Lecture Notes for "Stochastic Modeling and Computations"

M. Chertkov (lecturer), S. Belan and V. Parfeneyv (recitation instructors)
M.Sc. and Ph.D. level course at Skoltech
Moscow, March 28 - May 28, 2016
https://sites.google.com/site/mchertkov/courses

The course offers a soft and self-contained introduction to modern applied probability, covering theory and application of stochastic models. Emphasis is placed on intuitive explanations of the theoretical concepts, such as random walks, law of large numbers, Markov processes, reversibility, sampling, etc., supplemented by practical/computational implementations of basic algorithms. In the second part of the course, the focus shifts from general concepts and algorithms per se to their applications in science and engineering with examples, aiming to illustrate the models and make the methods of solution, originating from physics, chemistry, machine learning, control and operations research, clear and exciting.

Contents

Brief Description. Structure. Requirements.

This course is recommended to M.Sc. and Ph.D. students planning to work on the subjects containing elements of uncertainty, irregularity or what is also called 'stochasticity'. The 'stochastic' subjects are prevalent in natural sciences (physics, chemistry, biology) or engineering disciplines (electrical-, mechanical-, chemical-, industrial-, etc). The course introduces students to modeling and computational concepts, approaches, methods and algorithms which require dealing with stochasticity and uncertainty.

This is a general course recommended to Energy, IT and other Skoltech students. The course is recommended as a core course for students who specializes in computations. It can be chosen as elective by students who use computations and algorithms in their work, however not as the prime focus.

There will be 12 lectures, 11 recitations, 2 homework assignments, a special recitation explaining homework solutions, journal club presentations+reports and, finally, the exam.

Lectures. Lecture notes are to be provided (online) before the actual lecture. Lecturer will mainly be using white-board, sometimes supplemented by computer demonstrations in IJulia.

Recitations. Recitation notes are to be provided (online) after the actual recitations. Recitations will be lead by two instructors (alternating) with the use of whiteboard and computer demonstrations (ijulia notebooks). Students in the class will be called to lead solution of some problems.

Homework assignments. Two assignments will be given. Each homework will consist of ~ 6 problems, including multiple ($\sim 2-4$) sub-problems of varying difficulty. First homework will be distributed in the beginning of the first week and will be collected by Wed, Apr 20, 11:59pm Moscow time. Second homework will be distributed in the beginning of the 4th week and will be collected by Wed, May 18, 11:59pm Moscow time. Problems in the homework will be similar in principle, but different in details from these discussed in lectures and recitations (prior to the homework distribution). Solutions from the homework will be discussed at the recitations after the homework collection. It is encouraged to use electronic formats (latex and/or ipython/ijulia) for the homework reports. Submission of the homework(s) is electronic only.

Each student will be required to choose a subject for **journal club presentation and report**. List of suggested subjects is listed below in the document. In terms of picking a subject – the policy is 'first come first served'. The list is not meant to be complete or exclusive. In particular, the students are encouraged to suggest additional subjects linked to the course material and possibly related to their own research focus/interest. All additional subjects should be discussed with and approved by the lecturer. Subjects should be presented during the presentation session (tentatively) scheduled for May 24. Each presentation is 20 mins. All reports should be submitted by May 28, 11:59pm. Reports are individual, should be at least 10 pages but not longer that 20 pages. Presentations and reports will be graded together.

A written **exam** will be administered. The exam will include $3-4$ problems similar to these discussed at the recitations and contained in the homework. Format of the exam (in class or take home) will be decided at later time depending on how the class progresses.

The three **books** referred extensively in lectures, recitations and homework are [1–3]. In addition, many relevant reviews and papers available online are cite in the lecture notes. Students may also find it useful to check [4–7] for related (but often alternative) explanations. (A number of hard copies of all the aforementioned books are available at the edu@skoltech library.

On pre-requisites and requirements. All necessary concepts from statistics, probability theory and statistical mechanics will be introduced in the course self-consistently (no formal pre-requisites in these disciplines are required). However, solid preparation in practical math (ability to solve problems in linear algebra, calculus, and differential equations) will be required from anybody taking this course.

We will mainly be using in lectures and recitations for computations and illustrations Julia `http://julialang.org/` under IJulia/Ipython-notebook environment `https://github.com/JuliaLang/IJulia.jl`. Students are encouraged to self-learn and use Julia and IJulia. However, computations (e.g. for homework and exam) in any other (reasonably common and transparent) programming languages will also be accepted.

Grading:

- Homework – 35%
- Exam – 35%
- Journal Club Presentation & Report – 20%
- Participation – 10%

(Tentative) Schedule

Mon, Tue, Thu 9:00–10:30 + 10:30–12:00 (two periods 1.5 hours each)
l=lecture, r=recitation

First week:

March 28, Mon
9:00–10:30 l#1 Random Variables: Characterization and Description
10:30–12:00 free period
March 29, Tue
9:00–10:30 l#2 Random Variables: Operations & Transformations
10:30–12:00 free period
March 31, Thu
9:00–10:30 r#1 Moments/Averages/Cumulants/Generation Function on Examples
10:30–12:00 r#2 Example of Gaussian Variables: Matrix Inversion, Normalization, Moments

Second week:

Apr 4, Mon
9:00–10:30 l#3 Information-Theoretic View on Randomness
10:30–12:00 r#3 Entropy, Mutual Information and Probabilistic Inequalities on Example (Communication over Noisy Channel)
Apr 5, Tue
9:00–10:30 l#4 Markov Chains [discrete space, discrete time]
10:30–12:00 l#5 From Bernoulli Processes to Poisson Processes [discrete space, discrete & continuous time]
Apr 7, Thu
free day

Third week:

Apr 11, Mon
9:00–10:30 r#4 Markov Chains: Detailed Balance. Mixing time.
10:30–12:00 r#5 Examples of Bernoulli & Poisson Processes
Apr 12, Tue
9:00–10:30 free period
10:30–12:00 l#6 Monte-Carlo Algorithms: General Concepts and Direct Sampling
Apr 14, Thu
9:00–10:30 l#7 Markov-Chain Monte-Carlo
10:30–12:00 r#6 MC and MCMC on example of the Ising model

Fourth week:

Apr 18, Mon
9:00–10:30 l#8 Exact & Approximate Inference
10:30–12:00 l#9 Inference & Learning with Belief Propagation
Apr 19, Tue
9:00–10:30 r#7 Inference & Learning on Trees
10:30–12:00 l#10 Space-time Continuous Stochastic Processes
Apr 21, Thu
free day

Fifth week:

Apr 25, Mon
9:00–10:30 r#8 Homogeneous and Forced Brownian Motion
10:30–12:00 r#9 First Passage Problem and Effects of Boundaries
Apr 26, Tue
9:00–10:30 free period
10:30-12:00 l#11 Queuing Systems
Apr 28, Thu
9:00–10:30 free period 10:30–12:00 l#12 Markov Decision Processes & Stochastic Optimal Control

Sixth week:

May 16, Mon
9:00–10:30 r#10 Queuing Systems
10:30–12:00 r#11 Markov Decision Processes & Stochastic Optimal Control
May 17, Tue
free day
May 19, Thu
9:00–10:30 solution of homework (#1 & #2)
10:30–12:00 consultation for exam (questions-answers with TAs)

Seventh week:

May 23, Mon
nothing is scheduled (reserve)
May 24, Tue
9:00–12:00 & 14:00-17:00 Project presentations
May 26, Thu
9:00-13:00 exam

I. THEME #1. BASIC CONCEPTS FROM STATISTICS

A. Lecture #1. Random Variables: Characterization & Description.

1. Probability of an event

Discrete vs Continuous events. State/phase/sample space (for events), Σ.

Example of discrete events: two states, $\Sigma = \{0, 1\}$ -also called Bernoulli random variable (derived from a "process", i.e. dynamics - to be discussed later in the course a lot). Probability of a state, σ,

$$\forall \sigma \in \Sigma: \quad \text{Prob}(\sigma) = P(\sigma) \tag{I.1}$$

$$0 \leq P(\sigma) \leq 1 \tag{I.2}$$

$$\sum_{\sigma \in \Sigma} P(\sigma) = 1 \tag{I.3}$$

For Bernoulli process, $P(1) = \beta$, $P(0) = 1 - \beta$.

Question: Can you give an example of the Bernoulli distribution from life/science?

Answer: A biased coin.

Another important discrete event distribution is the Poisson. An event can occur $k = 0, 1, 2, \cdots$ times in an interval. The average number of events in an interval is λ - called event rate. The probability of observing k events within the interval is

$$\forall k \in \mathbb{Z}^* = \{0\} \bigcup \mathbb{Z}: \quad P(k) = \frac{\lambda^k e^{-\lambda}}{k!}. \tag{I.4}$$

(Check that the probability is properly normalized, in the sense of Eq. (I.3).) The distribution is also called exponential distribution (for obvious reason).

Questions: Are Bernoulli and Poisson distributions related? Can you "design" Poisson from Bernoulli? Can you give an example of the Poisson process from life/science?

Answer: Consider repeating Bernoulli - thus drawing a Bernoulli process. You get sequence of zeros and ones. Then check only for ones and record times/slots associated with arrivals of ones. Study probability distribution of t arrivals in n step, and then analyze $n \to \infty$, to get the Poisson distribution. We will discuss it in details in Lect.# 5. Example of Poisson — arrival of customers at the shop.

The domain can be continuous, bounded or unbounded. Example of distribution which is bounded - is uniform distribution from the $[0, 1]$ interval:

$$\forall x \in [0, 1]: \quad p(x) = 1, \tag{I.5}$$

$$\int_0^1 dx p(x) = 1, \tag{I.6}$$

where $p(x)$ is the probability density. Gaussian distribution is the most important continuous distribution:

$$\forall x \in \mathbb{Z}: \quad p(x|\sigma, \mu) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right). \tag{I.7}$$

$p_{\sigma,\mu}(x)$ another possible notation. It is also called "normal distribution" - where "normality" refers to significance of the distribution for the central limit theorem (law of large numbers), which we will be discussing shortly. The distribution is parameterized by μ and λ - what is the significance of the two parameters? (mean and variance) Standard notation in math for the Gaussian/normal distribution is $N(\mu, \sigma^2)$.

There are many more 'standard' distributions but Bernoulli, Poisson and Gaussian are the 'golden' three. One can generate practically any other distribution from the 'golden set' (possibly extended by the uniform distribution.

Some discussion of notations, e.g. $P_X(x)$, $\mathbb{E}[\cdots] = \langle \cdots \rangle$.

2. Sampling. Histograms.

Random process generation. Random process is generated/sampled. Any computational package/software contains a random number generator (in fact a number of these). Designing a good random generation is important. In this course, however, we will mainly using the random number generators (in fact pseudo-random generators) already created by others.

To illustrate let us switch to IJulia notebook of the first two lectures.

Histogram. To show distributions graphically, you may also "bin" it in the domain - thus generating the histogram, which is a convenient way of showing $p(\sigma)$ (plot with Julia: breaking $[0,1]$ interval in $N>1$ bins). Use it as an opportunity to introduce statistical computational package (Julia should have one too).

3. Moments. Generating Function.

Expectations.

$$\mathbb{E}\left[A(\sigma)\right]_p = \langle A(\sigma)\rangle_p = \sum_{\sigma\in\Sigma} A(\sigma)p(\sigma).$$

Examples: mean,

$$\mathbb{E}\left[\sigma\right],$$

variance,

$$\text{Var}[\sigma] = \mathbb{E}\left[(\sigma - \mathbb{E}[\sigma])^2\right].$$

We have already discussed these for the Gaussian process. What are mean and variance for Bernoulli process?

Moments.

$$k = 0, \cdots, \quad \mathbb{E}\left[\sigma^k\right]_p = \langle \sigma^k\rangle_p = \sum_{\sigma\in\Sigma} \sigma^k p(\sigma) = m_k(\Sigma)$$

Moment Generating Function.

$$M_X(t) = \mathbb{E}\left[\exp(tx)\right], \quad t\in\mathbb{R}$$

One can also view it as a Laplace transform of the probability density function,

$$M_X(t) = \int dx p(x)\exp(tx)$$

. Examples of the moment generating functions for aforementioned (and other) distributions — derived it yourself, see tables online ... and it will also be discussed at the recitations. Characteristic function is a related object — Fourier transform of the probability density:

$$\mathbb{E}\left[\exp(itx)\right] = \int dx p(x)\exp(itx)$$

where $i^2 = -1$.

4. Probabilistic Inequalities.

Here are some useful probabilistic inequalities.

- (Markov Inequality)

$$P(x\geq c) \leq \frac{\mathbb{E}[x]}{c} \tag{I.8}$$

- (Chebyshev's inequality)

$$P(|x-\mu|\geq k) \leq \frac{\sigma^2}{k^2} \tag{I.9}$$

- (Chernoff bound)

$$P(x \geq a) = P(e^{tx} \geq e^{ta}) \leq \frac{\mathbb{E}[e^{tx}]}{e^{ta}} \tag{I.10}$$

where μ and σ are mean and variance of x.

We will get back to discussion of these and some additional inequalities in the third lecture.

Exercise: Play in IJulia checking the three inequalities for the distributions mentioned through out the lecture.

Exercise: Prove the Markov inequality. Chebyshev inequality will follow from the Markov, prove it too. Chernoff is trickier, can you prove it too? [See `http://jeremykun.com/2013/04/15/probabilistic-bounds-a-primer/` to check your answers]

Exercise: Provide examples of the distributions for which the tree inequalities are saturated (becomes equalities)?

5. *Recitation. Random Variables. Moments. Characteristic Function.*

B. Lecture #2. Random Variables: from one to many.

1. *Law of Large Numbers*

Take n samples $x_1, \cdots, x_n$ generated i.i.d. from a distribution with mean μ and variance, $\sigma > 0$, and compute $y_n = \sum_{i=1}^n x_i/n$. What is Prob(y_n)? $\sqrt{n}(y_n - \mu)$, converges in distribution to Gaussian with mean, μ, and variance, σ:

$$\sqrt{n}\left(\frac{1}{n}\sum_{i=1}^{n} x_i - \mu\right) \xrightarrow{d} N(0,\ \sigma^2). \tag{I.11}$$

This is so-called Weak Version of the Central Limit Theorem. (Large Deviation Theorem is an alternative name.)

Let us prove the weak-CLT (I.11) in a simple case $\mu = 0$, $\sigma = 1$. (Generalization is obvious.) Obviously, $m_1(Y_n\sqrt{n}) = 0$. Compute

$$m_2(Y_n\sqrt{n}) = \mathbb{E}\left[\left(\frac{x_1 + \cdots + x_n}{\sqrt{n}}\right)^2\right] = \frac{\sum_i \mathbb{E}\left[x_i^2\right]}{n} + \frac{\sum_{i\neq j} \mathbb{E}\left[x_i x_j\right]}{n} = 1.$$

Now the third moment:

$$m_3(Y_n\sqrt{n}) = \mathbb{E}\left[\left(\frac{x_1 + \cdots + x_n}{\sqrt{n}}\right)^3\right] = \frac{\sum_i \mathbb{E}\left[x_i^3\right]}{n^{3/2}} \to 0,$$

at $n \to \infty$, assuming $\mathbb{E}\left[x_i^3\right] = O(1)$. Can you guess what will happen with the fourth moment? $m_4(Y_n\sqrt{n}) = 3 = 3m_2(Y_n)$ - Wick theorem (physics jargon). And how about higher odd/even moments?

Exercise: Check IJulia notebok for the lecture and experiment with the law of large numbers for different distributions.

The theorem holds for independent but not identically distributed variables too.

If one is interested in not only the asymptotic itself, $n \to \infty$, but also in how the asymptotic is approached, the so-called strong version of CLT (can also be found in some literature under the name of Cramér theorem) states

$$\forall z > \mu: \quad \lim_{n\to\infty} \frac{1}{n} \log \text{Prob}\left(y_n \geq z\right) = -\Phi^*(z) \tag{I.12}$$

$$\Phi^*(z) \doteq \sup_{\lambda\in\mathbb{R}} \left(\lambda z - \Phi(z)\right) \tag{I.13}$$

$$\Phi(\lambda) \doteq \log\left(\mathbb{E}\exp\left(\lambda z\right)\right) \tag{I.14}$$

$\Phi^*(z)$ is a convex function (also called Cramér function). This was a formal (mathematical) statement. A less formal (physical) version of Eq. (I.12) is

$$n \to \infty: \quad \text{Prob}\left(y_n\right) \propto \exp\left(-n\Phi^*(x)\right) \tag{I.15}$$

One of our journal club projects is on this subject.

Note, that the weak version of the CLT (I.11) is equivalent to approximating the Cramer function (asymptotically exact) by a Gaussian around its minimum.

Exercise (bonus): Prove the strong-CLT (I.12,I.13). [Hint: use saddle point/stationary point method to evaluate the integrals.]

Exercise: Give an example of an expectation for which not only vicinity of the minimum but also other details of $\Phi^*(x)$ are significant at $n \to \infty$? More specifically give an example of the object which behavior is controlled solely by left/right tail of $\Phi^*(x)$? $\Phi^*(0)$ and its vicinity?

Example of Bernoulli process – a (possibly unfair) coin toss

$$x = \begin{cases} 0 & \text{with probability } 1-p \\ 1 & \text{with probability } p \end{cases} \tag{I.16}$$

Then

$$\Phi(\lambda) = \log(pe^{\lambda} + 1 - p) \tag{I.17}$$

$$0 < x < 1: \quad \Phi^*(z) = z \log \frac{z}{p} + (1-z) \log \frac{1-z}{1-p} \tag{I.18}$$

Eqs. (I.17,I.18) are noticeable for two reasons. First of all it leads (after some algebraic manipulations) to the famous Stirling formula for the asymptotic of a factorial

$$n! = \sqrt{2\pi n} n^n e^{-n} \left(1 + O(1/n)\right)$$

. Do you see how? Second, the $z \log z$ structure is an "entropy" which will appear few more times in the course - stay tuned.

2. *Multivariate Distribution. Marginalization. Conditional Probability.*

Consider an n-component vector build of components each taking a value from a set, Σ, $\sigma = (\sigma_i \in \Sigma | i = 1, \cdots, n)$. Σ may be discrete, e.g. $\Sigma = \{0, 1\}$, or continuous, e.g. $\Sigma = \mathbb{R}$. Assume that any state, σ, occur with the probability, $P(\sigma)$, where $\sum_\sigma P(\sigma) = 1$.

Consider a simple example of bi-variate distribution.

$$\sigma = (\sigma_i = \pm 1 | i = 1, \cdots, n) : P(\sigma) = Z^{-1} \prod_{i=1}^{n-1} \exp(J \sigma_i \sigma_{i+1}) \tag{I.19}$$

$$Z = \sum_\sigma \prod_{i=1}^{n-1} \exp(J \sigma_i \sigma_{i+1}) \tag{I.20}$$

where Z is the normalization constant (also called partition function in physics), introduced to guarantee that the sum over all the states is unity. For $n = 2$ we can also write

$$P(\sigma) = P(\sigma_1, \sigma_2) = \frac{\exp(J \sigma_1 \sigma_2)}{4 \cosh(J)}. \tag{I.21}$$

$P(\sigma)$ is also called a joint distribution function of the σ vector components, $\sigma_1, \cdots, \sigma_n$. It is also useful to consider conditional distribution, say for the example above with $n = 2$,

$$P(\sigma_1 | \sigma_2) = \frac{P(\sigma_1, \sigma_2)}{\sum_{\sigma_1} P(\sigma_1, \sigma_2)} = \frac{\exp(J \sigma_1 \sigma_2)}{2 \cosh(J \sigma_2)} \tag{I.22}$$

is the probability to observe σ_1 under condition that σ_2 is known. Notice that, $\sum_{\sigma_1} P(\sigma_1 | \sigma_2) = 1, \ \forall \sigma_2$.

Let us now marginalize the multivariate (joint) distribution over a subset of variables. For example,

$$P(\sigma_1) = \sum_{\sigma \setminus \sigma_1} P(\sigma) = \sum_{\sigma_2, \cdots, \sigma_n} P(\sigma_1, \cdots, \sigma_n). \tag{I.23}$$

We will repeat exercises (joint, conditional, marginal) with multivariate Gaussian distribution at the recitations. The Gaussian distributions are remarkably unique, because application of any of the aforementioned operations, joint-to-conditional and joint-to-marginal, will also be Gaussian ... not to mention that the Gaussian emerges naturally in the result of the CLT.

3. Bayes Theorem

We already saw how to get conditional distribution and marginal distribution from the joint one

$$P(x|y) = \frac{P(x,y)}{P(y)}, \quad P(y|x) = \frac{P(x,y)}{P(x)}. \tag{I.24}$$

Combining the two formulas to exclude the joint probability distribution we arrive at the famous Bayes formula

$$P(x|y)P(y) = P(y|x)P(x). \tag{I.25}$$

Here, in Eqs. (I.24,I.26) both x and y may be multivariate.

