Machine Learning

Probability Basics

Marc Toussaint
University of Stuttgart
Summer 2016
The need for modelling

• Given a real world problem, translating it to a well-defined learning problem is non-trivial

• The “framework” of plain regression/classification is restricted: input \( x \), output \( y \).

• Graphical models (probabilistic models with multiple random variables and dependencies) are a more general framework for modelling “problems”; regression & classification become a special case; Reinforcement Learning, decision making, unsupervised learning, but also language processing, image segmentation, can be represented.
Thomas Bayes (1702-1761)

“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

• Learning as Inference:
  – given pieces of information: data, assumed model, prior over $\beta$
  – non-observed variable: $\beta$
Probability Theory

• Why do we need probabilities?
  – Obvious: to express inherent stochasticity of the world (data)

• But beyond this: (also in a “deterministic world”):
  – lack of knowledge!
  – hidden (latent) variables
  – expressing uncertainty
  – expressing information (and lack of information)

• Probability Theory: an information calculus
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 rain tomorrow is 0.3” – not possible to repeat “tomorrow”
Outline

- Basic definitions
  - Random variables
  - joint, conditional, marginal distribution
  - Bayes’ theorem
- Examples for Bayes
- Probability distributions [skipped, only Gauss]
  - Binomial; Beta
  - Multinomial; Dirichlet
  - Conjugate priors
  - Gauss; Wichart
  - Student-t, Dirak, Particles
- Monte Carlo, MCMC [skipped]

These are generic slides on probabilities I use throughout my lecture. Only parts are mandatory for the AI course.
Basic definitions
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 $\Omega = \{1, .., 6\}$.
$X$ 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 a domain (sample space) and thereby introduces a probability measure on the domain (“assigns a probability to each possible value”)
Probabilty Distributions

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

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

Example: A dice can take values $\Omega = \{1, 2, 3, 4, 5, 6\}$.
By $P(X)$ we describe the full distribution over possible values $\{1, \ldots, 6\}$. 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
  $$\sum_{X} P(X) \cdots$$
  as shorthand for the explicit notation $$\sum_{x \in \text{dom}(X)} P(X = x) \cdots$$
Joint distributions

Assume we have two random variables $X$ and $Y$

- **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

\[ P(X|Y) = \frac{P(Y|X) \cdot P(X)}{P(Y)} \]

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:

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

  \[
P(X_1|X_{2:n}) = \frac{P(X_2|X_{1},X_{3:n}) P(X_1|X_{3:n})}{P(X_2|X_{3:n})}
  \]

  \[
P(X, Z, Y) = P(X|Y, Z) \ P(Y|Z) \ P(Z)
  \]

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

  \[
P(X, Y|Z) = \frac{P(X, Z|Y) \ P(Y)}{P(Z)}
  \]
Example 1: Bavarian dialect

- 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)

\[ P(D = 1 \mid B = 1) = 0.4 \]
\[ P(D = 1 \mid B = 0) = 0.01 \]
\[ P(B = 1) = 0.15 \]
Example 1: Bavarian dialect

- 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)

\[ P(D = 1 \mid B = 1) = 0.4 \]
\[ P(D = 1 \mid B = 0) = 0.01 \]
\[ P(B = 1) = 0.15 \]

If follows
\[ P(B = 1 \mid D = 0) = \frac{P(D = 0 \mid B = 1) \cdot P(B = 1)}{P(D = 0)} = \frac{0.6 \cdot 0.15}{0.6 \cdot 0.15 + 0.99 \cdot 0.85} \approx 0.097 \]
Example 2: Coin flipping

HHTHT

HHHHH

• What process produces these sequences?

• We compare two hypothesis:
  \( H = 1 \) : fair coin \( P(d_i = H \mid H = 1) = \frac{1}{2} \)
  \( H = 2 \) : always heads coin \( P(d_i = H \mid H = 2) = 1 \)

• Bayes’ theorem:

\[
P(H \mid D) = \frac{P(D \mid H)P(H)}{P(D)}
\]
Coin flipping

\[ D = \text{HHTHT} \]

\[ P(D \mid H = 1) = 1/2^5 \quad P(H = 1) = \frac{999}{1000} \]
\[ P(D \mid H = 2) = 0 \quad P(H = 2) = \frac{1}{1000} \]

\[ \frac{P(H = 1 \mid D)}{P(H = 2 \mid D)} = \frac{P(D \mid H = 1)}{P(D \mid H = 2)} \cdot \frac{P(H = 1)}{P(H = 2)} = \frac{1/32}{0} \cdot \frac{999}{1} = \infty \]
Coin flipping

