
 - Repeat n times
- This generates an weighted sample set \mathcal{S} to approximate $p(x)$
The weights $w_i = \frac{p(x_i)}{q(x_i)}$ are called **importance weights**
- Crucial for efficiency: a good choice of the proposal $q(x)$

Applications

- MCTS estimates the Q -function at branchings in decision trees or games
- Inference in graphical models (models involving many depending random variables)

Some more continuous distributions

Gaussian

$$\mathcal{N}(x | a, A) = \frac{1}{|2\pi A|^{1/2}} e^{-\frac{1}{2}(x-a)^T A^{-1} (x-a)}$$

Dirac or δ

$$\delta(x) = \frac{\partial}{\partial x} H(x)$$

Student's t

(=Gaussian for $\nu \rightarrow \infty$, otherwise heavy tails)

$$p(x; \nu) \propto \left[1 + \frac{x^2}{\nu}\right]^{-\frac{\nu+1}{2}}$$

Exponential

(distribution over single event time)

$$p(x; \lambda) = [x \geq 0] \lambda e^{-\lambda x}$$

Laplace

("double exponential")

$$p(x; \mu, b) = \frac{1}{2b} e^{-|x-\mu|/b}$$

Chi-squared

$$p(x; k) \propto [x \geq 0] x^{k/2-1} e^{-x/2}$$

Gamma

$$p(x; k, \theta) \propto [x \geq 0] x^{k-1} e^{-x/\theta}$$