Rewriting Eq. (I.26) as

$$P(x|y) = \frac{P(y|x)P(x)}{P(y)}, \tag{I.26}$$

one often refers (in the field of the so-called Bayesian inference/reconstruction) to $P(x)$ as the "prior" probability – degree of initial "belief" in x, $P(x|y)$ - the "posterior" - degree of belief having accounted for y, and quotient $\frac{P(y|x)}{P(y)}$ representing "support/knowledge" y provides for x.

A good illustration of the notion of conditional probability can be found at `http://setosa.io/ev/conditional-probability/`

Let us conclude the lecture playing with a made up bi-variate binary (with the total of 2^2 states) case.

4. Recitation. Properties of Gaussian Distributions. Laws of Large Numbers.

C. Lecture #3. Information-Theoretic View on Randomness

1. Entropy.

Entropy is defined as an expectation of $-log$-probability

$$S(X) = -\mathbf{E}\left[\log(P(x))\right] = -\sum_{x\in\mathcal{X}} P(x)\log(P(x)). \tag{I.27}$$

Intuitively, entropy is a measure of uncertainty. Simple illustration that entropy of a deterministic process (when a state happens with probability 1) is 0 ($\lim_{p\to 0} p\log p = 0$). Note, that following statistical physics tradition we use S for entropy, while it is also custom in information theory to use H for the same object.

Importantly, logarithm of the probability distribution is chosen as a measure of information in the definition of entropy (and not another function) because it is **additive** for independent sources.

Let us familiarize ourselves with the concepts of entropy on the example of the Bernoulli $\{0,1\}$ process (I.16)

$$S(X) = -p\log p - (1-p)\log(1-p). \tag{I.28}$$

If we plot the entropy as the function of p. It has a bell-like shape with the maximum at $p = 1/2$ - fare coin has the largest entropy (most uncertain). Entropy is zero at $p = 0$ and $p = 1$ - the two cases are deterministic, i.e. fully certain.

Entropy (I.27), $S(X)$, has the following properties (some can be interpreted as alternative definitions):

- $S(X) \geq 0$
- $S(X) = 0$ iff x is deterministic.
- $S(X) \leq \log(|\mathcal{X}|)$ and $S(X) = \log(|\mathcal{X}|)$ iff x is distributed uniformly over the set $\mathcal{X}$.
- Choice of the logarithm base is custom - just a re-scaling. (Base 2 is custom in the information , when dealing with binary variables.)
- Entropy is the measure of average uncertainty.
- Entropy is less than the average number of bits needed to describe the random variable (the equality is achieved for uniform distribution). (*)

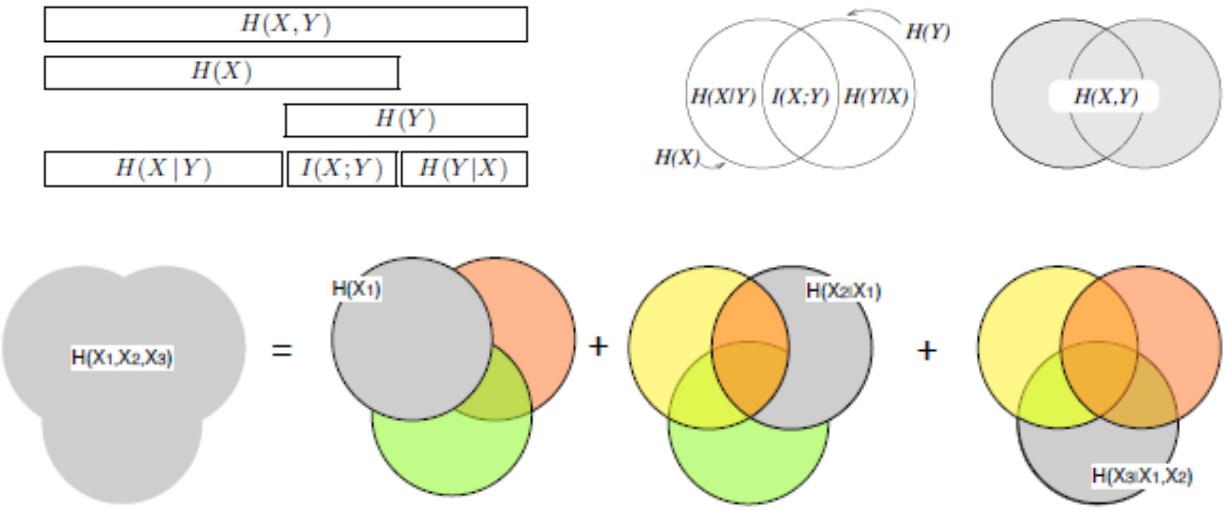

FIG. 1: Venn diagram(s) explaining the chain rule for computing multivariate entropy.

- Entropy is the lower bound on the average length of the shortest description of the random variable

(*) requires a clarification. Take integers which are smaller or equal then n, and represent them in the binary system. We will need $\log_2(n)$ binary variables (bits) to represent any of the integers. If all the integers are equally probable then $\log_2(n)$ is exactly the entropy of the distribution. If the random variable is distributed non-uniformly than the entropy is less than the estimate.

Exercise: Order the following three cases in terms of entropy: (a) 5 equally probable states; b) 3 states which happens with the probabilities $1/2, 1/6, 1/3$; c) 6 states which happen with the probabilities $1/2, 1/10, 1/10, 1/10, 1/10, 1/10$.

If we have a pair of (discrete for concreteness) variables, $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ their joint entropy is

$$S(X,Y) \doteq -\sum_{y\in\mathcal{Y}} p(x,y)\log(p(x,y)). \tag{I.29}$$

Conditional entropies are

$$S(Y|X) \doteq -\mathbb{E}_{p(x,y)}\left[\log(p(y|x)\right] = -\sum_{y\in\mathcal{Y}} p(x,y)\log(p(y|x)). \tag{I.30}$$

Note, that $S(Y|X) \neq S(X|Y)$.

The so-called chain rule states (check)

$$S(X,Y) = S(X) + S(Y|X). \tag{I.31}$$

One can also extend it to the multi-variate case $(X_1, \cdots, X_n) \sim P(x_1, \cdots, x_n)$ (this notation is standard in statistics) becomes

$$S(X_n, \cdots, X_1) = \sum_{i=1}^{n} S(X_i|X_{i-1}, \cdots, X_1). \tag{I.32}$$

The name "chain-rule" should become clear from (I.32). The chain rule is illustrated in Fig. (1).

2. *Independence/Dependence. Mutual Information.*

The essence of our next theme is in comparing random numbers, or more accurately thei probabilities. Kublack-Leibler divergence offers a convenient way of measuring two probabilities

$$D(p_1\|p_2) \doteq \sum_{x\in\mathcal{X}} p_1(x)\log\frac{p_1(x)}{p_2(x)}. \tag{I.33}$$

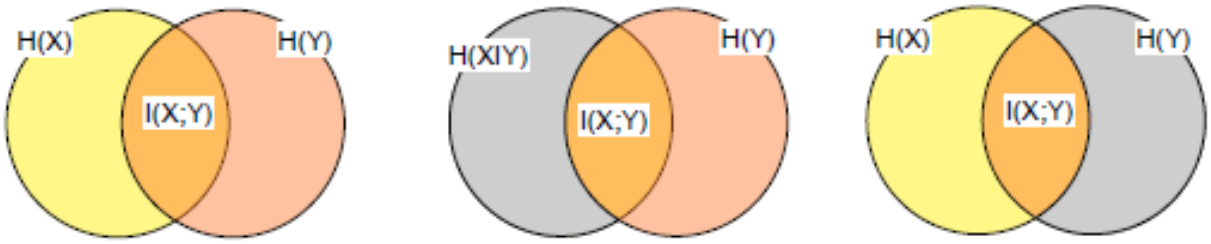

FIG. 2: Venn diagram explaining relations between mutual information and entropies.

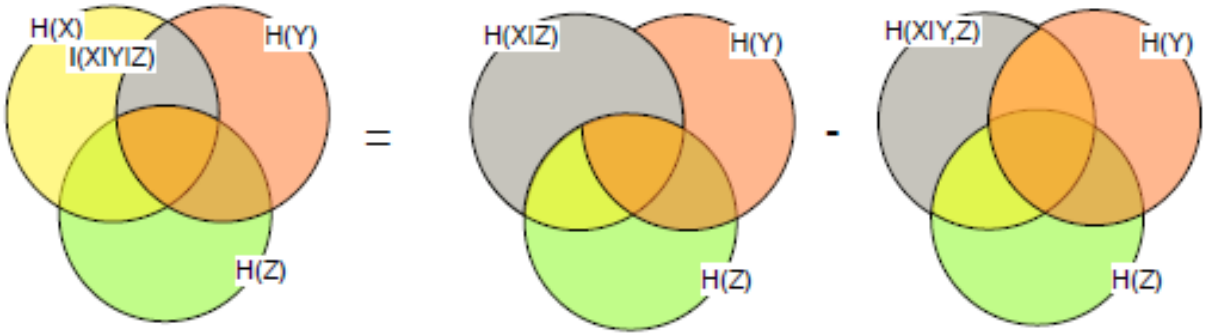

FIG. 3: Venn diagram explaining the chain rules for mutual information.

Note that the KL difference is not symmetric wrt exchange of the order of the two distribution. Moreover it is not a proper metric of comparison (it does not satisfy the triangle inequality (Any proper metric of a space should be a) positive, b) zero when comparing identical states; c) symmetric, and d) satisfy the triangle inequality, $d_(a,b) \leq d_{a,c} + d_{bc}$. The last two do not satisfy in the case of the KL divergence. However, an infintesimal version of KL divergence - Hesian of the KL distance, related to the so-called Fisher informarmation.)

Comparing the two information sources, say tracking events x and y, the extreme case is when the probabilities are independent, i.e. $P(x,y) = P(x)P(y)$ and $P(x|y) = P(x)$, $P(y|x) = P(y)$. Mutual information is the measure of dependence

$$I(X;Y) = \mathbb{E}_{P(x,y)}\left[\log \frac{P(x,y)}{P(x)P(y)}\right] = \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} P(x,y)\log\frac{P(x,y)}{P(x)P(y)}. \tag{I.34}$$

Intuitively the mutual information measures the information that x and y share. In other words, it measures how much knowing one of these variables reduces uncertainty about the other. For example, if x and y are independent, then knowing x does not give any information about y and vice versa - the mutual information is zero. In the other extreme, if x is a deterministic function of y then all information conveyed by x is shared with y. In this case the mutual information is the same as the uncertainty contained in x itself (or y itself), namely the entropy of x (or y).

Back to mutual information. Mutual information is obviously related to entropies,

$$I(X;Y) = S(X) - S(X|Y) = S(Y) - S(Y|X) = S(X) + S(Y) - S(X,Y). \tag{I.35}$$

which is illustrated in Fig. (2). It also possesses the following properties

$$I(X;Y) = I(Y;X) \text{ (symmetry)} \tag{I.36}$$
$$I(X;X) = S(X) \text{ (self-information)} \tag{I.37}$$

The conditional mutual information between X and Y given Z is

$$I(X;Y|Z) \doteq S(X|Z) - S(X|Y,Z) = \mathbb{E}_{P(x,y,z)}\left[\log\frac{P(x,y|z)}{P(x|z)P(y|z)}\right] \tag{I.38}$$

The entropy chain rule (I.31) when applied to the mutual information of $(X_1, \cdots, X_n) \sim P(x_1, \cdots, x_n)$ results in

$$I(X_n, \cdots, X_1; Y) = \sum_{i=1}^{n} I(X_i; Y | X_{i-1}, \cdots, X_1) \tag{I.39}$$

See Fig. (3) for the Venn diagram illustration of Eq. (I.39).

See [2] for extra discussions on entropy, mutual information and related.

3. Information Channel

Information channel, $X \to Y$, through the channel, $P(y|x)$. Information Channel Capacity is

$$C \doteq \max_{p(x)} I(X; Y). \tag{I.40}$$

Main - Shannon – theorem of the information theory (channel coding): Maximum rate at which we can communicate reliably over the channel is the information channel capacity C. More at the recitation.

4. Probabilistic Inequalities for Entropy and Mutual Information

Jensen's inequality. Let $f(X)$ be a convex function then

$$\mathbb{E}\left[f(X)\right] \geq f\left(\mathbb{E}\left[X\right]\right). \tag{I.41}$$

Here convexity of $f(x)$ on an interval $[a, b]$ means (reminder):

$$f(\lambda u + (1-\lambda)v) \leq \lambda f(u) + (1-\lambda) f(v), \quad \forall u, v \in [a, b], \quad 0 < \lambda < 1 \tag{I.42}$$

See Fig. (4) with the hint on the proof of the Jensen inequality.

Consequences of the Jensen inequality (for entropy and mutual information):

- (Information Inequality)

$$D(p\|q) \geq 0, \quad \text{with equality iff} p = q$$

- (conditioning reduces entropy)

$$S(X|Y) \leq S(X) \quad \text{with equality iff } X \text{ and } Y \text{ are independent}$$

- (Independence Bound on Entropy)

$$S(X_1, \cdots, X_n) \leq \sum_{i=1}^{n} S(X_i) \quad \text{with equality iff } X_i \text{are independent}$$

Another useful inequality [Log-Sum Theorem]

$$\sum_{i=1}^{n} a_i \log \frac{a_i}{b_i} \geq \left(\sum_{i=1}^{n} a_i\right) \log \frac{\sum_{i=1}^{n} a_i}{\sum_{i=1}^{n} b_i}, \tag{I.43}$$

with equality iff a_i/b_i is constant. Convention: $0 \log 0 = 0$, $a \log(a/0) = \infty$ if $a > 0$ and $0 \log 0/0 = 0$. Consequences of the Log-Sum theorem

- (Convexity of Relative Entropy) $D(p\|q)$ is convex in the pair p and q
- (Concavity of Entropy) For $X \sim p(x)$ we have $S(P) \doteq S_P(X)$ (notations are extended) is a concave function of $P(x)$.
- (Concavity of the mutual information in $P(x)$) Let $(X, Y) \sim P(x, y) = P(x)P(y|x)$. Then $I(X; Y)$ is a concave function of $P(x)$ for fixed $P(y|x)$.
- (Concavity of the mutual information in $P(y|x)$) Let $(X, Y) \sim P(x, y) = P(x)P(y|x)$. Then $I(X; Y)$ is a concave function of $P(y|x)$ for fixed $P(x)$.

We will see later (studying Graphical Models) why the convexity/concavity properties are useful.

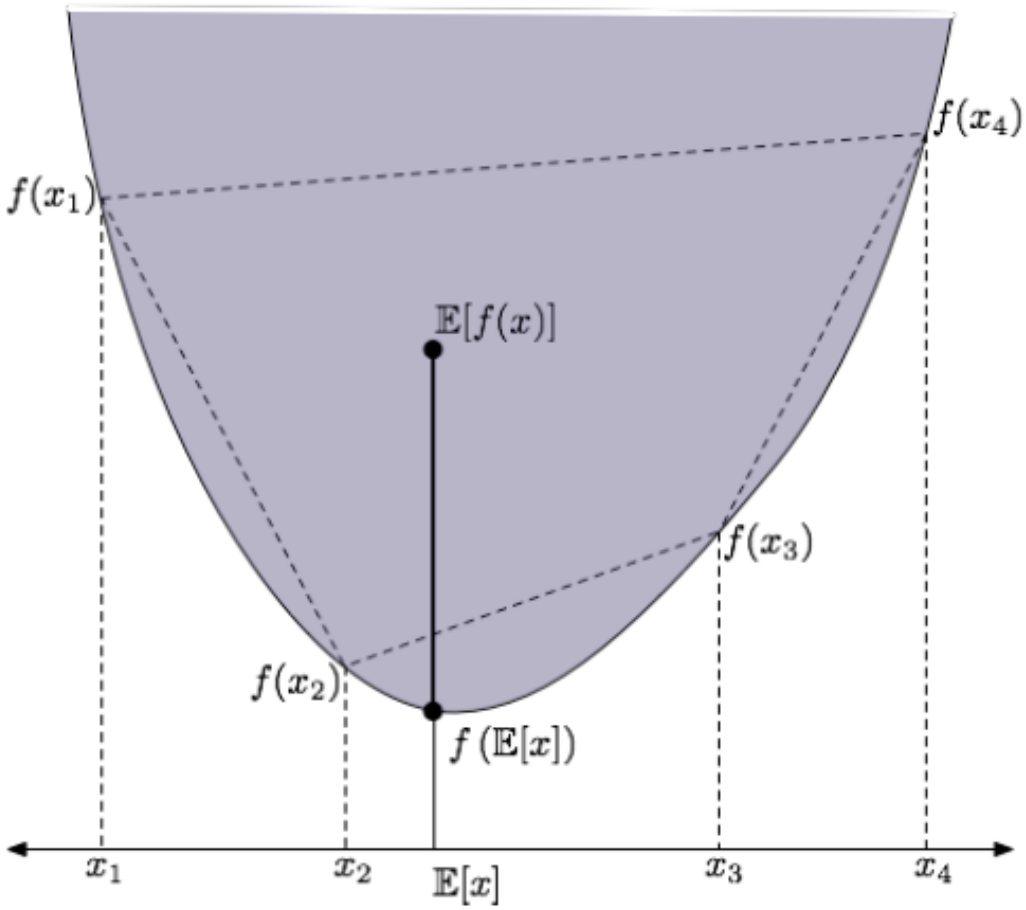

FIG. 4:

5. Recitation. Entropy, Mutual Information and Probabilistic Inequalities

II. THEME # 2. STOCHASTIC PROCESSES

A. Lecture #4: Markov Chains [discrete space, discrete time].

1. Transition Probabilities

So far we have studied random variables and events often assuming that these are i.i.d. = independent identically distributed. However, in real world we "jump" from one random state to another so that the transition depends on the original state. We may have a memory which last more than one jump, however there is also a big family of interesting random processes which do not have long memory - only current state influences where we jump to. This is the class of random processes described by Markov Chains (MCs).

MCs can be explained in terms of directed graphs, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where the set of vertices, $\mathcal{V} = (i)$, is asociated with the set of states, and the set of directed edges, $\mathcal{E} = (j \leftarrow i)$, correspond to possible transitions between the states. Note that we may also have "self-loops", $(i \leftarrow i)$ included in the set of edges. To make description complete we need to associate to each vertex a

transition probability, $p_{j\leftarrow i} = p_{ji}$ from the state i to the state j. Since p_{ji} is the probability, $\forall(j \leftarrow i) \in \mathcal{E}: \quad p_{ji} \geq 0$, and

$$\forall i: \quad \sum_{j:(j\leftarrow i)\in\mathcal{E}} p_{ji} = 1. \tag{II.1}$$

Then, the combination of $\mathcal{G}$ and $p \doteq (p_{ji}|(j \leftarrow i) \in \mathcal{E})$ defines a MC. Mathematically we also say that the tuple (finite ordered set of elements), $(\mathcal{V}, \mathcal{E}, p)$, defines the Markov chain. We will mainly consider in the following stationary Markov chains, i.e. these with p_{ji} constant - not changing in time. However, for many of the following statements/considerations generalization to the time-dependent processes is straightforward.

MC generates a random (stochastic) dynamic process. Time flows continuously, however as a matter of convenient abstraction we consider discrete times (and sometimes, actually quite often, events do happen discreetly). One uses $t = 0, 1, 2, \cdots$ for the times when jumps occur. Then a particular random trajectory/path/sample of the system will look like

$$i_1(0), i_2(1), \cdots, i_k(t_k), \quad \text{where} \quad i_1, \cdots, i_k \in \mathcal{V}$$

We can also generate many samples (many trajectories)

$$n = 1, \cdots, N: \quad i_1^{(n)}(0), i_2^{(n)}(1), \cdots, i_k^{(n)}(t_k), \quad \text{where} \quad i_1, \cdots, i_k \in \mathcal{V}$$

where N is the number of trajectories.

How does one relates the directed graph with weights (associated to the transition probabilities) to samples? The relation, actually, has two sides. The direct one - is about how one generates the samples. The samples are generated by advancing the trajectory from the current-time state flipping coin according to the transition probability p_{ij}. The inverse one - is about reconstructing characteristics of Markov chain from samples (or may be, even on the first place, verifying if the samples where indeed generated according to (rather restrictive) MC rules.