\[ D = \text{HHHHHH} \]

\[ P(D \mid H = 1) = \frac{1}{2^5} \]
\[ P(D \mid H = 2) = 1 \]

\[ P(H = 1) = \frac{999}{1000} \]
\[ P(H = 2) = \frac{1}{1000} \]

\[ \frac{P(H = 1 \mid D)}{P(H = 2 \mid D)} = \frac{P(D \mid H = 1)}{P(D \mid H = 2)} \frac{P(H = 1)}{P(H = 2)} = \frac{1/32}{1} \frac{999}{1} \approx 30 \]
Coin flipping

\[ D = HHHHHHHHHHH \]

\[ P(D \mid H = 1) = 1/2^{10} \quad P(H = 1) = \frac{999}{1000} \]
\[ P(D \mid H = 2) = 1 \quad P(H = 2) = \frac{1}{1000} \]

\[ \frac{P(H = 1 \mid D)}{P(H = 2 \mid D)} = \frac{P(D \mid H = 1)}{P(D \mid H = 2)} \cdot \frac{P(H = 1)}{P(H = 2)} = \frac{1/1024}{1} \cdot \frac{999}{1} \approx 1 \]
Learning as Bayesian inference

\[ P(\text{World}|\text{Data}) = \frac{P(\text{Data}|\text{World}) \cdot P(\text{World})}{P(\text{Data})} \]

\( P(\text{World}) \) describes our prior over all possible worlds. Learning means to infer about the world we live in based on the data we have!
Learning as Bayesian inference

\[ P(\text{World} | \text{Data}) = \frac{P(\text{Data} | \text{World}) \ P(\text{World})}{P(\text{Data})} \]

\( P(\text{World}) \) describes our prior over all possible worlds. Learning means to infer about the world we live in based on the data we have!

- In the context of regression, the “world” is the function \( f(x) \)

\[ P(f | \text{Data}) = \frac{P(\text{Data} | f) \ P(f)}{P(\text{Data})} \]

\( P(f) \) describes our prior over possible functions

Regression means to infer the function based on the data we have.
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 $\mu$,

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

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

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

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

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

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

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

How to express uncertainty over a Bernoulli parameter $\mu$

• The $Beta$ distribution is over the interval $[0, 1]$, typically the parameter $\mu$ of a Bernoulli:

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

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 $\mu$
  – Assume we have a prior distribution $P(\mu) = Beta(\mu | a, b)$
  – Assume we collected some data $D = \{x_1, .., x_n\}$, $x_i \in \{0, 1\}$, 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

$$P(\mu | D) = \frac{P(D | \mu)}{P(D)} \times P(\mu) \propto Bin(D | \mu) \times Beta(\mu | a, b)$$

$$\propto \mu^{a_D} (1 - \mu)^{b_D} \mu^{a-1} (1 - \mu)^{b-1} = \mu^{a-1+a_D} (1 - \mu)^{b-1+b_D}$$

$$= Beta(\mu | a + a_D, b + b_D)$$
Beta

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

- 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 $x \in \{1, .., K\}$
  The probability of a single $x$ can be parameterized by $\mu = (\mu_1, .., \mu_K)$:

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

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

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

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

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

  $$\arg\max_{\mu} \log P(D \mid \mu) = \frac{1}{n} (m_1, .., m_K)$$

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

  $$\text{Mult}(m_1, .., m_K \mid n, \mu) \propto \prod_{k=1}^{K} \mu_{k}^{m_k}$$
Dirichlet

How to express uncertainty over a Multinomial parameter $\mu$

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

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

It is parameterized by $\alpha = (\alpha_1, .., \alpha_K)$, 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 $a_i > 1$.

- 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 $\mu$
  - Assume we have a prior distribution $P(\mu) = \text{Dir}(\mu \mid \alpha)$
  - Assume we collected some data $D = \{x_1, .., x_n\}$, $x_i \in \{1, .., K\}$, with counts $m_k = \sum_i [x_i = k]$
  - The posterior is

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

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

The prior is Dir(μ | α), the posterior is Dir(μ | α + 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 $\alpha$ of the prior is continuous, but the posterior updates live on a discrete “grid” (adding counts to $\alpha$).
  - 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 $D = \{x_1, \ldots, x_n\}$ with likelihood

$$P(D \mid \theta)$$

that depends on an uncertain parameter $\theta$

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

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

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

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

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

$$\lim_{y \to \infty} F(y) = 1$$

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

- Two basic examples:
  - **Gaussian**:
    $$\mathcal{N}(x \mid \mu, \Sigma) = \frac{1}{(2\pi\Sigma)^{1/2}} \, e^{-\frac{1}{2} (x-\mu)^\top \Sigma^{-1} (x-\mu)}$$
  - **Dirac or $\delta$ (“point particle”)**
    $$\delta(x) = 0 \text{ except at } x = 0, \int \delta(x) \, dx = 1$$
    $$\delta(x) = \frac{\partial}{\partial x} H(x) \text{ where } H(x) = [x \geq 0] = \text{Heavyside step function}$$
Gaussian distribution

