ε.δε.μ²
ΕΠΙΣΤΗΜΗ ΔΕΔΟΜΕΝΩΝ & ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

STOCHASTIC PROCESSES & OPTIMIZATION IN MACHINE LEARNING

Metropolis-Hastings Algorithm
Gibbs Sampling
Markov Random Fields, Ising Model, Simulated Annealing

Prof. Vasilis Maglaris
maglaris@netmode.ntua.gr
www.netmode.ntua.gr
Room 002, New ECE Building

Tuesday March 11, 2025

STOCHASTIC PROCESSES & OPTIMIZATION IN MACHINE LEARNING

Evolution of Systems to Thermal Equilibrium: Markov Chain Monte Carlo (MCMC) Simulation Algorithms of Metropolis (1/5)

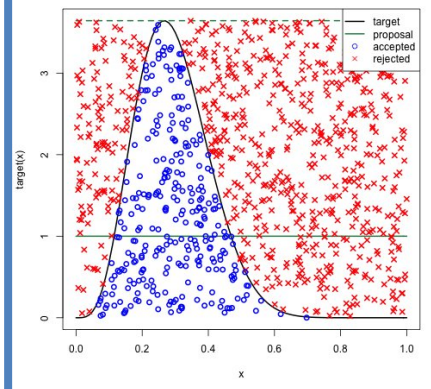
- The aim of ***MCMC simulations*** is to record trajectories of a ***sample element*** (discrete time series) $X_n = x, n = 1,2,...$ as a ***random walk*** of a ***Markov Chain*** (***MC***) state with random yet controllable transitions. The resulting ***sample*** consists of elements approaching a ***Gibbs*** probability distribution
- A ***MCMC*** program ***generates*** trajectories of state x to thermal equilibrium and estimates ergodic probabilities $\pi(x)$ as the limit of relative frequencies $f_n(x)$ in n transitions:

$$\pi(x) = \lim_{n\to\infty} P(X_n = x) = \lim_{n\to\infty} \frac{f_n(x)}{n}$$

Common use of Monte Carlo Simulations:

- Estimation of moments (e.g. averages and 2nd moments) of discrete random variables drawn form a sample space tough (or impossible) to register
- Evaluation of normalization (***partition function)*** for models with intractable possible states

- Sample values generated (***sampled***) via ***MCMC*** may be distributed according to the target ***Gibbs*** distribution but are correlated due to the ***Markov random walk*** generation method
- Sample values ***Independent Identically Distributed*** (***IID***) can be sampled using simple methods, e.g. random generation of multi-dimensional points and counting those under the histogram curve to estimate metrics such as area under the curve (***Rejection Sampling***)



https://en.wikipedia.org/wiki/Rejection_sampling#Adaptive_rejection_sampling

STOCHASTIC PROCESSES & OPTIMIZATION IN MACHINE LEARNING

Evolution of Systems to Thermal Equilibrium:
Markov Chain Monte Carlo (MCMC) Simulation Algorithms of Metropolis (2/5)

Approach & Limitations

- Generate a ***Target*** distribution $\pi(x)$ proportional to the relative frequency of values assumed by the random variable (or vector) X_n over the sample space: $\sum_x \pi(x) = \sum_x P(X_n = x) = 1$

- $\pi(x)$ has a given form $\pi(x) \propto \tilde{\pi}(x)$, e.g. ***Gibbs*** distribution $\pi(x) \propto \exp\left(-\frac{E(x)}{T}\right)$, but cannot generate samples directly as it may experience difficulties in registering the entire state space and computing the normalizing ***partition function***

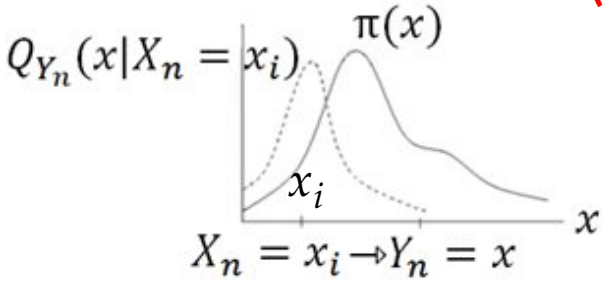
- The goal is to generate ***sample elements*** $X_n = x$, $n = 1,2,...$ with frequency of appearances $\tilde{\pi}(x)$, proportional to the target probability distribution $\pi(x) \propto \tilde{\pi}(x)$