Now let us get back to the direct problem where a MC is described in terms of $(\mathcal{V}, \mathcal{E}, p)$. However, instead of characterizing the system in terms of the trajectories/paths/samples, we can pose the question following evolution of the "state probability vector", or simply the "state vector":

$$\forall i \in \mathcal{V}, \quad \forall t = 0, \cdots: \quad \pi_i(t+1) = \sum_{j:(i\leftarrow j)\in\mathcal{E}} p_{ij}\pi_j(t). \tag{II.2}$$

Here, $\pi(t) \doteq (\pi_i(t) \geq 0 | i \in \mathcal{V})$ is the vector built of components each representing probability for the system to be in the state i at the moment of time t. Thus, $\sum_{i\in\mathcal{V}} \pi_i = 1$. We can also rewrite Eq. (II.2) in the vector/matrix form

$$\pi(t+1) = p\pi(t), \tag{II.3}$$

where $\pi(t)$ the column/state and $p(t)$ is the transition-probability matrix, which satisfies the so-called "stochasticity" property (II.1). Sequential application of Eq. (II.3) results in

$$\pi(t+k) = p^k\pi(t), \tag{II.4}$$

and we are interested to analyze properties of p^k, characterizing the Markov chain acting for k sequential periods.

Let us first study it on the example of the simple MC shown in Fig. (5). In this case p^k is 2×2 which dependence on k is as follows

$$p^1 = \begin{pmatrix} 0.7 & 0.5 \\ 0.3 & 0.5 \end{pmatrix}, \quad p^2 = \begin{pmatrix} 0.64 & 0.6 \\ 0.36 & 0.4 \end{pmatrix}, \quad p^{10} \approx p^{100} \approx \begin{pmatrix} 0.625 & 0.625 \\ 0.375 & 0.375 \end{pmatrix}. \tag{II.5}$$

2. *Properties of Markov Chains*

The MC is **irreducible** if one can access any state from any state, formally

$$\forall i, j \in \mathcal{V}: \quad \exists k > 1, \quad \text{s.t.} \quad (p^k)_{ij} > 0. \tag{II.6}$$

Example #1 is obviously irreducible. However, if we replace $0.3 \to 0$ and $0.7 \to 1$ the MC becomes reducible – state 1 is not accessible from 2.

A state i has period k if any return to the state must occur in multiples of k. If $k = 1$ than the state is **aperiodic**. MC is **aperiodic** if all states are aperiodic. An irreducable MC only needs one aperiodic state to imply all states are aperiodic. Any

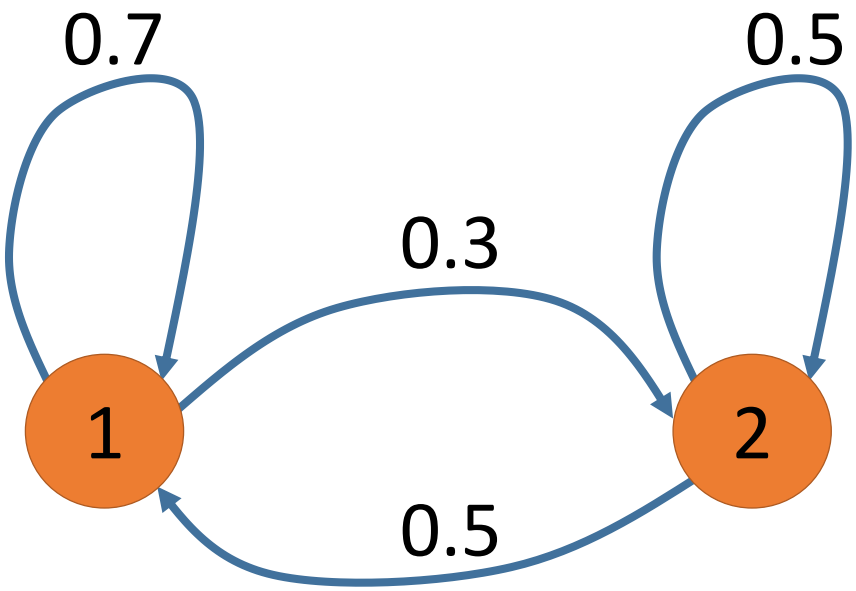

FIG. 5: Markov Chain (MC) - Example #1.

MC with at least one self-loop is aperiodic. Example #1 is obviously aperiodic. However, it becomes periodic with period two if the two self-loops are removed.

A state i is said to be transient if, given that we start in state i, there is a non-zero probability that we will never return to i. State i is recurrent if it is not transient. State i is **positive-recurrent** if the expected return time (to the state) is positive (this feature is important for infinite graphs).

A state is **ergodic** if the state is aperiodic and positive-recurrent. If all states in an irreducible MC are ergodic than the MC is said to be ergodic. A MC is ergodic if there is a finite number k_* such that any state can be reached from any other state in exactly k_* steps. For the example #1 $k_* = 2$. Note, that there are other (alternative) descriptions of ergodicity. Thus most intuitive one is: the MC is ergodic if it is irreducable and aperiodic. In this course we will not dwell much on the rich mathematical formalities and details, largely considering generic ergodic MC.

Practical consequence of ergodicity is that the teady state is unique and universal, the later refers to the fact that the steady state does not depend on the initial condition.

3. Steady State Analysis

Component-wise positive, normalized, π^*, is called stationary distribution (invariant measure) if

$$\pi^* = p\pi^* \tag{II.7}$$

An irreducible MC has a stationary distribution iff all of its states are positive recurrent. Solving Eq. (II.7) for the example # one finds

$$\pi^* = \begin{pmatrix} 0.625 \\ 0.375 \end{pmatrix}, \tag{II.8}$$

which is naturally consistent with Eq. (II.5). In general,

$$\pi^* = \frac{e}{\sum_i e_i}, \tag{II.9}$$

where e is the eigenvector with the eigenvalue 1. And how about other eigenvalues of the transition matrix?

4. Spectrum of the Transition Matrix & Speed of Convergence to the Stationary Distribution

Assume that p is diagonalizable (has $n = |p|$ linearly independent eigenvectors) then we can decompose p according to the following eigen-decomposition

$$p = U^{-1}\Sigma U \tag{II.10}$$

where $\Sigma = \text{diag}(\lambda_1, \cdots, \lambda_n)$, $1 = |\lambda_1| > \lambda_2 \geq |\lambda_3| \geq \cdots |\lambda_n|$ and U is the matrix of eigenvectors (each normalized to having an l_2 norm equal to 1) where each raw is a right eigenvector of p. Then

$$\pi^{(k)} = p^k\pi = (U^{-1}\Sigma U)^k\pi_0 = U^{-1}\Sigma^k U\pi_0. \tag{II.11}$$

Let us represent p_0 as an expansion over the normalized eigenvectors, $u_i, \cdots i = 1, \cdots, n$:

$$\pi = \sum_{i=1}^{n} a_i u_i. \tag{II.12}$$

Taking into account orthonormality of the eigenvectors one derives

$$\pi^{(k)} = \lambda_1 \left(a_1 u_1 + a_2 \left(\frac{\lambda_2}{\lambda_1}\right)^k u_2 + \cdots + a_n \left(\frac{\lambda_n}{\lambda_1}\right)^k u_n \right) \tag{II.13}$$

Since $\pi^{(k)}_{k\to\infty} \to \pi^* = u_1$, the second term on the rhs of Eq. (II.13) describes the rate of convergence of $\pi^{(k)}$ to the steady state – exponential in $\log(\lambda_1/\lambda_2$.

5. Reversible & Irreversible Markov Chains.

MC is called **reversible** if there exists π s.t.

$$\forall\{i,j\} \in \mathcal{E}: \quad p_{ji}\pi^*_i = p_{ij}\pi^*_j, \tag{II.14}$$

where $\{i,j\}$ is our notation for the undirected edge, assuming that both directed edges $(i \leftarrow j)$ and $(j \leftarrow i)$ are elements of the set $\mathcal{E}$. In physics this property is also called **Detailed Balance** (DB). If one introduces the so-called ergodicity matrix

$$Q \doteq (Q_{ji} = p_{ji}\pi^*_i | (j \leftarrow i) \in \mathcal{E}), \tag{II.15}$$

then DB translates into the statement that Q is symmetric, $Q = Q^T$. The MC for which the property does not hold is called **irreversible**. $Q - Q^T$ is nonzero, i.e. Q is asymmetric for reversible MC. An asymmetric component of Q is the matrix built from currents/flows (of probability). Thus for the case #1 shown in Fig. (5)

$$Q = \begin{pmatrix} 0.7*0.625 & 0.5*0.375 \\ 0.3*0.625 & 0.5*0.375 \end{pmatrix} = \begin{pmatrix} 0.4375 & 0.1875 \\ 0.1875 & 0.1875 \end{pmatrix} \tag{II.16}$$

Q is symmetric, i.e. even though $p_{12} \neq p_{21}$, there is still no flow of probability from 1 to 2 as the "population" of the two states, π^*_1 and π^*_2 respectively are different, $Q_{12} - Q_{21} = 0$. In fact, one can show that in the two node situation the steady state of the MC is always in DB.

6. Detailed Balance vs Global Balance. Adding cycles to accelerate mixing.

Note that if a steady distribution, π^*, satisfy the DB condition (II.14) for a MC, $(\mathcal{V}, \mathcal{E}, p)$, it will also be a steady state of another MC, $(\mathcal{V}, \mathcal{E}, \tilde{p})$, satisfying the more general Balance (or global balance) B-condition

$$\sum_{j:(j\leftarrow i)\in\mathcal{E}} \tilde{p}_{ji}\pi^*_i = \sum_{j:(i\leftarrow j)\in\mathcal{E}} \tilde{p}_{ij}\pi^*_j. \tag{II.17}$$

This suggests that many different MC (many different dynamics) may result in the same steady state. Obviously DB is a particular case of the B-condition (II.17).

The difference between DB- and B- can be nicely interpreted in terms of flows (think water) in the state space. From the hydrodynamic point of view reversible MCMC corresponds to an irrotational probability flows, while irreversibility relates to nonzero rotational part, e.g. correspondent to vortices contained in the flow. Putting it formally, in the irreversible case antisymmetric part of the ergodic flow matrix, $Q = (\tilde{p}_{ij}\pi^*_j|(i \leftarrow j))$, is nonzero and it actually allows the following cycle decomposition,

$$Q_{ij} - Q_{ji} = \sum_\alpha J_\alpha \left(C^\alpha_{ij} - C^\alpha_{ji}\right) \tag{II.18}$$

where index enumerates cycles on the graph of states with the adjacency matrices C^α. Then, J_α stands for the magnitude of the probability flux flowing over cycle .

One can use the cycle decomposition to modify MC such that the steady distribution stay the same (invariant). Of course, cycles should be added with care, e.g. to make sure that all the transition probabilities in the resulting $\tilde{p}$, are positive (stochasticity of the matrix will be guaranteed by construction). "Adding cycles" along with some additional tricks (e.g. lifting/repkication MC) may help to improve mixing, i.e. speed up convergence to the steady state — which is a very desirable property for sampling π^* efficiently. This and other features of MC will be discussed in details at the recitations on a three node example.

7. Recitation. Markov Chains: Detailed Balance. Mixing time.

B. Lecture #5. From Bernoulli Processes to Poisson Processes [discrete space, discrete & continuous time].

The two processes discussed here are the simplest dynamic random processes. Simplicity here is related to the fact that the processes are defined with the least number of characteristics. We will also focus on important properties of the processes, e.g. memorylessness, and also on working out interesting (and rather general) questions one may ask (and answer).

1. Bernoulli Process: Definition

Defined as a sequence of independent Bernoulli trials. At each trial

- P(success)=P(x=1)=p
- P(failure)=P(x=0)=1-p

Can be represented as a simple MC (two nodes + two self-loops). The sequence looks like $00101010001 = **S*S*S***S$. S here stands for "success".

Examples:

- Sequence of discrete updates – ups and downs (stock market)
- sequence of lottery wins
- arrivals of busses at a station checked every 1/5/? minutes

2. Bernoulli: Number of Successes

Number of k successes in n steps follows the binomial distribution

$$\forall k = 0, \cdots, n: \quad P(S = k) = \binom{n}{k} p^k (1-p)^{n-k} \tag{II.19}$$

$$\text{mean}: \quad \mathbb{E}[S] = np \tag{II.20}$$

$$\text{variance}: \quad var(S) = \mathbb{E}[(S - \mathbb{E}[S])^2] = np(1-p) \tag{II.21}$$

3. Bernoulli: Distribution of Inter-Arrivals

Call T_1 the number of trials till the first success (including the success event too). The Probability Mass Function (PMF)

$$t = 1, 2, \cdots : \quad P(T_1 = t) = p(1-p)^{t-1} \text{[Geometric PMF]} \tag{II.22}$$

The answer is product (thus memoreless) of the probabilities of $(t-1)$ failures and one success.

$$\text{mean}: \quad \mathbb{E}[T_1] = \frac{1}{p} \tag{II.23}$$

$$\text{variance}: \quad var(T_1) = \mathbb{E}[(T_1 - \mathbb{E}[T_1])^2] = \frac{1-p}{p^2} \tag{II.24}$$

More on the memoryless property. Given n, the future sequence $x_{n+1}, x_{n+2}, \cdots$ is also a Bernoulli process and is independent of the past. Moreover, suppose we observed the process for n times and no success has occurred. Then the PMF for the remaining arrival times is also geometric

$$P(T - n = k | T > n) = p(1-p)^{k-1} \tag{II.25}$$

And how about the k^{th} arrival? Let, y_k is the number of trials until k^{th} success (inclusive). T_k (previously introduced) is $T_k = Y_k - Y_{k-1}, \quad k = 2, 3, \cdots$ for k^{th} interarrival time. Also, $y_k = T_1 + T_2 + \cdots + T_k$. Then,

$$t = k, k+1, \cdots : \quad P(y_k = t) = \binom{t-1}{k-1} p^k (1-p)^{t-k} \text{[Pascal PMF]} \tag{II.26}$$

$$\text{mean}: \quad \mathbb{E}[y_k] = \frac{k(1-p)}{p^2} \tag{II.27}$$

$$\text{variance}: \quad var(y_k) = \mathbb{E}[(y_k - \mathbb{E}[y_k])^2] = \frac{k(1-p)}{p^2} \tag{II.28}$$

4. Poisson Process: Definition

Examples:

- all examples from the Bernoulli case in continuous time
- e-mails arrivals with infrequent check
- high-energy beams collide at a high frequency (10 MHz) with a small chance of good event
- radioactive decay of a nucleus with the trial being to observe a decay within a small interval
- spin flip in a magnetic field

Two way of thinking of it. One as of a continuous version of the Bernoulli process. Another through random time intervals between successes.

Start from the first one. Intervals become infinitesimally small and we replace probabilities (of success) by probability densities (per unit time). Let $P(k, \tau)$ be the probability of k arrivals in an interval of duration τ. We assume that

- number of arrivals in disjoint time intervals are independent
- for very small δ (regularization)

$$P(k, \delta) \approx \begin{cases} 1 - \lambda\delta & k = 0 \\ \lambda\delta & k = 1 \\ 0 & k > 0 \end{cases} \tag{II.29}$$

- λ is the arrival rate of the process

on the relations between Bernoulli and Poisson (assuming $n = t/\delta$ and $p = \lambda\delta$):

Bernoulli	Possion
arrival probability in each time slot $= p$	arrival probability in each δ-interval $= \lambda\delta$
number of arrivals in t intervals	number of successes in n time slots

5. Poisson: Inter-arrival Time

Probability density of the first arrival, y_1:

$$p_{Y_1}(y) = \lambda \exp(-\lambda y), \quad y \geq 0 \quad \text{[exponential]}$$

Then

$$P(Y_1 \leq y) = 1 - P(0, y) = 1 - \int_0^y dy' p_{Y_1}(y') = 1 - \exp(-\lambda y)$$

Like Bernoulli, the Poisson keeps the two key properties

- **Fresh Start Property**: the time of the next arrival is independent from the past
- **Memoryless property:** suppose we observe the process for t seconds and no success occurred. Then the density of the remaining time of arrival is exponential.

By extension (taking limit), for the probability density of time of the k^{th} arrival one derives

$$p_{Y_k}(y) = \frac{\lambda^k y^{k-1} \exp(-\lambda y)}{(k-1)!}, \quad y > 0 \quad \text{(Erlang "of order" k)} \tag{II.30}$$

To conclude

	Bernoulli	Poisson
Times of Arrival	Discrete	Continuous
Arrival Rate	p/per trail	λ/unit time
PMF of Number of arrivals	Binomial	Poisson
PMF of Interarrival Time	Geometric	Exponential
PMF of k^{th} Arrival Time	Pascal	Erlang

6. Poisson: Number of arrivals in a t-intervals as $n \to \infty$

$$\binom{n}{k} p^k (1-p)^{n-k} = \binom{n}{k} \left(\frac{\lambda t}{n}\right)^k (1 - \frac{\lambda t}{n})^{n-k} \quad \text{[Binomial]} \tag{II.31}$$

$$= \underbrace{\frac{n!}{(n-k)! n^k}}_{\to 1} \frac{(\lambda t)^k}{k!} \underbrace{\left(1 - \frac{\lambda t}{n}\right)^n}_{\to \exp(-\lambda t)} \underbrace{\left(1 - \frac{\lambda t}{n}\right)^{-k}}_{\to 1} \quad \text{(reoder terms)} \tag{II.32}$$

$$= P(N = k) = P(k = \tau) = \frac{(\lambda t)^k}{k!} e^{-\lambda t} \quad \text{[Poisson}(\lambda\tau)\text{]} \tag{II.33}$$

$$\text{mean :} \quad \mathbb{E}[N] = \lambda t \tag{II.34}$$

$$\text{variance :} \quad var(N) = \mathbb{E}[(N - \mathbb{E}[N])^2] = \lambda\tau \tag{II.35}$$

7. Merging and Splitting Processes

Most important feature shared by Bernoulli and Poisson processes is their invariance with respect to mixing and splitting. We will show it on an example of Poisson process but the same applies to Bernoulli process.

Merging: Let $N_1(t)$ and $N_2(t)$ be two independent Poisson processes with rates λ_1 and λ_2 respectively. Let us define $N(t) = N_1(t) + N_2(t)$. This rando process is derived combining the arrivals as shown in Fig. (**??**). The claim is that $N(t)$ is the Poisson process with the rate $\lambda_1 + \lambda_2$. To see it we first note that $N(0) = N_1(0) + N_2(0) = 0$. Next, since $N_1(t)$ and $N_2(t)$ are independent and have independent increments their sum also have an independent increment. Finally, consider an interval of length τ, $(t, t + \tau]$. Then the number of arrivals in the interval are Poisson$(\lambda_1\tau)$ and Poisson$(\lambda_2\tau)$ and the two numbers are independent. Therefore the number of arrivals in the interval associated with $N(t)$ is Poisson$((\lambda_1 + \lambda_2)\tau)$ - as sum of two

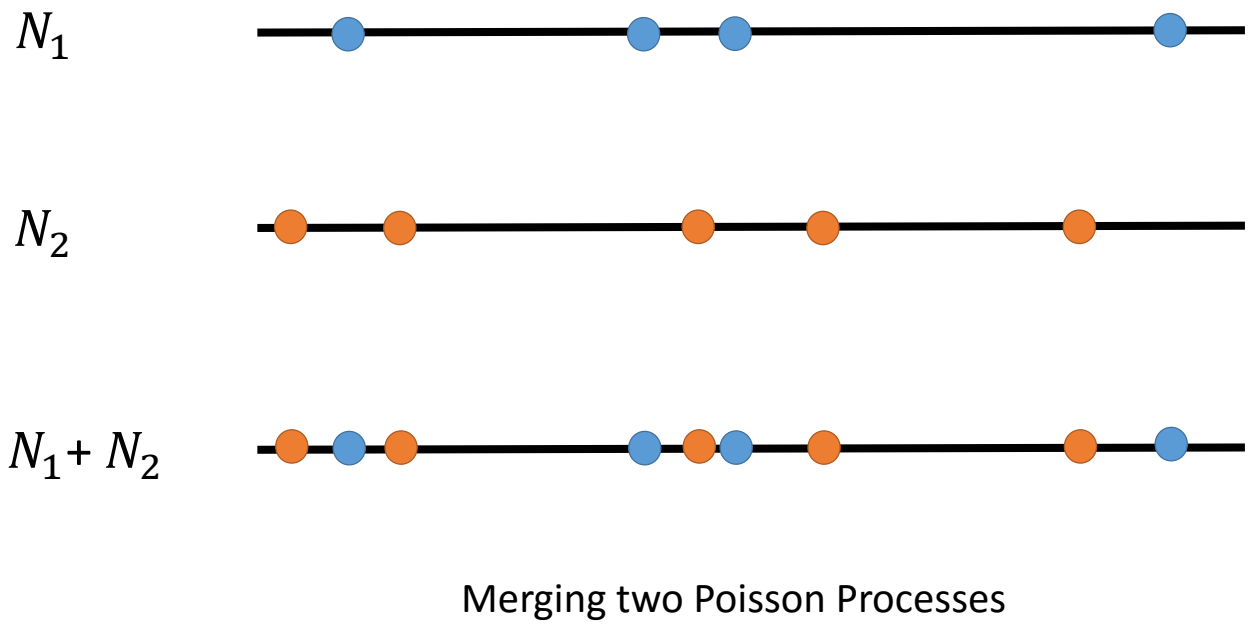

FIG. 6: Merging two Poisson processes.

independent Poisson random variables. We can obviously generalize the statement to a sum of many Poisson processes. Note that in the case of Bernoulli process the story is identical provided that collision is counted as one arrival.