• 1-dim: \( \mathcal{N}(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi} \sigma^2} e^{-\frac{1}{2} (x-\mu)^2 / \sigma^2} \)

• \( n \)-dim Gaussian in \textit{normal form}:

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

with \textbf{mean} \( \mu \) and \textbf{covariance} matrix \( \Sigma \). In \textit{canonical form}:

\[
\mathcal{N}[x \mid a, A] = \frac{\exp\{ -\frac{1}{2} a^\top A^{-1} a \}}{|2\pi A^{-1}|^{1/2}} \exp\{ -\frac{1}{2} x^\top A^{-1} x + x^\top a \} \tag{1}
\]

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

• Gaussian identities: see http://ipvs.informatik.uni-stuttgart.de/mlr/marc/notes/gaussians.pdf
Gaussian identities

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

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

“Propagation”:
\[
\int_y \mathcal{N}(x \mid a + Fy, A) \mathcal{N}(y \mid b, B) \, dy = \mathcal{N}(x \mid a + Fb, A + FBF^\top)
\]

Transformation:
\[
\mathcal{N}(Fx + f \mid a, A) = \frac{1}{|F|} \mathcal{N}(x \mid F^{-1}(a - f), F^{-1}AF^{-\top})
\]

Marginal & conditional:
\[
\mathcal{N}\left(\begin{array}{c} x \\ y \end{array} \mid \begin{array}{cc} a & A \\ b & C^\top \end{array} \begin{array}{c} C \\ B \end{array}\right) = \mathcal{N}(x \mid a, A) \cdot \mathcal{N}(y \mid b + C^\top A^{-1}(x - a), B - C^\top A^{-1}C)
\]

More Gaussian identities: see
http://ipvs.informatik.uni-stuttgart.de/mlr/marc/notes/gaussians.pdf
Gaussian prior and posterior

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

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

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

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

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

$$P(\mu | D) \propto P(D | \mu, \Sigma) P(\mu) = \prod_i \mathcal{N}(x_i | \mu, \Sigma) \mathcal{N}[\mu | a, A]$$

$$= \prod_i \mathcal{N}[\mu | \Sigma^{-1} x_i, \Sigma^{-1}] \mathcal{N}[\mu | a, A] \propto \mathcal{N}[\mu | \Sigma^{-1} \sum_i x_i, n\Sigma^{-1} + A]$$

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

$$P(\mu | D) = \mathcal{N}(\mu | \frac{1}{n} \sum x_i, \frac{1}{n} \Sigma)$$

which is consistent with the Maximum Likelihood estimator
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 $S = \{(x^i, w^i)\}_{i=1}^N$ 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 $\delta$-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
Particle Approximation of a Distribution

Histogram of a particle representation:

- $i=100$
- $i=500$
- $i=1000$
- $i=5000$
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 $\Omega$ (e.g., outcomes of a trial, a game, etc) the utility is a function
  \[ U : \Omega \to \mathbb{R} \]
- 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 Guassian $\leftrightarrow$ 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 $H(p) = E_{p(x)}\{- \log p(x)\}$ 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

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

- 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

- 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. \( f(x) \):

\[
\lim_{N \to \infty} \langle f(x) \rangle_q = \lim_{N \to \infty} \sum_{i=1}^{N} w^i f(x^i) = \int_x f(x) p(x) \, dx = \langle f(x) \rangle_p
\]

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 solitair would come out successful? (Original story by Stan Ulam.) Instead of trying to analytically compute this, generate many random solitairs 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 $S$; otherwise reject
  – Repeat until $|S| = n$

• This generates an unweighted sample set $S$ 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 $S$
  – Repeat $n$ times

• This generates an weighted sample set $S$ 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 can be viewed as estimating a distribution over games (action sequences) conditional to win
- Inference in graphical models (models involving many depending random variables)
Some more continuous distributions*

Gaussian

\[ \mathcal{N}(x \mid a, A) = \frac{1}{|2\pi A|^{1/2}} e^{-\frac{1}{2} (x-a)^\top A^{-1} (x-a)} \]

Dirac or $\delta$

\[ \delta(x) = \frac{\partial}{\partial x} H(x) \]

Student’s t

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

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

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

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