Splitting: Let $N(t)$ be a Poisson process with rate λ. Here, we split $N(t)$ into $N_1(t)$ and $N_2(t)$ where the splitting is decided by coin tossing (Bernoulli process) - when an arrival occur we toss a coin and with probability p and $1-p$ add arrival to N_1 and N_2 respectively. The coin tosses are independent of each other and are independent of $N(t)$. Then, the following statements can be made

- N_1 is a Poisson process with rate λp.
- N_2 is a Poisson process with rate $\lambda(1-p)$.
- N_1 and N_2 are independent, thus Poisson.

8. Recitation. Examples of Bernoulli & Poisson Processes

+

C. Lecture #6. Monte-Carlo Algorithms: General Concepts and Direct Sampling.

This lecture should be read in parallel with the respective IJulia notebook file. Monte-Carlo (MC) methods refers to a broad class of algorithms that rely on repeated random sampling to obtain results. Named after Monte Carlo -the city- which once was the capital of gambling (playing with randomness). The MC algorithms can be used for numerical integration, e.g. computing weighted sum of many contributions, expectations, marginals, etc. MC can also be used in optimization.

Sampling is a selection of a subset of individuals/configurations from within a statistical population to estimate characteristics of the whole population.

There are two basic flavors of sampling. Direct Sampling MC - mainly discussed in this lecture and Markov Chain MC. DS-MC focuses on drawing **independent** samples from a distribution, while MCMC draws correlated (according to the underlying Markov Chain) samples.

Let us illustrate both on the simple example of the 'pebble' game - calculating the value of π by sampling interior of a circle.

1. Direct-Sampling by Rejection vs MCMC for 'pebble game'

In this simple example we will construct distribution uniform within a disk from the distribution uniform over a square containing the circle. We will use direct product of two rand() to generate samples within the square and then simply reject samples which are not in the interior of the disk.

In the respective MCMC we build a sample (parameterized by a pair of coordinates) by taking previous sample and adding some random independent shifts to both variables, also making sure that when the sample crosses a side of the square it reappears on the opposite side. The sample "walks" the square, but we count (to compute area of the disk) only for samples which are within the disk (rejection again).

See IJulia notebook for an illustration.

2. Direct Sampling by Mapping

Direct Sampling by Mapping consists in application of the deterministic function to samples from a distribution you know how to sample from efficiently. The method is exact, i.e. it produces independent random samples distributed according to the new distribution. (We will discuss formal criteria for independence in the next lecture.)

For example, suppose we want to generate exponential samples, $y_i \sim \rho(y) = \exp(-y)$ – one dimensional exponential distribution over $[0,\infty]$, provided that one-dimensional uniform oracle, which generates independent samples, x_i from $[0,1]$, is available. Then $y_i = -\log(x_i)$ generates desired (exponentially distributed) samples.

Another example of DS MS by mapping is given by the Box-Miller algorithm which is a smart way to map two-dimensional random variable distributed uniformly within a box to the two-dimensional Gaussian (normal) random variable:

$$\int_{-\infty}^{\infty}\frac{dxdy}{2\pi}e^{-(x^2+y^2)/2}=\int_0^{2\pi}\frac{d\varphi}{2\pi}\int_0^{\infty}rdr\,e^{-r^2/2}=\int_0^{2\pi}\frac{d\varphi}{2\pi}\int_0^{\infty}dz\,e^{-z}=\int_0^1 d\theta\int_0^1 d\psi=1.$$

Thus, the deisred mapping is $(\psi,\theta)\to(x,y)$, where $x=\sqrt{-2\log\psi}\cos(2\pi\Theta)$ and $y=\sqrt{-2\log\psi}\sin(2\pi\theta)$.

See IJulia notebook for numerical illustrations.

3. Direct Sampling by Rejection (another example)

Let us now show how to get positive Gaussian (normal) random variable from an exponential random variable through rejection. We do it in two steps

- First, one samples from the exponential distribution: $x\sim\rho_0(x)=\begin{cases}e^{-x} & x>0,\\ 0 & \text{otherwise}\end{cases}$
- Second, aiming to get a sample from the positive half of Gaussian, $x\sim\rho_0(x)=\begin{cases}\sqrt{2/\pi}\exp(-x^2/2) & x>0,\\ 0 & \text{otherwise}\end{cases}$, one accepts the generated sample with the probability

$$p(x)=\frac{1}{M}\sqrt{2/\pi}\exp(x-x^2/2)$$

where M is a constant which should be larger than, $\max(\rho(x)/\rho_0(x))=\sqrt{2/\pi}\,e^{1/2}\approx 1.32$, to guarantee that $p(x)\le 1$ for all $x>0$.

Note that the rejection algorithm has the advantage that it can be applied even when the probability densities are known only up to a multiplicative constant. (We will discuss issues related to this constant, also called in the multivariate case the partition function extensively)

4. Importance Sampling

As we saw MC is useful for computing sums/integrals/expectations. Suppose we want to compute an expectation of a function, $f(x)$, over the distribution, $\rho(x)$, i.e. $\int dx\rho(x)f(x)$, in the regime where $f(x)$ and $\rho(x)$ are concentrated around very different x. In this case overlap of f and $\rho(x)$ is small and as a result a lot of MC samples will be 'wasted'.

Importance Sampling is the method which helps to fix the small-overlap problem. It is based on adjusting the distribution function from $\rho(x)$ to $\rho_a(x)$ and then utilizing the following obvious formula

$$\mathbb{E}_\rho[f(x)] = \int dx \rho(x) f(x) = \int dx \rho_a(x) \frac{f(x)\rho(x)}{\rho_a(x)} = \mathbb{E}_{\rho_a}\left[\frac{f(x)\rho(x)}{\rho_a(x)}\right]$$

Consider an illustrative DS example, $\rho(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right)$ and $f(x) = \exp\left(-\frac{(x-4)^2}{2}\right)$, vs IS with the proposal distribution, $\rho_a(x) = \frac{1}{\sqrt{\pi}} \exp\left(-(x-2)^2\right)$.

This example is illustrated in the the IJulia notebook

The simple example shows that we are clearly wasting samples with DS. Note one big problem with IS. In the real (multi-dimensional) cases we typically do not have a good guess for $\rho_a(x)$. One way of solving this problem is to search for good $\rho_a(x)$ adaptively. A sophisticated adaptive importance sampling package is available at `https://pypi.python.org/pypi/pypmc/1.0`.

See

- `https://statmechalgcomp.wikispaces.com/`
- `http://www.math.nyu.edu/faculty/goodman/teaching/Monte_Carlo/direct_sampling.ps`
- `http://www.cs.berkeley.edu/~jordan/courses/260-spring10/lectures/lecture17.pdf`

for additional material/discussions of DS-MC.

5. Direct Brut-force Sampling

This algorithm relies on availability of the uniform sampling algorithm from $[0, 1]$, rand(). One splits the $[0, 1]$ interval into pieces according to the weights of all possible states and then use rand() to select the state. The algorithm is impractical as it requires keeping in the memory information about all possible configurations. The use of this construction is in providing the bench-mark case useful for proving independence of samples.

6. Direct Sampling from a multi-variate distribution with a partition function oracle

So far we have discussed sampling from a single-valued distribution. Suppose we have an oracle capable of computing the partition function (normalization) for the multivariate probability itself and also for any of the marginal probabilities. (Notice that we are not asking about complexity of the oracle, at least not for now.) Does it give us the power to generate independent samples?

We get affirmative answer to this question through the following **decimation** algorithm generating independent sample $x \sim P(x)$, where $x \doteq (x_i | i = 1, \cdots, N)$:

Algorithm 1 Decimation Algorithm

Input: $P(x)$ (expression). Partition function oracle.

$x^{(d)} = \varnothing; \quad I = \varnothing$
while $|I| < N$ **do**
  Pick i at random from $\{1, \cdots, N\} \setminus I$.
  $x^{(I)} = (x_j | j \in I)$
  Compute $P(x_i|x^{(d)}) \doteq \sum_{x \setminus x_i; x^{(I)} = x^{(d)}} P(x)$ with the oracle.
  Generate random $x_i \sim P(x_i|x^{(d)})$.
  $I \cup i \leftarrow I$
  $x^{(d)} \cup x_i \leftarrow x^{(d)}$
end while

Output: $x^{(\text{dec})}$ is an independent sample from $P(x)$.

Validity of the algorithm follows from the following exact representation for the joint distribution function via a chain of ordered conditional distribution function (chain rule for distribution):

$$P(x_1, \cdots, x_n) = P(x_1)P(x_2|x_1)P(x_3|x_1, x_2) \cdots P(x_n|x_1, \cdots, x_{n-1}). \tag{II.36}$$

(The chain rule follows directly from the Bias rule. Notice also that ordering of variables in the chain rule is arbitrary.) One way of proving that an algorithm produce independent sample is to show that the algorithm outcome is equivalent to another algorithm for which the independence is already proven. The benchmark algorithm we can use to state that the Decimation algorithm (1) produces independent samples is the brute-force sampling algorithm described in the beginning of the lecture. The crucial point here is in splitting the $[0,1]$ interval hierarchically, first according to $P(x_1)$, then subdividing pieces for different x_1 according to $P(x_2,x_1)$, etc. This guidanken experiment will result in the desired proof.

In general efforts of the oracle (for computing the partition function) are exponential. However in some special case the partition function can be computed efficiently (polynomially in the number of steps). For example this is the case for (glassy) Ising model without magnetic field over planar graph. See `http://surface.syr.edu/cgi/viewcontent.cgi?article=1176&context=phy` and references there in for details.

7. Ising Model

Let us digress and consider the Ising model – example of a larger class of important/interesting multi-variate statistics often referred to (in theoretical engineering) as Graphical Models (GM). We will study GM in even more details later. Consider a system of spins or pixels (binary variables) on a graph, $\mathcal{G}=(\mathcal{V},\mathcal{E})$, where $\mathcal{V}$ is a set of nodes/vertices and $\mathcal{E}$ is the set of edges. The graph may be 1d chain, tree, 2d lattice ... or any other graph. (The cases of regular lattices are prevalent in physics, while graphs of interest for engineering are, generally, richer.) Consider a binary variable, residing at every node of the graph, $\forall i\in\mathcal{V}:\ \sigma_i=\pm 1$, we call them "spins". If there are N spins in the system, 2^N is the number of possible configuration of spins — notice exponential scaling with N, meaning, in particular that just counting the number of configurations is "difficult". If we would be able to do it in algebraic/polynomial number of steps, we would call it "easy", or rather "theoretically easy", while the practically easy case - which is the goal – would correspond to the case when "complexity" of, say counting, would be $O(N)$ - linear in N at $N\to\infty$. (Btw $o(N)$ is the notation used to state that the behavior is actually slower than $O(N)$, say $\sim\sqrt{N}$ at $N\to\infty$, i.e. asymptotically $o(N)\ll O(N)$.) In magnetism (field of physics where magnetic materials are studied) probability of a spin configuration (vector) is

$$p(\sigma)=\frac{\exp\left(-\beta E(\sigma)\right)}{Z},\quad E(\sigma)=-\frac{1}{2}\sum_{\{i,j\}\in\mathcal{E}}\sigma_i J_{ij}\sigma_j+\sum_{i\in\mathcal{V}}h_i\sigma_i, \tag{II.37}$$

$$Z=\sum_{\sigma}\exp\left(-\beta E(\sigma)\right). \tag{II.38}$$

$E(\sigma)$ is energy of the given spin configuration, σ. The first term in $E(\sigma)$ is pair-wise (wrt nodal spins), spin exchange/interaction term. The last term in $E(\sigma)$ stands from (potentially node dependent) contribution of the magnetic field, $h=(h_i|i\in\mathcal{V})$ on individual spins. Z is the partition function - weighted sum of the spin configurations. Introduced to enforce the normalization condition, $\sum_\sigma p(\sigma)=1$. For a general graph, arbitrary values of J and h, Z is the difficult object to compute, i.e. complexity is $O(2^N)$. (Notice that for some special cases, such as when graph is a tree, or when graph is planar and $h=0$, the computations become easy.) Moreover computing other important characteristics, such as the most probable configuration of spins

$$\sigma_{ML}=\arg\max_{\sigma}p(\sigma), \tag{II.39}$$

also called Maximum Likelihood and Ground State in information sciences and physics respectively, or probability of observing a particular node in state σ_i (can be $+$ or $-$)

$$p_i(\sigma_i)=\sum_{\sigma\setminus\sigma_i}p(\sigma), \tag{II.40}$$

are also difficult problems. (Wrt notations – $\arg\max$ - pronounced argmax - stands for particular σ at which the maximum in Eq. (II.39) is reached. $\sigma\setminus\sigma_i$ in the argument of the sum in Eq. (II.40) means that we sum over all σ consistent with the fixed value of σ_i at the node i.)

D. Lecture #7. Markov-Chain Monte-Carlo.

Markov Chain Monte Carlo (MCMC) methods belong to the class of algorithms for sampling from a probability distribution based on constructing a Markov chain that converges to the target steady distribution.

Examples and flavors of MCMC are many (and some are quite similar) – heat bath, Glauber dynamics, Gibbs sampling, Metropolis Hastings, Cluster algorithm, Warm algorithm, etc – in all these cases we only need to know transition probability

between states while the actual stationary distribution may be not known or, more accurately, known up to the normalization factor, also called the partition function. Below, we will discuss in details two key examples: Gibbs sampling & Metropolis-Hastings.

1. Gibbs Sampling

Assume that the direct sampling is not feasible (because there are too many variables and computations are of "exponential" complexity — more on this latter). The main point of Gibbs sampling is that given a multivariate distribution it is simpler to sample from a conditional distribution than to marginalize by integrating over a joint distribution. Then we create a chain: start from a current sample of the vector x, pick a component at random, compute probability for this component/variable conditioned to the rest, and sample from this conditional distribution. (The conditional distribution is for a simple component and thus it is easy.) We continue the process till convergence, which can be identified (empirically) by checking if estimation of the histogram or observable(s) stopped changing.

Algorithm 2 Gibbs Sampling

Input: Given $p(x_i|x_{\sim i} = x \setminus x_i), \quad \forall i \in \{1, \cdots, N\}$. Start with a sample $x^{(t)}$.

loop Till convergence
  Draw an i.i.d. i from $\{1, \cdots, N\}$.
  Generate a random $x_i \sim p\left(x_i|x^{(t)}_{\sim i}\right)$.
  $x^{(t)}_i = x_i$.
  $\forall j \in \{1, \cdots, N\} \setminus i: \quad x^{(t)}_j \leftarrow x^{(t)}_j$.
  Output $x^{(t+1)}$ as the next sample.
end loop

Question (specific): Does the Gibbs sampling obey the Detailed Balance? Show it.

Question (generic): if one wants to prove convergence of an MCMC, how one can do it? (What is the expectation/average to study?)

2. Metropolis-Hastings Sampling

Metropolis-Hastings sampling is an MCMC method which explores efficiently the detailed balance, i.e. reversibility of the underlying Markov Chain. The algorithm also uses sampling from the conditional probabilities and smart use of the rejection strategy. Assume that the probability of any state x from which one wants to sample (call it the target distribution) is explicitly known up to the normalization constant, Z, i.e. $p(x) = \pi(x)/Z$, where $Z = \sum_x \pi(x)$. Let us also introduce the so-called proposal distribution, $p(x'|x)$, and assume that drawing a sample proposal x' from the current sample x is (computationally) easy.

Algorithm 3 Metropolis-Hastings Sampling

Input: Given $\pi(x)$ and $p(x'|x)$. Start with a sample x_t.

loop Till convergence
  Draw a random $x' \sim p(x'|x_t)$.
  Compute $\alpha = \frac{p(x_t|x')\pi(x')}{p(x'|x_t)\pi(x_t)}$.
  Draw random $\beta \in U([0, 1])$, uniform i.i.d. from $[0, 1]$.
  if $\beta < \min\{1, \alpha\}$ **then**
    $x_t \leftarrow x'$ [accept]
  else
    x' is ignored [reject]
  end if
  x_t is recordered as a new sample
end loop

Note that the Gibbs sampling previously introduced can be considered as the Metropolis-Hastings without rejection (thus it is a particular case).

Exercise: Arguing in terms of transitions between states show that the algorithm maintains the DB.

The proposals (conditional probabilities) may vary. Details are critical (change mixing time), especially for large system. There is a (heuristic) rule of thumb: **lower bound on number of iterations of MH**. If the largest distance between the states is L, the MH will mix in time

$$T \approx (L/\varepsilon)^2 \tag{II.41}$$

where ε is the typical step size of the random walk.

Question: How can one reason the quadratic behavior in Eq. (II.41)? What would acceleration from quadratic to linear mean? (Diffusive vs ballistic regime.)

Mixing may be extremely slow if the proposal distribution is not selected carefully. Let us illustrate how slow MCMC can be on a simple example. (See Section 29 of the McKay book for extra details.) Consider the following target distribution over N states

$$\pi(x) = \begin{cases} 1/N & x \in \{0, \cdots, N-1\} \\ 0 & \text{otherwise} \end{cases} \tag{II.42}$$

and proposal distribution over $N+2$ states (extended by -1 and N)

$$p(x'|x) = \begin{cases} 1/2 & x' = x \pm 1 \\ 0 & \text{otherwise} \end{cases} \tag{II.43}$$

Notice that the rejection can only occur when the proposed state is $x' = -1$ or $x' = N$.

A more sophisticated example of the Glauber algorithm (version of MH) on the example of the Ising Model is to be discussed next.

3. Glauber Sampling of Ising Model

Let us return to the special version of MH developed specifically for the Ising model - the Glauber dynamics/algorithms:

Algorithm 4 Glauber Sampling

Input: Ising model on a graph. Start with a sample σ

loop Till convergence
  Pick a node i at random.
  $-\sigma_i \leftarrow \sigma_i$
  Compute $\alpha = \exp\left(\sigma_i \left(\sum_{j \in \mathcal{V}:\{i,j\} \in \mathcal{E}} J_{ij}\sigma_j - 2h_i\right)\right)$.
  Draw random $\beta \in U([0,1])$, uniform i.i.d. from $[0,1]$.
  if $\alpha < \beta < 1$ **then**
    $-\sigma_i \leftarrow \sigma_i$ [reject]
  end if
  Output: σ as a sample
end loop

Question: What is the proposal distribution turning the MH sampling into the Glauber sampling (for the Ising model)?

Exercise [Advanced]: Consider running parallel dynamics, based on the Glauber algorithm, i.e. at every moment of time update all variables in parallel according to the Glauber Sampling rule applied to the previous state. What is the resulting stationary distribution? Is it different from the Ising model? Does the algorithm satisfy the DB conditions?

For useful additional reading on sampling and computations for the Ising model see `https://www.physik.uni-leipzig.de/~janke/Paper/lnp739_079_2008.pdf`. MCMC recitation will focus on discussion of the Glauber algorithm.

4. Exactness and Convergence

MCMC algorithm is called (casually) exact if one can show that the generated distribution "converges" to the desired stationary distribution. However, "convergence" may mean different things.

The strongest form of convergence – called **exact independence test** (warning - this is our 'custom' term) – states that at each step we generate an independent sample from the target distribution. To prove this statement means to show that empirical correlation of the consecutive samples is zero in the limit when Nnumber of samples $\to \infty$:

$$\lim_{N\to+\infty} \frac{1}{N} \sum_{n=0}^{N} \sum_{n=1}^{N} f(x_n) g(x_{n-1}) \to \mathbb{E}\left[f(x)\right] \mathbb{E}\left[g(x)\right], \tag{II.44}$$

where $f(x)$ and $g(x)$ are arbitrary functions (however such that respective expectations on the rhs of Eq. (**??**) are well-defined).

A weaker statement – call it **asymptotic convergence** – suggests that in the limit of $N \to \infty$ we reconstruct the target distribution (and all the respective existing moments):

$$\lim_{N\to+\infty} \frac{1}{N} \sum_{n=0}^{N} \sum_{n=1}^{N} f(x_n) \to \mathbb{E}\left[f(x)\right], \tag{II.45}$$

where $f(x)$ is an arbitrary function such that the expectation on the rhs is well defined.

Finally, the weakest statement – call it **parametric convergence** – corresponds to the case when one arrives at the target estimate only in a special limit with respect to a special parameter. It is common, e.g. in statistical/theoretical physics and computer sicence, to study the so-called thermodynamic limit, where the number of degrees of freedom (for example number of spins/variables in the Ising model) becomes infinite:

$$\lim_{s\to s_*} \lim_{N\to+\infty} \frac{1}{N} \sum_{n=0}^{N} \sum_{n=1}^{N} f_s(x_n) \to \mathbb{E}\left[f_{s_*}(x)\right]. \tag{II.46}$$

For additional math (but also intuitive as written for physicists) reading on the MCMC (and in general MC) convergence see `http://statweb.stanford.edu/~cgates/PERSI/papers/mixing.pdf` and also [3].

5. Exact Monte Carlo Sampling (Did it converge yet?)

(This part of the lecture is a bonus material - we discuss it only if time permits.)

The material follows Chapter 32 of D.J.C. MacKay book. Some useful set of modern references and discussions are also available at `http://dimacs.rutgers.edu/~dbwilson/exact/`.

As mentioned already the main problem with MCMC methods is that one need to wait (and sometimes for too long) to make sure that the generated samples (from the target distribution) are i.i.d. If one starts to form a histogram (empirical distribution) too early it will deviate from the target distribution. One important question in this regards is: For how long shall one run the Markov Chain before it has 'converged'? To answer this question (prove) it is very difficult, in many cases not possible. However, there is a technique which allows to check the **exact convergence**, for some cases, and do it on the fly - as we run MCMC.

This smart technique is the Propp-Wilson exact sampling method, also called **coupling from the past**. The technique is based on a combination of three ideas:

- The main idea is related to the notion of **trajectory coalescence**. Let us observe that if starting from different initial conditions MCMC chains share a single random number generator, then their trajectories in the phase space can coalesce; and having coalesced, will not separate again. This is clearly an indication that the initial conditions are forgotten.

  Will running All the initial conditions forward in time till coalescence generate exact sample? Apparently not. One can show (sufficient to do it for a simple example) that the point of coalescence does not represent an exact sample.

- However, one can still achieve the goal by **sampling from a time T_0 in the past**, up to the present. If the coalescence has occurred the present sample is an unbiased sample; and if not we restart the simulation from the time T_0 further into the past, reusing the same random numbers. The simulation is repeated till a coalescence occur at a time before the present. One can show that the resulting sample at the present is exact.

- One problem with the scheme is that we need to test it for all the initial conditions - which are too many to track. Is there a way to **reduce the number of necessary trials**. Remarkably, it appears possible for sub-class of probabilistic models the so-called **'attractive'** models. Loosely speaking and using 'physics' jargon - these are **'ferromagnetic'** models - which are the models where for a stand alone pair of variables the preferred configuration is the one with the same values of the two variables. In the case of attractive model monotonicity (sub-modularity) of the underlying model suggests that the paths do not cross. This allows to only study limiting trajectories and deduce interesting properties of all the other trajectories from the limiting cases.

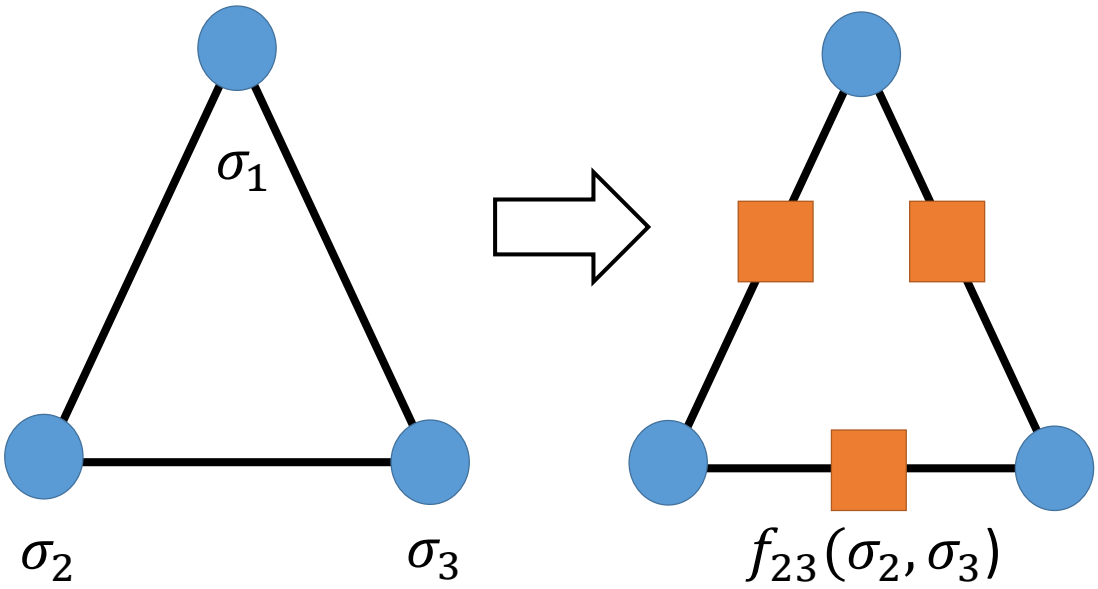

FIG. 7: Factor Graph Representation for the (simple case) with pair-wise factors only. In the case of the Ising model: $f_{12}(\sigma_1,\sigma_2) = \exp\left(-J_{12}\sigma_1\sigma_2 + h_1\sigma_1 + h_2\sigma_2\right)$.

III. THEME # 3: GRAPHICAL MODELS

A. Lecture #8. Exact & Approximate Inference.

This lecture and the next lecture largely follow the materials of `https://dl.dropboxusercontent.com/u/8802428/Talks_on_Line/Alg/SPA2011.pdf`.

1. *From Ising Model to (Factor) Graphical Models*

Brief reminder of what we have learned so far about the Ising Model. It is fully described by Eqs. (II.37,II.38). The weight of a 'spin' configuration is given by Eq. (II.37). Let us not pay much of attention for now to the normalization factor Z and observe that the weight is nicely factorized. Indeed, it is a product of pair-wise terms. Each term describes 'interaction' between spins. Obviously we can represent the factorization through a graph. For example, if our spin system consists of only three spins connected to each other, then the respective graph is a triangle. Spins are associated with nodes of the graphs and 'interactions', which may also be called (pair-wise) factors, are associated with edges.

It is useful, for resolving this and other factorized problems, to introduce a bit more general representation — in terms of graphs where both factors and variables are associated with nodes/vertices. The transformation to the factor-graph representation for the three spin example is shown in Fig. (7).

Ising Model, as well as other models discussed later in the lectures, can thus be stated in terms of the general factor-graph framework/model

$$P(\sigma) = Z^{-1}\prod_{a\in\mathcal{V}_f} f_a(\sigma_a), \quad \sigma_a \doteq (\sigma_i | i\in\mathcal{V}_n, \quad (i,a)\in\mathcal{E}), \tag{III.1}$$

where $(\mathcal{V}_f,\mathcal{V}_n,\mathcal{E})$ is the bi-partite graph build of factors and nodes.

The factor graph language (representation) is more general. We will see it next - discussing another interesting problem from Information Theory - decoding of error-correction codes.

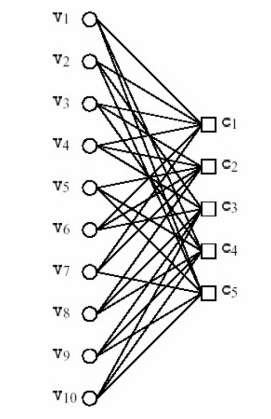

FIG. 8: Tanner graph of a linear code, represented with $N = 10$ bits, $M = 5$ checks, and $L = N - M = 5$ information bits. This code selects 2^5 codewords from $2^1 0$ possible patterns. This adjacency, parity-check matrix of the code is given by Eq. (III.2).

2. *Decoding of Graphical Codes as a Factor Graph problem*

First, let us discuss decoding of a graphical code. (Our description here is terse, and we advise interested reader to check the book by Richardson and Urbanke [8] for more details.) A message word consisting of L information bits is encoded in an N-bit long code word, $N > L$. In the case of binary, linear coding discussed here, a convenient representation of the code is given by $M \geq N - L$ constraints, often called parity checks or simply, checks. Formally, $\boldsymbol{\varsigma} = (\varsigma_i = 0, 1 | i = 1, \cdots, N)$ is one of the 2^L code words iff $\sum_{i\sim\alpha} \varsigma_i = 0 \ (\mathrm{mod}\ 2)$ for all checks $\alpha = 1, \cdots, M$, where $i \sim \alpha$ if the bit if the bit i contributes the check α, and $\alpha \sim i$ will indicate that the check α contains bit i. The relation between bits and checks is often described in terms of the $M \times N$ parity-check matrix $\boldsymbol{H}$ consisting of ones and zeros: $H_{i\alpha} = 1$ if $i \sim \alpha$ and $H_{i\alpha} = 0$ otherwise. The set of the codewords is thus defined as $\Xi^{(\mathrm{CW})} = (\varsigma | \boldsymbol{H}\boldsymbol{\varsigma} = \mathbf{0} \ (\mathrm{mod}\ 2))$. A bipartite graph representation of $\boldsymbol{H}$, with bits marked as circles, checks marked as squares, and edges corresponding to respective nonzero elements of $\boldsymbol{H}$, is usually called (in the coding theory) the Tanner graph of the code, or parity-check graph of the code. (Notice that, fundamentally, code is defined in terms of the set of its codewords, and there are many parity check matrixes/graphs parameterizing the code. We ignore this unambiguity here, choosing one convenient parametrization $\boldsymbol{H}$ for the code.) Therefore the bi-partite Tanner graph of the code is defined as $\mathcal{G} = (\mathcal{G}_0, \mathcal{G}_1)$, where the set of nodes is the union of the sets associated with variables and checks, $\mathcal{G}_0 = \mathcal{G}_{0;v} \cup \mathcal{G}_{0;e}$ and only edges connecting variables and checks contribute $\mathcal{G}_1$.

For a simple example with 10 bits and 5 checks, the parity check (adjacency) matrix of the code with the Tanner graph shown in Fig. (8) is

$$\boldsymbol{H} = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}. \tag{III.2}$$

Another example of a bigger code and respective parity check matrix are shown in Fig. (9). For this example, $N = 155$, $L = 64$, $M = 91$ and the Hamming distance, defined as the minimum l_0-distance between two distinct codewords, is 20.

Assume that each bit of the transmitted signal is polluted independently of others and with some known conditional probability, $p(x|\sigma)$, where $\sigma = 0, 1$ is the valued of the bit before transmission, and x is its polluted image. Once $\boldsymbol{x} = (x_i | i = 1, \cdots, N)$ is measured, the task of the Maximum-A-Posteriori (MAP) decoding becomes to reconstruct the most probable codeword consistent with the measurement:

$$\boldsymbol{\sigma}^{(MAP)} = \arg \min_{\boldsymbol{\sigma}\in\Xi^{(\mathrm{CW})}} \prod_{i=1}^{N} p(x_i|\sigma_i). \tag{III.3}$$

More generally, the probability of a codeword $\boldsymbol{\varsigma} \in \Xi^{(\mathrm{CW})}$ to be a pre-image for $\boldsymbol{x}$ is

$$\mathcal{P}(\boldsymbol{\sigma}|\boldsymbol{x}) = (Z(\boldsymbol{x}))^{-1} \prod_{i\in\mathcal{G}_{0;v}} g^{(\mathrm{channel})}(x_i|\varsigma_i), \quad Z(\boldsymbol{x}) = \sum_{\boldsymbol{\varsigma}\in\Xi^{(\mathrm{CW})}} \prod_{i\in\mathcal{G}_{0;v}} g^{(\mathrm{channel})}(x_i|\varsigma_i), \tag{III.4}$$

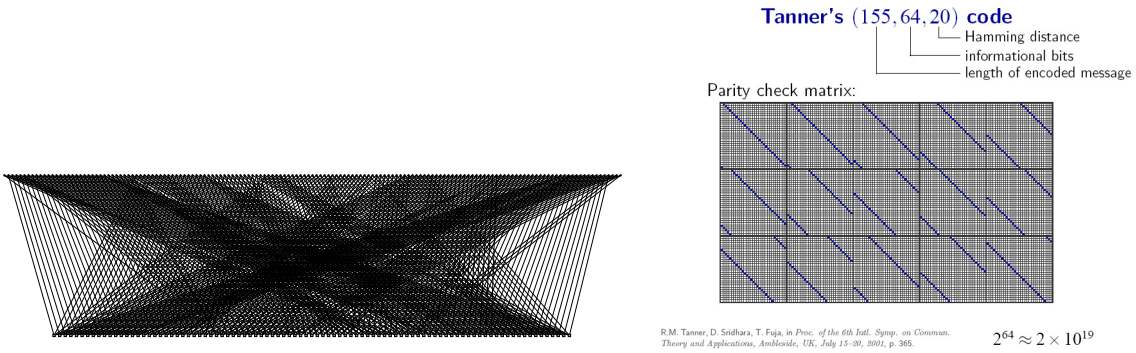

FIG. 9: Tanner graph and parity check matrix of the $(155, 64, 20)$ Tanner code, where $N = 155$ is the length of the code (size of the code word), $L = 64$ and the Hamming distance of the code, $d = 20$.

where $Z(\boldsymbol{x})$ is thus the partition function dependent on the detected vector $\boldsymbol{x}$. One may also consider the signal (bit-wise) MAP decoder

$$\forall\, i: \quad \varsigma_i^{(s-MAP)} = \arg\max_{\varsigma_i} \sum_{\boldsymbol{\varsigma}\setminus\varsigma_i}^{\boldsymbol{\varsigma}\in\Xi^{(\mathrm{CW})}} \mathcal{P}(\boldsymbol{\varsigma}|\boldsymbol{x}). \tag{III.5}$$

3. *Partition Function. Marginal Probabilities. Maximum Likelihood.*

The partition function in Eq. (III.1) is the normalization factor

$$Z = \sum_{\sigma} \prod_{a\in\mathcal{V}_f} f_a(\sigma_a), \quad \sigma_a \doteq (\sigma_i | i \in \mathcal{V}_n, \quad (i,a) \in \mathcal{E}), \tag{III.6}$$

where $\sigma = (\sigma_i \in \{0,1\} \in \mathcal{V}_n)$. Here, we assume that the alphabet of the elementary random variable is binary, however generalization to the case of a higher alphabet is straightforward.

We are interested to 'marginalize' Eq. (III.1) over a subset of variables, for example over all the elementary/nodal variable but one

$$P(\sigma_i) \doteq \sum_{\sigma\setminus\sigma_i} P(\sigma). \tag{III.7}$$

Expectation of σ_i computed with the probability Eq. (III.7) is also called (in physics) 'magnetization' of the variable.

Exercise: Does a partition function oracle sufficient for computing $P(\sigma_i)$?

Exercise: What is the relation in the case of the Ising model between $P(\sigma_i)$ and $Z(h)$?

Another object of interest is the so-called Maximum Likelihood. Stated formally, is the most probable state of all represented in Eq. (III.1):

$$\sigma_* = \arg\max_{\sigma} P(\sigma). \tag{III.8}$$

Exercise: Consider Ising model at temperature, T, where $J \to J/T$ and $h \to h/T$ in Eq. (II.37). How can one extract ML (III.8) from $Z(T)$?

All these objects are difficult to compute. 'Difficulty' - still stated casually - means that the number of operations needed is exponential in the system size (e.g. number of variables/spins in the Ising model). This is in general, i.e. for a GM of a general position. However, for some special cases, or even special classes of cases, the computations may be much easier than in the worst case. Thus, ML (III.8) for the case of the so-called ferromagnetic (attractive, sub-modular) Ising model can be computed with efforts polynomial in the system size. Note that the partition function computation (at any nonzero temperatures) is still exponential even in this case, thus illustrating the general statement - computing Z or $P(\sigma_i)$ is a more difficult problem than computing σ_*.

A curious fact. Ising model (ferromagnetic, anti-ferromagnetic or glassy) when the 'magnetic field' is zero, $h = 0$, and the graph is planar, represents a very unique class of problems for which even computations of Z are easy. In this case the partition

function is expressed via determinant of a finite matrix, while computing determinant of a size N matrix is a problem of $O(N^3)$ complexity (actually $O(N^{3/2})$ in the planar case).

In the general (difficult) case we will need to relay on approximations to make computations scalable. And some of these approximations will be discussed later in the lecture. However, let us first prepare for that - restating the most general problem discussed so far – computation of the partition function, Z – as an optimization problem.

4. Kublack-Leibler Formulation & Probability Polytope

We will go from counting (computing partition function is the problem of weighted counting) to optimization by changing description from states to probabilities of the states, which we will also call beliefs. $b(\sigma)$ will be a belief - which is our probabilistic guess - for the probability of state σ. Consider it on the example of the triangle system shown in Fig. (7). There are 2^3 states in this case: $(\sigma_1 = \pm 1, \sigma_2 = \pm 1, \sigma_3 = \pm 1)$, which can occur with the probabilities, $b(\sigma_1, \sigma_2, \sigma_3)$. All the beliefs are positive and together should some to unity. We would like to compare a particular assignment of the beliefs with $P(\sigma)$, generally described by Eq. (III.1). Let us recall a tool which we already used to compare probabilities - the Kublack-Leibler (KL) divergence (of probabilities) discussed in Lecture #2:

$$D(b\|P) = \sum_{\sigma} b(\sigma) \log\left(\frac{b(\sigma)}{P(\sigma)}\right) \tag{III.9}$$

Note that the KL divergence (III.9) is a convex function of the beliefs (remember, there are 2^3 of the beliefs in the our enabling three node example) within the following polytope – domain in the space of beliefs bounded by linear constraints:

$$\forall \sigma: \quad b(\sigma) \geq 0, \tag{III.10}$$

$$\sum_{\sigma} b(\sigma) = 1. \tag{III.11}$$

Moreover, it is straightforward to check (please do it at home!) that the unique minimum of $D(b\|P)$ is achieved at $b = P$, where the KL divergence is zero:

$$P = \arg\min_{b} D(b\|P), \quad \min_{b} D(b\|P) = 0. \tag{III.12}$$

Substituting Eq. (III.1) into Eq. (III.12) one derives

$$\log Z = -\min_{b} \mathcal{F}(b), \quad \mathcal{F}(b) \doteq \sum_{\sigma} b(\sigma) \log\left(\frac{\prod_a f_a(\sigma_a)}{b(\sigma)}\right), \tag{III.13}$$

where $F(b)$, considered as a function of all the beliefs, is called (configurational) free energy (where configuration is one of the beliefs). The terminology originates from statistical physics.

To summarize, we did manage to reduce counting problem to an optimization problem. Which is great, however so far it is just a reformulation – as the number of variational degrees of freedom (beliefs) is as much as the number of terms in the original sum (the partition function). Indeed, it is not the formula itself but (as we will see below) its further use for approximations which will be extremely useful.

5. Variational Approximations. Mean Field.

The main idea is to reduce the search space from exploration of the $2^N - 1$ dimensional beliefs to their lower dimensional, i.e. parameterized with fewer variables, proxy/approximation. What kind of factorization can one suggest for the multivariate (N-spin) probabilities/belliefs? The idea of postulating independence of all the N variables/spins comes to mind:

$$b(\sigma) \to b_{MF}(\sigma) = \prod_i b_i(\sigma_i) \tag{III.14}$$

$$\forall i \in \mathcal{V}_i, \ \forall \sigma_i: \quad b_i(\sigma_i) \geq 0 \tag{III.15}$$

$$\forall i \in \mathcal{V}_i: \quad \sum_{\sigma_i} b_i(\sigma_i) = 1. \tag{III.16}$$

Clearly $b_i(\sigma_i)$ is interpreted within this substitution as the single-node marginal belief (estimate for the single-node marginal probability).

Substituting b by b_{MF} in Eq. (III.13) one arrives at the MF estimation for the partition function

$$\log Z_{mf} = -\min_{b_{mf}} \mathcal{F}(b_{mf}), \quad \mathcal{F}(b_{mf}) \doteq \sum_a \sum_{\sigma_a} \left(\prod_{i\sim a} b_i(\sigma_i) \right) \log f_a(\sigma_a) - \sum_i \sum_{\sigma_i} b_i(\sigma_i) \log(b_i(\sigma_i)). \tag{III.17}$$

Exercise: Show that $Z_{mf} \geq Z$. Exercise: Show that $\mathcal{F}(b_{mf})$ is a convex function of its (vector) argument.

To solve the variational problem (III.17) constrained by Eqs. (III.14,III.15,III.16) is equivalent to searching for the (unique) stationary point of the following MF Lagrangian

$$\mathcal{L}(b_{mf}) \doteq \mathcal{F}(b_{mf}) + \sum_i \lambda_i \sum_{\sigma_i} b_i(\sigma_i) \tag{III.18}$$

Exercise: Write down equations defining the stationary point of $\mathcal{L}(b_{mf})$. Suggest an iterative algorithm converging to the stationary point.

The fact that Z_{mf} gives an upper bound on Z is a good news. However, in general the approximation is very crude, i.e. the gap between the bound and the actual value is large. The main reason for that is clear - by assuming that the variables are independent we have ignored significant correlations.

In the next lecture we will analyze what, very frequently, provides a much better approximation for ML inference - the so called Belief Propagation approach.

B. Lecture #9. Inference & Learning with Belief Propagation

This lecture will continue the thread of the previous lecture. We will mainly focus on the so-called Belief Propagation, related theory and techniques. In addition to discussing inference with Belief Propagation we will also have a brief discussions (pointers) to respective inverse problem – learning with Graphical Models.

1. Bethe Free Energy & Belief Propagation

Bethe-Peierls or Belief Propagation (we will use the same abbreviation BP for both) usually provides a good approximation, which is provably exact in some important cases (when graph in the underlying GM is a tree) and also provides an empirically good approximation for a very broad family of problems stated on loopy graphs. See the original paper [9], a comprehensive review[10], and lecture notes from the author of the review, `http://www.eecs.berkeley.edu/~wainwrig/Talks/A_GraphModel_Tutorial`, for an advanced/additional reading.

Instead of Eq. (III.14) one uses the following BP substitution

$$b(\sigma) \to b_{bp}(\sigma) = \frac{\prod_a b_a(\sigma_a)}{\prod_i (b_i(\sigma_i))^{q_i-1}} \tag{III.19}$$

$$\forall a \in \mathcal{V}_f, \ \ \forall \sigma_a : \quad b_a(\sigma_a) \geq 0 \tag{III.20}$$

$$\forall i \in \mathcal{V}_n, \ \ \forall a \sim i : \quad b_i(\sigma_i) = \sum_{\sigma_a \setminus \sigma_i} b_a(\sigma_a) \tag{III.21}$$

$$\forall i \in \mathcal{V}_n : \quad \sum_{\sigma_i} b_i(\sigma_i) = 1. \tag{III.22}$$

where q_i stands for degree of node i. The physical meaning of the factor $q_i - 1$ on the rhs of Eq. (III.35) is straightforward: by placing beliefs associated with the factor-nodes connected by an edge with a node, i, we over-count contributions of an individual variable q_i times and thus the denominator term in Eq. (III.35) comes as a correction for this over-counting.

Substitution of Eqs. (III.35) into Eq. (III.13) results in what is called Bethe (of BP) Free Energy

$$\mathcal{F}_{bp} \doteq E_{bp} - S_{bp}, \tag{III.23}$$

$$E_{bp} \doteq \sum_a \sum_{\sigma_a} b_a(\sigma_a) \log f_a(\sigma_a) \tag{III.24}$$

$$\mathcal{S}_{bp} = \sum_a \sum_{\sigma_a} b_a(\sigma_a) \log b_a(\sigma_a) - \sum_i \sum_{\sigma_i} (q_i - 1) b_i(\sigma_i) \log b_i(\sigma_i), \tag{III.25}$$

where E_{bp} is the so-called self-energy (physics jargon) and S_{bp} is the BP-entropy (this name should be clear in view of what we have discussed about entropy so far). Thus the BP version of the KL-divergence minimization becomes

$$\arg\min_{b_a,b_i} \mathcal{F}_{bp}\Big|_{\text{Eqs. (III.36,III.21,III.22)}}, \tag{III.26}$$

$$\min_{b_a,b_i} \mathcal{F}_{bp}\Big|_{\text{Eqs. (III.36,III.21,III.22)}} \tag{III.27}$$

Question: Is $\mathcal{F}_{bp}$ a convex function (of its arguments)? [Not always, however for some graphs and/or some factor functions it might be.]

The ML (zero temperature) version of Eq. (III.26) is the following optimization

$$\min_{b_a,b_i} E_{bp}\Big|_{\text{Eqs. (III.36,III.21,III.22)}} \tag{III.28}$$

Note that the resulting optimization problem is a Linear Programming (LP) — minimizing linear objective over the set of linear constraints.

2. *Belief Propagation & Message Passing*

Let us restate Eq. (III.26) as an unconditional optimization. We use the standard method of Lagrangian multipliers to achieve it. The resulting Lagrangian is

$$\mathcal{L}_{bp}(b,\eta,\lambda) \doteq \sum_a \sum_{\sigma_a} b_a(\sigma_a)\log f_a(\sigma_a) - \sum_a \sum_{\sigma_a} b_a(\sigma_a)\log b_a(\sigma_a) + \sum_i \sum_{\sigma_i} (q_i-1) b_i(\sigma_i)\log b_i(\sigma_i)$$
$$- \sum_i \sum_{a\sim i} \sum_{\sigma_i} \eta_{ia}(\sigma_i)\left(b_i(\sigma_i) - \sum_{\sigma_a\setminus\sigma_i} b_a(\sigma_a)\right) + \sum_i \lambda_i\left(\sum_{\sigma_i} b_i(\sigma_i) - 1\right), \tag{III.29}$$

where η and λ are the dual (Lagrangian) variables associated with the conditions Eqs. (III.21,III.22) respectively. Then Eq. (III.26) become the following min-max problem

$$\min_b \max_{\eta,\lambda} \mathcal{L}_{bp}(b,\eta,\lambda). \tag{III.30}$$

Changing the order of optimizations in Eq. (III.30) and then minimizing over η one arrives at the following expressions for the beliefs via messages ((check the derivation details)

$$\forall a,\ \forall \sigma_a: \quad b_a(\sigma_a) \sim f_a(\sigma_a)\exp\left(\sum_{i\sim a}\eta_{ia}(\sigma_i)\right) \doteq f_a(\sigma_a)\prod_{i\sim a} n_{i\to a}(\sigma_i) \doteq f_a(\sigma_a)\prod_{i\sim a}\prod_{b\sim i}^{b\neq a} m_{b\to i}(\sigma_i) \tag{III.31}$$

$$\forall i,\ \forall \sigma_i: \quad b_i(\sigma_i) \sim \exp\left(\frac{\sum_{a\sim i}\eta_{ia}(\sigma_i)}{q_i-1}\right) \doteq \prod_{a\sim i} m_{a\to i}(\sigma_i), \tag{III.32}$$

where, as usual, $\sim$ for beliefs means equality up the constant which guarantees that the sum of respective beliefs is unity, and we have also introduce the auxiliary variables , m and n, called messages, and related to the Lagrangian multipliers η as follows

$$\forall i,\ \forall a\sim i: \quad n_{i\to a}(\sigma_i) \doteq \exp\left(\eta_{ia}(\sigma_i)\right) \tag{III.33}$$

$$\forall a,\ \forall i\sim a: \quad m_{a\to i}(\sigma_i) \doteq \exp\left(\frac{\eta_{ia}(\sigma_i)}{q_i-1}\right). \tag{III.34}$$

Combining Eqs. (III.31,III.32) with Eq. (III.22) results in the following set of BP-equations stated in terms of the message variables

$$\forall i,\ \forall a\sim i,\ \forall \sigma_i: \quad n_{i\to a}(\sigma_i) = \prod_{b\sim i}^{b\neq a} m_{a\to i}(\sigma_i) \tag{III.35}$$

$$\forall a,\ \forall i\sim a,\ \forall \sigma_i: \quad m_{a\to i}(\sigma_i) = \sum_{\sigma_a\setminus\sigma_i} f_a(\sigma_a)\prod_{j\sim a}^{j\neq i} n_{j\to a}(\sigma_j). \tag{III.36}$$

Note that if the Bethe Free Energy (III.23) is non-convex there may be multiple fixed points of the Eqs. (III.35,III.36). The following iterative, so called Message Passing (MP), algorithm (5) is used to find a fixed point solution of the BP Eqs. (III.35,III.36)

Algorithm 5 Message Passing, Sum-Product Algorithm [factor graph representation]

Input: The graph. The factors.

1: $\forall i,\ \forall a \sim i, \forall \sigma_i :\quad m_{a\to i} = 1$ [initialize variable-to-factor messages]
2: $\forall a,\ \forall i \sim a, \forall \sigma_i :\quad n_{i\to 1} = 1$ [initialize factor-to-variable messages]
3: **loop**Till convergence within an error [or proceed with a fixed number of iterations]
4: $\quad \forall i,\ \forall a \sim i,\ \forall \sigma_i :\quad n_{i\to a}(\sigma_i) \leftarrow \prod_{b\sim i}^{b\neq a} m_{a\to i}(\sigma_i)$
5: $\quad \forall a,\ \forall i \sim a,\ \forall \sigma_i :\quad m_{a\to i}(\sigma_i) \leftarrow \sum_{\sigma_a\setminus\sigma_i} f_a(\sigma_a) \prod_{j\sim a}^{j\neq i} n_{j\to a}(\sigma_j)$
6: **end loop**

Exercise: Derive the $T = 0$ version of the aforementioned (see previous exercise) message-passing equations. A hint: the iterative equations should contain alternating min- and sum- steps — thus the name min-sum algorithm.

Exercise (bonus): Study performance of the message-passing algorithm on example of a small code decoding, for example see (this is a student midterm paper !) `http://www.people.fas.harvard.edu/~rpoddar/Papers/ldpc.pdf` for discussion of decoding of a binary $(3, 6)$ code over the Binary-Erasure Channel (BEC). Show how BP decodes and contrast the BP decoding against the MAP decoding. What is the (best) complexity of the MAP decoder for a code over the BEC channel. [Hint: Use Gaussian Elimination over GL(2).]

3. Sufficient Statistics

So far we have been discussing direct (inference) GM problem. In the remainder of this lecture we will briefly talk about inverse problems. This subject will also be discussed (on example of the tree) in the following recitation.

Stated casually - the inverse problem is about 'learning' GM from data/samples. Think about the two room setting. In one room a GM is known and many samples are generated. The samples, but not GM (!!!), are passed to the second room. The task becomes to reconstruct GM from samples.

The first question we should ask is if this is possible in principle, even if we have an infinite number of samples. A very powerful notion of *sufficient statics* helps to answer this question.

Consider the Ising model (not the first time in this course) using a little bit different notations then before

$$P(\sigma) = \frac{1}{Z(\theta)} \exp\left\{\sum_{i\in V} \theta_i\sigma_i + \sum_{\{i,j\}\in E} \theta_{ij}\sigma_i\sigma_j\right\} = \exp\{\theta^T\phi(\sigma) - \log Z(\theta)\}, \tag{III.37}$$

where $\sigma_i \in \{-1, 1\}$ and the *partition function* $Z(\theta)$ serves to normalize the probability distribution. In fact, Eq. (III.37) describes what is called an *exponential family* - emphasizing 'exponential' dependence on the factors θ.

Exercise: Show that any pairwise GM over binary variables can be represented as an Ising model.

Consider collection of all first and second moments (but only these two) of the spin variables, $\mu^{(1)} \doteq (\mu_i = \mathbb{E}[\sigma_i], i \in V)$ and $\mu^{(2)} \doteq (\mu_{ij} = \mathbb{E}[\sigma_i\sigma_j], \{i, j\} \in E)$. The *sufficient statistics* statement is that to reconstruct θ, fully defining the GM, it is *sufficient* to know $\mu^{(1)}$ and $\mu^{(2)}$.

4. Maximum-Likelihood Estimation/Learning of GM

Let us turn the *sufficiency* into a constructive statement – the *Maximum-likelihood estimation* over an exponential family.

First, notice that (according to the definition of μ)

$$\forall i :\quad \partial_{\theta_i} \log Z(\theta) = -\mu_i, \quad \forall i, j :\quad \partial_{\theta_{ij}} \log Z(\theta) = -\mu_{ij}. \tag{III.38}$$

This leads to the following statement: if we know how to compute log-partition function for any values of θ - reconstructing 'correct' θ is a convex optimization problem (over θ):

$$\theta^* = \arg\max_\theta \{\mu^T\theta - \log Z(\theta)\} \tag{III.39}$$

If P represents the empirical distribution of a set of independent identically-distributed (iid) samples $\{\sigma^{(s)}, s = 1, \dots, S\}$ then μ are the corresponding empirical moments, e.g. $\mu_{ij} = \frac{1}{S}\sum_s \sigma_i^{(s)}\sigma_j^{(s)}$.

General (concluding) Remarks about GM Learning. The ML parameter Estimation (III.39) is the best we can do. It is fundamental for the task of Machine Learning, and in fact it generalizes beyond the case of the Ising model.

Unfortunately, there are only very few nontrivial cases when the partition function can be calculated efficiently for any values of θ (or parametrization parameters if we work with more general class of GM than described by the Ising models).

Therefore, to make the task of parameter estimation practical one needs to rely on one of the following approaches:

- Limit consideration to the class of functions for which computation of the partition function can be done efficiently for any values of the parameters. We will discuss such case in the recitation – this will be the so-called tree (Chow-Lou) learning. (In fact, the partition function can also be computed efficiently in the case of the planar Ising - one of the suggested projects covers this advance subject.)
- Relay on approximations, e.g. such as MF and BP (but there are also other).
- There is exists a very innovative new approach - which allows to learn GM efficiently however using more information than suggested by the notion of the *sufficient statistics*. How one of scientists contributing to this line of research put it – 'the sufficient statistics is not sufficient'. This is a fascinating novel subjects, which is however beyond the scope of this course.

5. Recitation: Direct (Inference) and Inverse (Learning) Problems over Trees.

IV. THEME #4: STOCHASTIC MODELING & OPTIMIZATION

A. Lecture #10. Space-time Continuous Stochastic Processes

In this lecture we discuss stochastic dynamics of continuous variables governed by the Langevin equation. We discuss how to derive dynamic equation for the probability of a state - called Fokker-Planck. We go through the basic example of the Brownian motion and the diffusion equation.

1. Langevin equation in continuous and discrete time

Let us start with a continuous time forced and stochastic process in 1d, described in the time-continuous and time-discrete forms as follows

$$\dot{x} = -F(x) + \sqrt{D}\xi(t), \quad \langle x\rangle = 0, \quad \langle \xi(t_1)\xi(t_2)\rangle = \delta(t_1 - t_2) \tag{IV.1}$$

$$x_{n+1} - x_n = -F(x_n) + \sqrt{D\Delta}\zeta(t_n), \quad \langle \zeta(t_n)\rangle = 0, \quad \langle \zeta(t_n)\zeta(t_k)\rangle = \delta_{kn} \tag{IV.2}$$

The first and second terms on the rhs of Eq. (IV.1) stand for the force and "noise". The noise at each time step is independent. These equations, also called Langevin equations, describe spatial advance of x originating from two contributions (e.g. two terms on the rhs of Eq. (IV.1)), one dependent on the previous position deterministically and another being a random increment. In physics, this type of equation describes forced evolution in time t of a particle subject to random kicks by other particles - modeled through the stochastic term ξ. At any given moment of time t the particle is characterized by a its position x taken values from $\mathbb{R}$. Stochasticity means that trajectory of a particle under the deterministic force f and stochastic kicks, associated with ξ can take many values - thus we will be talking about probability distribution functions of possible (actually infinitely many) paths of the particle.

The square root on the rhs of Eq. (IV.2) may seem mysterious, let us clarify its origin on example of $F(x) = 0$. (This will be a running example through out this lecture.) In this, $F(x) = 0$, the basic Langevin equation describes Brownian motion, were the direct integration of the linear equation with the inhomogeneous source gives

$$\forall t \geq 0: \quad x(t) = \int_0^t dt' \xi(t'), \tag{IV.3}$$

$$\forall t \geq 0 \quad \langle x^2(t)\rangle = \int_0^t dt_1 \int_0^t dt_2 D\delta(t_1 - t_2) = D\int_0^t dt_1 = Dt, \tag{IV.4}$$

where we have also accounted that $x(0) = 0$. Infinitesimal version of Eq. (IV.5) is

$$\delta x = \sqrt{D\Delta}, \tag{IV.5}$$

which is just the Brownian version of Eq. (IV.2).

2. From Langevin to Path Integral

The Langevin equation can also be viewed as a way to relate change in $x(t)$, i.e. dynamic of interest, to stochastic dynamic of the δ-correlated source $\zeta(t_n) = \zeta_n$ characterized by the Probability Density Function (PDF)

$$P(\zeta_1, \cdots, \zeta_N) = (2\pi)^{-N/2} \exp\left(-\sum_{n=1}^{N} \frac{\zeta_n^2}{2}\right) \tag{IV.6}$$

Eqs. (IV.1,IV.2,IV.7) are starting points for our further derivations, but they should also be viewed as a way (explanation) of how to simulate Langevin - generating many paths (can do it simultaneously). Notice that there are other ways of simulating the Langevin equation, e.g. through the telegraph process, which we will not discuss here.

Let us express ζ_n via x_n from Eq. (IV.2) and substitute it into Eq. (IV.7)

$$P(\zeta_1, \cdots, \zeta_{N-1}) \to P(x_1, \cdots, x_N) = (2\pi D)^{-(N-1)/2} \exp\left(-\frac{1}{2D\Delta} \sum_{n=1}^{N-1} (x_{n+1} - x_n + \Delta F(x))^2\right) \tag{IV.7}$$

one gets an explicit expression for the measure over a path written in the discretized way. Also here is a typical way of how we state it in the continuous form (useful as a shortcut for notations)

$$P\{x(t)\} \propto \exp\left(-\frac{1}{2D} \int_0^T dt\, (\dot{x} + F(x))^2\right) \tag{IV.8}$$

This object is called (in physics and math) "path integral" and/or Feynmann/Kac integral.

3. From Path Integral to Fokker-Planch (through sequential Gaussian integration)

PDF over the path is a useful general object. However we may also want to marginalize it and extract

$$P_N(x_N) = \int dx_1 \cdots dx_N P(x_1, \cdots, x_N) P_1(x_1), \tag{IV.9}$$

from $P(x_1, \cdots, x_N)$ and also the prior/initial (distribution) $P_1(x_1)$ – both are assumed known. It is actually convenient to derive relation between $P_N(\cdot)$ and $P_1(\cdot)$ in steps, i.e. through a recurrence/induction. To simplify (and thus to illustrate further evaluations), let us proceed analyzing the case of the Brownian motion where, $F = 0$. Then the first step of the induction becomes

$$P_2(x_2) = (2\pi D)^{-1/2} \int dx_1 \exp\left(-\frac{1}{2D\Delta} (x_2 - x_1)^2\right) P_1(x_1) \tag{IV.10}$$

$$= (2\pi D)^{-1/2} \int d\epsilon \exp\left(-\frac{\epsilon^2}{2D\Delta}\right) P_1(x_2 - \epsilon) \tag{IV.11}$$

$$\approx (2\pi D)^{-1/2} \int d\epsilon \exp\left(-\frac{\epsilon^2}{2D\Delta}\right) \left(P_1(x_2) - \epsilon \partial_x P_1(x_2) + \frac{\epsilon^2}{2} \partial_x^2 P_1(x_2)\right) \tag{IV.12}$$

$$= P_1(x_2) + \Delta \frac{D}{2} \partial_x^2 P_1(x_2), \tag{IV.13}$$

where transitioning from Eq. () to Eq. () one makes Taylor expansion in ϵ, assuming that $\epsilon \sim \sqrt{\Delta}$ and keeping only the leading terms in Δ. The resulting Gaussian integrations are straightforward resulting in the discretized (in time) version of the diffusion equation

$$\partial_t P_t(x) = \frac{D}{2} \partial_x^2 P_t(x). \tag{IV.14}$$

(Not surprisingly the Brownian motion for stochastic dynamic has resulted in the diffusion equation for the probability distribution. Restoring the $U(x)$ term (derivation is straightforward) one arrives at the Fokker-Planck equation, generalizing the zero-force diffusion equation

$$\partial_t P_t(x) - \partial_x (U(x) P_t(x)) = \frac{D}{2} \partial_x^2 P_t(x). \tag{IV.15}$$

4. Analysis of Fokker-Planck: General Features and Examples

Here we only give a very brief and incomplete description on the properties of the distribution which analysis is fundamental to Statistical Physics. See e.g. [1]. Some selected subjects will also be discussed in the recitations.

The Fokker-Planck equation (IV.15) is linear and deterministic Partial Differential Equation (PDE). It describes continuous in phase space, x, and time, t, evolution/flow of the probability distribution.

Derivation was for a particle moving in 1d, $\mathbb{R}$, but the same ideology and logic extends to higher dimensions, $\mathbb{R}^d$, $d = 1, 2, \cdots$. There are also extension of this consideration to compact continuous spaces. Thus one can analyze dynamics on a circle, sphere or torus.

Analogs of the Fokker-Planck can be derived and analyzed for more complicated probabilities than just the marginal probability of the state (path integral marginalized to given time). An example – of the "first passage" probability will be given in the recitation.

The temporal evolution is driven by two terms - "diffusion" and "advecton" - the terminology is from fluid mechanics - indeed not only fluids but also probabilities can flow. The flow of probability is in the phase space. The diffusion originates from the stochastic source while advection from the deterministic (but possibly nonlinear) deterministic force.

Linearity of the Fokker-Planck does not imply that it is simpler than the original nonlinear problem. Deriving the Fokker-Planck we made a transition from nonlinear, stochastic but ODE to linear PDE. This type of transition from nonlinear representation of many trajectories to linear probabilistic representation is typical in math/statistics/physics. One one view linear Fokker-Planck as the continuous-time, continuous-space version of the discrete-time/discrete space Master equation describing evolution of a (finite dimensional) probability vector in the case of a Markov Chain.

The Fokker-Planck Eq. (IV.15) can be represented in the 'flux' form:

$$\partial_t P_t + \partial_x J_t(x) = 0 \tag{IV.16}$$

where $J_t(x)$ is the flux of probability through the space-state point x at the moment of time t. The fact that the second (flux) term in Eq. (IV.16) has a gradient form, corresponds to the global conservation of probability. Indeed, integrating Eq. (IV.16) over the whole continuous domain of achievable x, and assuming that if the domain is bounded there is no injection (or dissipation) of probability on the boundary, one finds that the integral of the second term is zero (according to the standard Gauss theorem of calculus) and thus, $\partial_t \int dx P_t(x) = 0$. In the steady state, when $\partial_t P_t = 0$ for all x (and not only in the result of integration over the entire domain) the flux is constant - does not depend on x. The case of zero-flux is the special case of the so-called 'equilibrium' statistical mechanics. (See some further comments below on the latter.)

If the initial probability distribution, $P_{t=0}(x)$ is known, $P_t(x)$ for any consecutive t is well defined, in the sense that the Fokker-Planck is the Cauchi (initial value) problem with unique solution.

Remarks about simulations. One can solve PDE but can also analyze stochastic ODE approaching the problem in tw complementary ways - correspondent to Eulerian and Lagrangian analysis in Fluid Mechanics.

Main and the simplest (already mentioned) example of the Langevin dynamic is the Brownian motion, i.e. the case of $F = 0$. Another example, principal for the so-called 'equilibrium statistical physics', is of the potential force $F = \partial_x U(x)$, where $U(x)$ is a potential. Think, for example about x representing a particle connected to the origin by a spring. $U(x)$ is the potential/energy stored in the spring. In this case (of the gradient force) the stationary (i.e. time-independent) solution of the Fokker Planck Eq. (IV.15) can be found explicitly,

$$P_{st}(x) = Z^{-1} \exp\left(-\frac{U(x)}{D}\right). \tag{IV.17}$$

This solution is called the Gibbs distribution, or equilibrium distribution.

Exercise: Show that the dynamic in the gradient force case obeys the Detailed Balance.

5. Recitation. Homogeneous and Forced Brownian Motion.

6. Recitation. First Passage Problem. Effects of Boundaries. Kramers Escape Problem.

B. Lecture #11. Queuing Systems.

1. Queuing: a bit of History & Applications

There are number of books written on the subject. The old [11] and new [**?**] books of Frank Kelly are recommended.

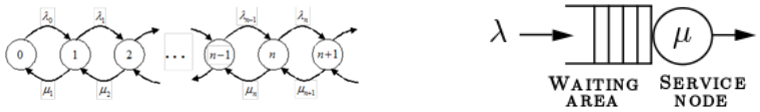

FIG. 10: On the left: Markov Chain representation of the M/M/1 queue. In the standard situation considered $\forall i: \quad \lambda_i = \lambda, \mu_i = \mu$. On the right: reduced graphical description of a single queue.

Agner Krarup Erlang, a Danish engineer who worked for the Copenhagen Telephone Exchange, published the first paper on what would now be called queueing theory in 1909. He modeled the number of telephone calls arriving at an exchange by a Poisson process and solved the $M/D/1/\infty$ queue in 1917 and $M/D/k/\infty$ queueing model in 1920.

The notations are now standard in the Queueing theory – which is a discipline traditionally considered as a part of Operation Research with deep connection to statistics and physics. In $M/D/k/\infty$, for example,

- **M** stands for Markov or memoryless and it means that arrivals occur according to a Poisson process. Arrivals may also be deterministic, D.
- **D** stands for deterministic and means that the jobs arriving at the queue require a fixed=deterministic amount of service/processing. Processing can also be stochastic, Markovian (or non-Markovian, in which case it is custom to mark it as G - generic sevice; arrival can also be G=generic).
- k describes the number of servers at the queueing node $k = 1, 2, ...$ If there are more jobs at the node than there are servers then jobs will queue and wait for service.
- ∞ stands for the allowed size of the queue (waiting room) - in this case no limit to the waiting room capacity (everybody arriving is admitted to the queue - not denied)

We will only be dealing with the case of ∞ waiting room, thus dropping the last argument.

The $M/M/1$ queue is a simple model where a single server handles jobs that arrive according to a Poisson process and have exponentially distributed service requirements.

In an M/G/1 queue the G stands for general and indicates an arbitrary probability distribution.

Many mathematicians and math-engineers contributed the subject since 1930 — Pollaczek, Khinchin, Kendall, Kingman, Jackson, Kelly and others.

Applications: call centers, logistics (at different scales), manufacturing, checkout at the super-market, processing of electric vehicles at the charging stations, etc. In general, any kind of practical systems where arrivals (of whatever coming in units) and processing fits the framework. We are talking about design which would

- Manage queue (controls its size).
- Keep processing units busy (good utilization).
- Have waiting time in the queue under control.

2. *Single Open Queue = Birth/Death process. Markov Chain representation.*

Let us discuss in details $M/M/1$. We start by playing with the Java modeling tool – JMT (can upload it from `http://jmt.sourceforge.net/Download.html`)

The process is also called birth-death process - the name is clear from the Markov-Chain representation shown in Fig. (10). The MC has infinitely many states, each representing # of customers in the system (waiting room). Arrival of customers is modeled as the Poisson process with the arrival rate, λ. We assume that all customers are identical. The customers are taken from the waiting room based on availability of the servant, and the service is completed with the rate μ of the other Poisson process.

Everything is Poisson in here (recall that mixing and splitting of the Poisson processes is Poisson again).

Let us analyze the (relatively simple) system. Let us start from finding the steady state of the Markov Chain: $\forall i = 0, \cdots, \infty \quad P_i$, where P_i is the probability that the system is in the i-th state, i.e. with i customers in the queue.

The balance equations are

$$\text{\# 0 customers:} \quad \underbrace{\mu P_1}_{arrival} = \underbrace{\lambda P_0}_{departure} \tag{IV.18}$$

$$\text{\# 1 customer:} \quad \lambda P_0 + \mu P_2 = (\lambda + \mu) P_1 \tag{IV.19}$$

$$\text{\# n customers:} \quad \lambda P_{n-1} + \mu P_{n+1} = (\lambda + \mu) P_n \tag{IV.20}$$

Resolving the equations (sequentially), and requiring that the total probability is normalized, $\sum_{i=0}^{\infty} P_i = 1$, we derive

$$P_n = \left(\prod_{i=0}^{n-1} \frac{\lambda}{\mu} \right) P_0 = \left(\frac{\lambda}{\mu} \right)^n P_0 = \rho^n P_0 \tag{IV.21}$$

$$1 = \sum_{n=0}^{\infty} P_n = P_0 \sum_{n=0}^{\infty} \rho^n = \frac{P_0}{1-\rho} \tag{IV.22}$$

$$P_n = (1-\rho)\,\rho^n. \tag{IV.23}$$

where $\rho \doteq \lambda/\mu$ is the traffic intensity.

The average size of the queue is:

$$\sum_{n=0}^{\infty} n P_n = (1-\rho) \sum_{n=0}^{\infty} n \rho^n = \frac{\rho}{1-\rho}. \tag{IV.24}$$

We observe that the average queue becomes infinite at $\rho = 1$, i.e. the steady state exists only when $\rho < 1$. This criterium (existence of the steady state) can also be referred to as "stability".

Exercise: Consider a single M/M/m queue, i.e. the system when the number of servers is m. Derive steady state? What is the modified stability criterium? Can a single queue system withe $m = 2$ be unstable?

In this simple queue system we can also study transient time dynamics. The steady state system of Eqs. (IV.20) transitions to

$$\forall n: \quad \frac{d}{dt} P_n = \underbrace{\lambda P_{n-1} + \mu P_{n+1}}_{\text{arrival}} - \underbrace{(\lambda + \mu) P_n}_{\text{departure}} \tag{IV.25}$$

Solution of this system can be found in an analytic form [12]

$$P_k(t) = e^{-(\lambda+\mu)t} \left(\rho^{(k-i)/2} I_{k-i}(at) + \rho^{(k-i-1)/2} I_{k+i+1}(at) + (1-\rho)\rho^k \sum_{j=k+i+2} \rho^{-j/2} I_j(at) \right) \tag{IV.26}$$

where $a \doteq 2\sqrt{\lambda\mu}$, $I_k(x)$ is the modified Bessel function of the first kind, and we have assumed that the system was in the state i at $t = 0$.

Exercises [bonus/difficult]:

- Derive Eq. (IV.26) from Eq. (IV.25).
- Compute distribution time of the busy period of server.
- Assuming first come-first served policy, compute distribution of the waiting time and distribution of the total time in the system.

We can also write the dynamical Eq. (IV.25) in the following matrix form

$$\frac{d}{dt} P = P^{(\text{tr})} P, \quad P^{(\text{tr})} = \begin{pmatrix} \ddots & & & & & & \\ \cdots & 0 & \mu & -(\lambda+\mu) & \lambda & 0 & \cdots \\ & & & & & & \ddots \end{pmatrix} \tag{IV.27}$$

Notice that in the steady state (achievable at $\rho < 1$) the Detailed Balance (DB) does not hold, $P^{(\text{tr})}_{nm} P^{(\text{st})}_m \neq P^{(\text{tr})}_{mn} P^{(\text{st})}_n$.

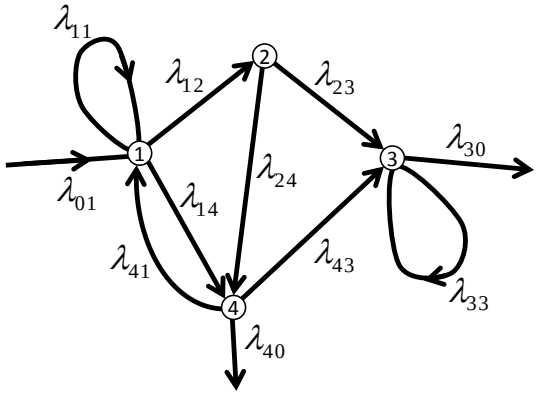

FIG. 11: Example of a Queuing network.

3. Generalization to (Jackson) Networks. Product Solution for the Steady State.

It appears that the description of a single Q- can be extended to a network, e.g. of the type shown in Fig. (11). λ are arrivals and processing rates – now denoted in the same way and indexed by nodes/stations from where the process is coming from and where it is going to (two indexes). We study, $P(n_1, \cdots, n_N; t)$, probability over the entire network and, like in the single queue case, write the balance (also called Master) equation – stated for any state of the network at any time

$$\frac{\partial}{\partial t}P(\boldsymbol{n};t) = \sum_{(i,j)\in\mathcal{E}} \lambda_{ij}\left(\underbrace{(n_i+1)P(\cdots,n_i+1,\cdots,n_j-1,\cdots;t)}_{\text{customers leaving } i \text{ for } j} - \underbrace{n_i P(\cdots,n_i,\cdots,n_j,\cdots;t)}_{\text{customers staying at } i}\right)$$
$$+ \sum_{i\in]calV} \lambda_{0i}\left(P(\cdots,n_i-1,\cdots;t) - P(\cdots,n_i,\cdots;t)\right)$$
$$+ \sum_{i\in\mathcal{V}} \lambda_{i0}\left((n_i+1)P(\cdots,n_i+1,\cdots;t) - n_i P(\cdots,n_i,\cdots;t)\right) \tag{IV.28}$$

This equation is written here for the case of $M/M/\infty$ - when the number of servers is infinite - this is the case when jobs are not waiting but are taken for processing (by tellers which are always available) immediately.

Exercise: Write down a $M/M/m$ version of Eq. (IV.28).

Remarkably the (complicated looking) Eq. (IV.28) allows an explicit steady state solution (for any graph!)

$$P(\boldsymbol{n}) = Z^{-1}\prod_{i\in\mathcal{E}}\frac{h_i^{n_i}}{\prod_{l_i=1}^{n_i} l_i},$$
$$\forall i\in\mathcal{V}: - \quad h_i \sum_{j\neq 0}^{(i,j)\in\mathcal{E}} \lambda_{ij} + \sum_{j\neq 0}^{(j,i)\in\mathcal{E}} \lambda_{ji}h_j + \lambda_{0i} - \lambda_{i0}h_i = 0$$

which is also called product solution/factorization (name is according to the structure).

Few important things to mention in here

- This is a product form solution which IS NOT a Gibbs (equilibrium) distribution. [Remember our discussions of the Fokker-Planck.]
- The system is stable (solution is finite) if: $\forall i\in\mathcal{V}, \quad h_i < m_i$
- $\boldsymbol{h}$ is a "single-customer" object

Exercise: Generalize the steady state formula and reformulate the stability for the general $M/M/m$ case.

4. Heavy Traffic Limit

Our discussion here is (mainly) based on the material from `http://www.columbia.edu/~ww2040/A1a.html`.

The Heavy traffic limit applies when either of the two cases (or some special combination of the two, which we will not discuss here) takes place (we discuss a single queue case - to make the arguments simpler):

- The number of servers is fixed and the traffic intensity (utilization), λ/μ, approaches unity (from below). The queue length approximation is the so-called "reflected Brownian motion" [13–15].
- Traffic intensity is fixed and the number of servers and arrival rate are increased to infinity. Here the queue length limit converges to the normal distribution [16–18].

Let us give some intuitive picture and then pose a number of technical questions/challenges (some of these with answers not yet fully known).

When the system is congested, i.e. most of the time it has many customers, customer which arrives to find an average queue, say with L customers in it, will remain there for $L+1$ services (assuming FIFO service), so he will have a long delay. While he is there, he will see the queue of customers ahead of him move down from L to 0. However, in this time of $L+1$ departures, there will also be many arrivals. If traffic intensity, ρ, is close to 1, the number of arrivals will be of order L as well. Thus, when the customer leaves he will have a queue behind him which is comparable to what he found. For the system to go from average to empty will take more like L busy periods.

Exercise: Assume that $1-\rho \ll 1$ and estimate

- How typical type a customer spend in the system scales with, $1-\rho$? $[(1\rho)^{-1}]$
- How typical typical time for the system to change say from average to empty scales with, $1-\rho$? $[(1\rho)^{-2}]$

Hint (following from the preceding the discussions): The time scale at which the system changes is much longer then the time scale of a single customer.

We therefore have a time scale separation and, therefore, may study a Q-system with many customers on two scales, fluid and diffusive. Let $X(t)$ be some Q-system related process. $\bar{X}_n(t) = nX(nt)$ defines the fluid re-scaling by n. This means that we measure time in the units of n and we measure the state (# of customers) in the units of n. As $n \to \infty$ we shall look for $n^{-1}X(nt) \to \bar{X}(t)$, where $\bar{X}(t)$ is the fluid limit.

At this scale as $n \to \infty$ the arrival process and the service process have fluid limits λt and μt which means that they are deterministic. As we said, queueing is the result of variability, and so on a fluid scale, **when input and output are not variable, there will be no real queueing behavior in the system**. We may see the queue length grow linearly indefinitely ($\rho > 1$), or go to zero linearly and then stay at 0 ($\rho < 1$), or we may see it constant, ($\rho = 1$). For queueing networks we may observe piecewise linear behavior of queue lengths. This will capture changes in the queue on the fluid scale: The queue changes by n in a time of order n. The stochastic fluctuations of a queue in steady state are scaled down to be identically 0 and uninteresting.

The diffusion scaling looks at the difference between the process and its fluid limit, and measures the time in units of n and the state (counts of customers) in units of $\sqrt{n}$. The diffusion re-scaling of $A(t)$ by n is $\hat{A}_n(t) = \sqrt{n}(\bar{A}_n(t)\bar{A}(t))$. As $n \to \infty$ we shall look (in analogy with the Central Limit Theorem) for $A_n(t)$ converging in the sense of distribution to $\hat{A}(t)$ describing the diffusion limit- it is a diffusion process, such as Brownian motion or reflected Brownian motion. The diffusion limit captures the random fluctuation of the system around its fluid limit.

Here is a (formal) statement on the heavy traffic asymptotic for the waiting time (including both fluid and diffusive limits): Consider $G/G/1$ indexed by j. For queue j let T_j denotes the random inter-arrival time, S_j denote the random service time; $\rho_j = \frac{\lambda_j}{\mu_j}$ denote the traffic intensity with $\frac{1}{\lambda_j} = \mathbb{E}(T_j)$ and $\frac{1}{\mu_j} = \mathbb{E}(S_j)$; $W_{q,j}$ denotes the waiting time in queue for a customer in steady state; $\alpha_j = -\mathbb{E}[S_j - T_j]$ and $\beta_j^2 = \text{var}[S_j - T_j]$. If $T_j \xrightarrow{d} T$, $S_j \xrightarrow{d} S$, and $\rho_j \to 1$, then $\frac{2\alpha_j}{\beta_j^2} W_{q,j} \xrightarrow{d} \exp(1)$ provided that: (a) Var[S-T] > 0, and (b) for some $\delta > 0, \mathbb{E}[S_j^{2+\delta}]$ and $\mathbb{E}[T_j^{2+\delta}]$ are both less than some constant C, $\quad \forall j$.

5. Recitation. Tandem Queue Example.

C. Lecture #12. Markov Decision Processes & Stochastic Optimal Control.

This lecture is based on the lecture notes of Bert Kappen on 'Optimal control theory and the linear Bellman Equation' `http://www.snn.ru.nl/v2/serve.php?doc=timeseriesbook.pdf`. Later in the lecture, while discussing the Markov Decision Processes (discrete time/discrete space/stochastic version of the general optimal control setting), we will also be using materials from the Berkeley Artificial Intelligence (AI) course `http://ai.berkeley.edu/lecture_videos.html`.

1. Optimal Control [discrete time, deterministic] & Dynamic Programming

We start from discrete-time deterministic setting described by the following equations

$$s_{t+1} = g(t, s_t, a_t), \quad t = 0, 1, \cdots, H-1 \tag{IV.29}$$

where s_t is an n-dimensional vector describing the state of the system (can be discrete or continuous) and a_t is an m-dimensional vector that specifies the control action or action at time t. If we specify $s = s_0$ at $t = 0$ and specify the sequence of controls, $a_{0:H-1} = a_0, a_1, \cdots, a_{H-1}$, we can compute future states of the system, $s_{1:H}$ recursively from Eq. (IV.29).

Define a function that assigns cost to each sequence of controls

$$C(s_0, a_{0:H-1}) = \phi(s_H) + \sum_{t=0}^{H-1} R(t, s_t, a_t), \tag{IV.30}$$

$R(t, s, a)$ is the reward/cost associated with taking action a at time t in state s, and $\phi(s_H)$ is the reward/cost associated with ending up in the state s_H at time horizon time H. The problem is to find the sequence $a_{0:H-1}$ that minimizes the cost, $C(s_0, a_{0:H-1}$. Notice that emphasizing dependence of $C(s_0, a_{0:H-1})$ on s_0 and $a_{0:H-1})$ one assumes that s_t at $t > 0$ on the rhs of Eq. (IV.30) are expressed via s_0 and $a_{0:H-1})$ according to Eq. (IV.29).

We are interested to find the *optimal cost to go*, also called the *optimal value*, and the respective *optimal control*

$$\min_{a_{0:H-1}} C(s_0, a_{0:H-1}), \quad \forall s_0 \tag{IV.31}$$

$$\arg\min_{a_{0:H-1}} C(s_0, a_{0:H-1}). \quad \forall s_0 \tag{IV.32}$$

Algorithm 6 Dynamic Programming [Backward in time Value Iteration]

Input: $R(t, s, a)$, $f(t, s, a)$ return the value of reward and the vector of incremental state corrections $\forall t, s, a$.

1: $V(H, s) = \phi(s)$
2: **for** $t = H, H-1, \cdots, 0$ **do**
3: $\quad {}^*_t(s) = \arg\min_a \left(R(t, s, a) + V(t+1, f(t, s, a))\right), \quad \forall s$
4: $\quad V(t, s) = R(t, s, a^*_t) + V(t+1, f(t, s, a^*_t)), \quad \forall s$
5: **end for**

Output: $a^*_t(s), \quad \forall s, t;$

Let us also introduce, exploring causality and Markovian (independence of the past) features of Eq. (IV.29), the *current optimal value*

$$V(t, s_t) = \min_{a_{t:H-1}} C(s_t, a_{t:H-1}), \quad \forall s_t, \quad \forall t \tag{IV.33}$$

The usefulness of this "current" object becomes apparent when one discovers, following the original contribution of Bellman made in 1952 and coined Dynamic Programming (DP), that the optimal value can be computed recursively backwards in time starting from $V(t+1, s)$ for all s

$$t = H: \qquad V(H, s) = \phi(s) \tag{IV.34}$$

$$\begin{aligned} t = H-1, \cdots, 0: \qquad V(t, s_t) &= \min_{a_t:H-1} \left(\phi(s_H) + \sum_{\tau=t}^{H-1} R(\tau, s_\tau, a_\tau) \right) \\ &= \min_{a_t} \left(R(t, s_t, a_t) + \min_{a_{t+1:H-1}} \left(\phi(s_H) + \sum_{\tau=t+1}^{H-1} R(\tau, s_\tau, a_\tau) \right) \right) \\ &= \min_{a_t} \left(R(t, s_t, a_t) + V(t+1, s_{t+1}) \right) \\ &= \min_{a_t} \left(R(t, s_t, a_t) + V(t+1, g(t, s_t, a_t)) \right) \end{aligned} \tag{IV.35}$$

The Bellman equation (IV.35) can be implemented through the following DP algorithm (6), which is linear in H (the horizon), linear in the size of the state space and linear in the size of the action space. (This is if we discretize the state space and action space proper.)

The DP algorithm is an example of what is also called in Computer Science a greedy algorithm, that is an algorithm that makes the locally optimal choice at each stage. In general greedy algorithms offer only a heuristic, that is an approximate (suboptimal) solution. However, the remarkable feature of the optimal control problem, which we just proved through the sequence of transformations shown in Eq. (IV.35), is that the greedy algorithm is also globally optimal/exact.

Note that we have already used a similar greedy algorithm of the DP type before when we discussed inference and learning over tree-graphs. For example in the learning setting we were solving the problem of finding the minimum spanning tree exactly in a similar greedy fashion.

Note also that our description so far allows straightforward generalization to the discrete space (Markov Decision Processes), which will be discussed latter in the lecture, and also to the continuous time - discussed next.

Exercise: Consider the classical (in optimization) problem of finding the shortest path over a graph. Solve it through the DP approach by analogy with what was discussed above.

2. *Optimal Control [continuous space, continuous time, deterministic]*

Generalizing consideration of the preceding Subsection to the continuous space, continuous time setting one derives at the following extension of Eq. (IV.35), after replacement of $t+1$ by $t+dt$ with $dt \to 0$ and $g(t,s,a) \to s + f(t,s,a)$ and keeping the leading (linear) in dt terms,

$$-\partial_t V(t,s) = \min_a \left(R(t,s,a) + f(t,s,a)\partial_s V(t,s) \right). \tag{IV.36}$$

The partial differential equation (IV.36) is called Bellman-Hamilton-Jacobi equation (the last two names were added to emphasize equivalence of the underlying formulas to the action-angle representation of the classical mechanics).

To ease the notations, Eq. (IV.36), and what follows below in the remainder of this lecture, is stated for the one-dimensional case, $s \in \mathbb{R}^1$. Generalization to a higher dimensional case is possible and (almost) straightforward. (See `http://www.snn.ru.nl/v2/serve.php?doc=timeseriesbook.pdf` for details.)

3. *Stochastic Optimal Control [continuous space, continuous time, stochastic]*

Next step consists in adding stochastic term to the Langevien equation with control

$$\frac{ds}{dt} = f(t,s(t),a(t)) + \xi(t,s(t),a(t)) \tag{IV.37}$$

$$\mathbb{E}[\xi] = 0, \quad \mathbb{E}[\xi(t,s(t),a(t))\xi(t',s(t'),a(t'))] = \nu(t,s(t),a(t))\delta(t-t'). \tag{IV.38}$$

Stochasticity requires modification of the cost function, for example replacing the cost by its averaged/expected value. (There may be other options, e.g. enhancing or suppressing, importance of deviations from the mean, not discussed here.) Direct implementation of this strategy will result in the following generalization of the deterministic BHJ equation (IV.36)

$$-\partial_t V(t,s) = \min_a \left(R(t,s,a) + f(t,s,a)\partial_s V(t,s) + \frac{1}{2}\nu(t,s,a)\partial_s^2 V(t,s) \right) \tag{IV.39}$$

Taken into account our experience with transition from Langevien to Fokker-Planck the addition of the 'diffusion' term is obvious/natural.

In a popular special case when the stochastic dynamic is linear in a, i.e. Eq. (IV.37) is replaced by

$$ds = f(t,s)dt + \sum_{j=1}^{p} G(t,s)\left(adt + d\xi\right), \tag{IV.40}$$

and the reward function is quadratic in a

$$R(t,s,a) = W(t,s) + \frac{1}{2}aRa, \tag{IV.41}$$

one can evaluate $\min_a$ in Eq. (IV.39) analytically. Here in Eqs. (IV.40,**??**) we use vector/matrix notations. Then the optimal action/control gets the following explicit form

$$a^* = -R^{-1}G\partial_s V, \tag{IV.42}$$

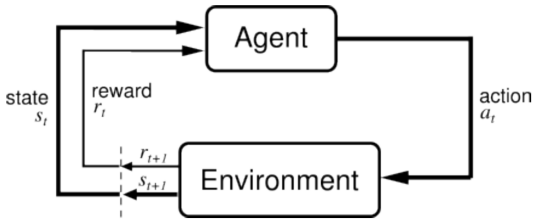

FIG. 12: General Scheme of Markov Decision Process. Drawing from Sutton & Barton 'Reenforcement Learning: An Introduction'.

utilizing the vector/matrix notations, e.g. assuming that the vector operator ∇ is constructed from the ∂_{s_i} components. Substitution of Eq. (IV.42) in the vector version of Eq. (IV.39) results in the following *nonlinear* equation for the value function

$$-\partial_t V = W + (\partial_s V) f + \frac{1}{2}\left(-GR^{-1}G(\partial_s V)(\partial_s V)^T + G\nu G\partial_s^2 V\right) \tag{IV.43}$$

The PDE is rich and interesting, however its analysis goes beyond what we can cover in the course. (See `http://www.snn.ru.nl/v2/serve.php?doc=timeseriesbook.pdf` and references therein for details.)

4. *Markov Decision Processes [discrete space, discrete time, stochastic]*

Let us consider the Markov Decision Processes which representstochastic, discrete space, discrete time version of the stochastic optimal control problem. Here we modify the setting

(a) assuming that the reward function does not depend explicitly on time,

(b) incorporating the future state in the reward function,

(c) averaging over the transition probability from the current state to the next state,

(d) introducing γ^t discount factor (less reward as time progresses).

Overall it results in the following formulation

$$a^* = \arg\max_{a.(\cdot)} \mathbb{E}\left[\sum_{t=0}^{H} \gamma^t R(s_t, a_t(s_t), s_{t+1})\right], \quad \mathbb{E}[X] \doteq \sum_{s_1,\cdots,s_i} X\left(\prod_{t'=0}^{i-1} P(s_{t'+1}|s_{t'}, a_{t'})\right). \tag{IV.44}$$

Solving MDP means finding optimal a, i.e. set of actions for each state at each moment of time, as illustrated on the GridWorld example (to be discussed next) in Fig. 13.

Our description here is intentionally terse/introductory. For a more colloquial, detailed and mathematical exposition of MDP check the lecture notes of Pieter Abbeel (UC Berkeley) `http://www.cs.berkeley.edu/~pabbeel/cs287-fa12/slides/mdps-exact-methods.pdf` from the Berkley AI course. In fact, the Berkeley course on AI also contains a very good repository of materials at `http://ai.berkeley.edu/lecture_videos.html`. Our running 'Grid World' example/illustration of MDP (comes next) is used intensively in the lecture series, see `http://aima.cs.berkeley.edu/demos.html` and also `http://www2.hawaii.edu/~chenx/ics699rl/grid/`.

5. *MDP: Grid World Example*

MDP can be considered as an interactive probabilistic game one plays against computer (random number generator). The game consists in defining transition rates between the states to achieve certain objectives. Once optimal (or suboptimal) rates are fixed the implementation becomes just a Markov Process we have studied already.

Let us play this 'Grid World' game a bit. The rules are introduced in Fig. (14). An agent lives on the grid (3×4). Walls block the agent's path. The agent actions do not always go as planned: 80% of time the action 'North' take the agent 'North' (if there is no wall there), 10% of the time the action 'North' actually takes the agent West; 10% East. If there is a wall the agent would have been taken, she stays put. Big reward, $+1$, or penalty, -1 comes at the end. We will come to this example many times during this lecture.

We will consider the following *Value Iteration* algorithm [19]:

The Grid World implementation of the algorithm is illustrated in Fig. (15).

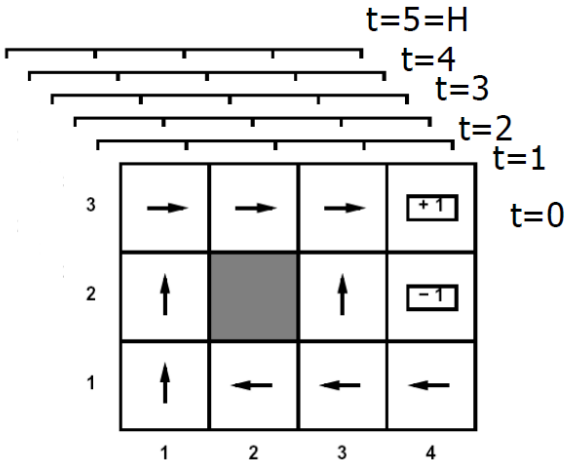

FIG. 13: Optimal solution set of actions (arrows) for each state, for each time.

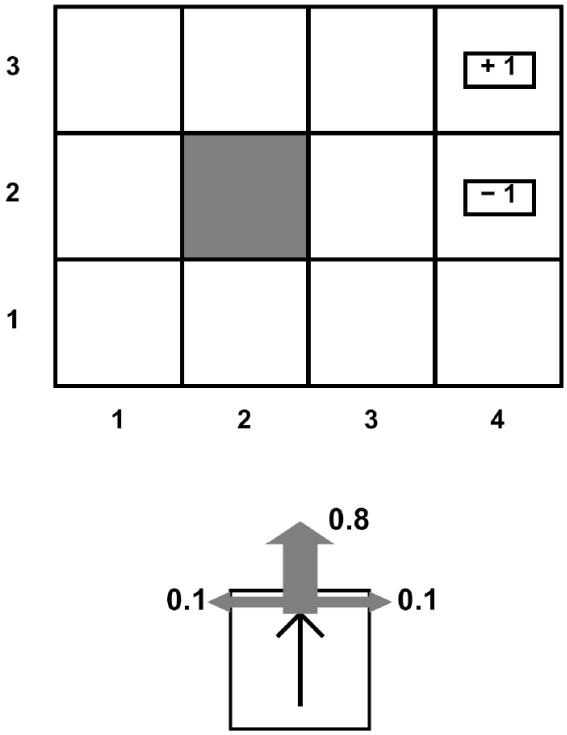

FIG. 14: Canonical example of MDP from 'Grid World' game.

Algorithm 7 MDP – Value Iteration

Input: Set of states, S; set of actions, A; Transition probabilities between states, $P(s'|s,a)$; rewards/costs, $R(s,a,s')$; γ discount factors

$\forall s: \quad V_0^*(s) = 0$

for $i = 0, \cdots, H-1$ **do**

$\forall s: \quad V_{i+1}^*(s) \leftarrow \max_a \sum_{s'} P(s'|s,a)\left[R(s,a,s') + \gamma V_i^*(s')\right]$ – [Bellman update/back-up] – the expected sum of rewards accumulated when starting from state s and acting optimally for a horizon of $i+1$ steps

end for

FIG. 15: Value Iteration in Grid World.

6. Recitation. Dynamic Programming.

V. SUBJECTS FOR JOURNAL CLUB PRESENTATIONS & REPORTS: (INCOMPLETE) POOL OF OPTIONS

A. General Information

Each student will be required to choose a subject for **journal club presentation and report**. List of suggested subjects is listed below. In terms of picking a subject – the policy is 'first come first served'. Please e-mail the lecturer as soon as possible.

The list is not meant to be complete or exclusive. In particular, the students are encouraged to suggest additional subjects linked to the course material and possibly related to their own research focus/interest. All additional subjects should be discussed with and approved by the lecturer.

Subjects should be presented during the presentation session which will be scheduled for May 24, Tue. Each presentation is 20 mins. All reports should be submitted by May 28, 11:59pm.

Reports are individual, should be at least 10 pages but not longer that 20 pages. Presentations and reports will be graded together. See `http://www.people.fas.harvard.edu/~rpoddar/Papers/ldpc.pdf` for an exemplary student report.

Projects resulting in julia/ijulia programs/illustrations on the subjects linked to the lectures, which can be used as a basis for illustrations in the course in the future, are especially encouraged.

B. Incomplete List of Suggested Subjects

Large Deviation for Multiplicative Processes

Stretching and Rotations of clouds and particles, ordered exponentials, long time statistics of Lyapunov exponents. Cramer/entropy function. `http://arxiv.org/abs/cond-mat/0105199`

The Noisy Channel Coding (Shannon) Theorem

Sec. 9.3 and 10 of [2]

Compressed Sensing and its many uses (How l_1 norm promotes sparcity?)

Pick a review from the extended list available at `https://en.wikipedia.org/wiki/Compressed_sensing` An original option is `http://statweb.stanford.edu/~candes/papers/DecodingLP.pdf`

Slice Sampling MCMC

See `https://en.wikipedia.org/wiki/Slice_sampling`. Recommended review is Neal, Radford M. (2003). "Slice Sampling". Annals of Statistics 31 (3): 705767.

Simulated Annealing Sampling

Important idea and algorithm allowing to explore seriously non-convex problems – rugged landscape with multiple valleys, saddle points, minima and peaks. The original paper is Kirkpatrick, S.; Gelatt Jr, C. D.; Vecchi, M. P. (1983). "Optimization by Simulated Annealing". Science 220 (4598): 671680. See also `https://en.wikipedia.org/wiki/Simulated_annealing` and references there in.

Hamiltonian MCMC

MCMC which is capable to accelerate sampling by adding additional degrees of freedom - related to controlled inertia/momenta expressed through a Hamiltonian description (from physics) — thus the name. Recommended review `http://www.cs.utoronto.ca/~radford/ftp/ham-mcmc.pdf`

Irreversible Monte Carlo algorithms for efficient sampling

The original paper is `http://arxiv.org/abs/0809.0916`.

Warm Algorithm in Classical and Quantum Statistical Physics

The original paper is `http://scholarworks.umass.edu/cgi/viewcontent.cgi?article=2194&context=physics_faculty_pubs`. See also `http://wiki.phys.ethz.ch/quantumsimulations/_media/lecture_101007.pdf`.

Gillespie algorithm

Sampling from stochastic equations (Langevin type) which proceeds by jumps. See the original paper Gillespie, Daniel T. (1977). "Exact Stochastic Simulation of Coupled Chemical Reactions". The Journal of Physical Chemistry 81 (25): 23402361 and also check `https://en.wikipedia.org/wiki/Gillespie_algorithm`.

Sequential Monte Carlo for Importance Sampling & Inference

Recommended paper `https://www.irisa.fr/aspi/legland/ensta/ref/doucet00b.pdf`.

Ising models and Other Graphical Models in Image Analysis

Recommended tutorial `https://www.math.ntnu.no/~joeid/TMA4250/image_ana.pdf`.

Efficient Exact Inference in Planar Ising Model

Recommended paper `http://arxiv.org/pdf/0810.4401.pdf`.

Stochastic Resonances

Curious physics phenomena important in optics & communications which explains how noise/randomness allows to amplify

signal and observe what otherwise would be difficult to detect. Recommended paper is Benzi, R.; Sutera, A.; and Vulpiani, A. "The Mechanism of Stochastic Resonance." J. Phys. A 14, L453-L457, 1981.

Decoding of Low Density Parity Check Codes

Section 47 of [2]. Implementation of a message passing decoding in julia/ijulia is especially encouraged.

Analytic and Algorithmic Solution of Satisfiability Problem

The original paper is `http://cacs.usc.edu/education/cs653/Mezard-RSAT-Science02.pdf` Also check the book of Mezard and Montanari + papers/reviews of Parisi, Mezard and Zechina.

Neural Network Learning

Part V of [2].

Jackson Networks of Queues

Recommended paper is Kelly, F. P. (Jun 1976). "Networks of Queues". Advances in Applied Probability 8 (2): 416432. See alos `https://en.wikipedia.org/wiki/Jackson_network` and references there in. It may also be useful to consult with the recent book: "Stochastic Networks" by E. Yudovina and F. Kelly, Cambridge University Press, 2014. Implementation of a julia/ijulia illustration/program for this subject is especially encouraged.

Path Integral Control & Reinforcement Learning

Recommended review `http://www.snn.ru.nl/~bertk/kappen_granada2006.pdf` Implementation of a julia/ilulia illustration/program for this subject is especially encouraged.

[1] N. van Kampen, *Stochastic Processes in Physics and Chemistry (Third Edition)*, third edition ed. Amsterdam: Elsevier, 2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B9780444529657500003

[2] D. J. C. Mackay, *Information theory, inference, and learning algorithms*. Cambridge: Cambridge University Press, 2003. [Online]. Available: http://www.inference.phy.cam.ac.uk/itprnn/book.html

[3] C. Moore and S. Mertens, *The Nature of Computation*. New York, NY, USA: Oxford University Press, Inc., 2011. [Online]. Available: http://www.nature-of-computation.org/

[4] H. E. Taylor and S. Karlin, *An Introduction to Stochastic Modeling*, 3rd ed. Academic Press, Feb. 1998. [Online]. Available: http://www.ime.usp.br/~fmachado/MAE5709/KarlinTaylorIntrodStochModeling.pdf

[5] C. W. Gardiner, *Handbook of stochastic methods for physics, chemistry and the natural sciences*, 3rd ed., ser. Springer Series in Synergetics. Berlin: Springer-Verlag, 2004, vol. 13.

[6] B. L. Nelson, *Stochastic modeling - analysis and simulation (reprint from 1995)*. Dover Publications, 2002.

[7] E. Cinclar, *Introduction to stochastic processes*. Prentice-Hall, 1975. [Online]. Available: http://gso.gbv.de/DB=2.1/CMD?ACT=SRCHA&SRT=YOP&IKT=1016&TRM=ppn+021423008&sourceid=fbw_bibsonomy

[8] T. Richardson and R. Urbanke, *Modern Coding Theory*. Cambridge University Press, 2008.

[9] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Constructing free-energy approximations and generalized belief propagation algorithms," *Information Theory, IEEE Transactions on*, vol. 51, no. 7, pp. 2282–2312, 2005.

[10] M. J. Wainwright and M. I. Jordan, "Graphical models, exponential families, and variational inference," *Found. Trends Mach. Learn.*, vol. 1, no. 1-2, pp. 1–305, Jan. 2008. [Online]. Available: https://www.eecs.berkeley.edu/~wainwrig/Papers/WaiJor08_FTML.pdf

[11] F. P. Kelly, *Reversibility and stochastic networks*, ser. Wiley series in probability and mathematical statistics. New York, Chichester: Wiley, 1979. [Online]. Available: http://opac.inria.fr/record=b1117013

[12] L. Kleinrock, *Queueing Systems Volume 1: Theory*. Wiley, 1975.

[13] J. F. C. Kingman, "On queues in heavy traffic," *Journal of the Royal Statistical Society. Series B (Methodological) (Wiley)*, vol. 24 (2), p. 383392, 1962.

[14] W. Iglehart, Donald L. Ward, "Multiple channel queues in heavy traffic. ii: Sequences, networks, and batches," *Advances in Applied Probability (Applied Probability Trust)*, vol. 2 (2), p. 355369, 1970.

[15] J. Köllerström, "Heavy traffic theory for queues with several servers i," *Journal of Applied Probability (Applied Probability Trust)*, vol. 11 (3), pp. 544–552, 1974.

[16] D. L. Iglehart, "Limiting diffusion approximations for the many server queue and the repairman problem," *Journal of Applied Probability*, vol. 2, no. 2, pp. 429–441, 1965.

[17] A. A. Borovkov, "On limit laws for service processes in multi-channel systems," *Siberian Mathematical Journal*, vol. 8 (5), pp. 746–763,, 1967.

[18] D. L. Iglehart, "Weak convergence in queueing theory," *Advances in Applied Probability*, vol. 5, no. 3, pp. 570–594, 1973.

[19] The algorithm is justified through a standard Dynamic Programming arguments, of the type discussed above.