- The approximation of relative frequencies (histogram) $P(X_n = x) \to \pi(x)$ is accomplished by ***simulating*** state transitions as a ***random walk*** of an ergodic ***Markov Chain*** with appropriately set transition probabilities. This technique, referred to as ***Markov Chain Monte Carlo*** (**MCMC**) simulation, records the number of visits to states $\{X_n\}$ in trajectories and thus approximates the distribution $\pi(x)$

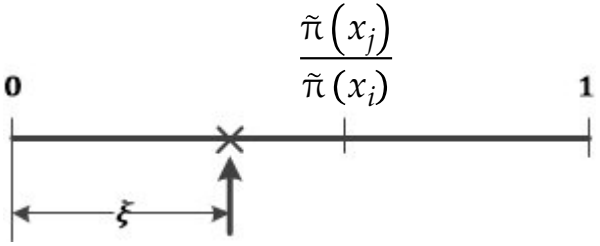
The ***MCMC methods*** may be powerful to sample random variables given a ***histogram*** or to approximate complicated ***partition functions***. However, they generate ***correlated*** Markovian time series, a problem when ***Independent Identically Distributed*** (***IID***) sampling is required

STOCHASTIC PROCESSES & OPTIMIZATION IN MACHINE LEARNING

Evolution of Systems to Thermal Equilibrium:
Markov Chain Monte Carlo (MCMC) Simulation Algorithms of Metropolis (3/5)



$$\pi(x) \propto \tilde{\pi}(x)$$



- Given the form $\tilde{\pi}(x)$ of ***ergodic target distribution*** $\pi(x) \propto \tilde{\pi}(x)$ for $x \in \{0,1,2,...\}$ we seek the ***generation*** of a time-reversible **Markov Chain** (***MC***) $X_n = x$, approximately distributed according to $\pi(x)$
- Let $X_n = x_i$. Generate a ***random variable*** $Y_n = x_j$ distributed according to an **arbitrary** but **symmetric** around x_i ***Proposal Conditional Density*** $Q_{Y_n}(x|X_n = x_i)$:

$$P\left(Y_n = x_j \middle| X_n = x_i\right) = P\left(Y_n = x_i \middle| X_n = x_j\right)$$

- For the candidate next state $Y_n = x_j$ of the ***MC*** random walk:
  - If $\tilde{\pi}\left(x_j\right) \geq \tilde{\pi}\left(x_i\right)$ the transition $X_n \rightarrow Y_n$ is accepted $\Rightarrow X_{n+1} = Y_n = x_j$
  - If $\tilde{\pi}\left(x_j\right) < \tilde{\pi}\left(x_i\right)$ generate a random number ξ, uniformly distributed in $(0,1)$

    If $\xi < \frac{\tilde{\pi}\left(x_j\right)}{\tilde{\pi}\left(x_i\right)} = \frac{\pi\left(x_j\right)}{\pi\left(x_i\right)}$ the transition $X_n \rightarrow Y_n$ is accepted $\Rightarrow X_{n+1} = Y_n = x_j$

    Else the state **remains at the same value** $\Rightarrow X_{n+1} = x_i$
- Caution in selecting the ***initial state*** $X_0 = x$ and the symmetric ***Proposal Conditional Density*** $Q_{Y_n}(x|X_n = x_i)$: It should cover a wide range of x so that the ***MC*** random walk would not be ***trapped*** within subsets of states that rarely cross communicate.

Common choices: (1) **Gauss** with average $\mu = x_i, \sigma^2 = 1$, (2) **Uniform Distribution** $x \in (x_i - a,\ x_i + a)$

STOCHASTIC PROCESSES & OPTIMIZATION IN MACHINE LEARNING

Evolution of Systems to Thermal Equilibrium:
Markov Chain Monte Carlo (MCMC) Simulation Algorithms of Metropolis (4/5)

Generation of Random Samples Thermal Equilibrium Probabilities

The ***Metropolis*** algorithm generates via simulation a time-reversible Markov Chain (***MC***) as a random walk $X_n = x_i$ distributed according to the ***Gibbs*** thermal equilibrium probabilities $\pi(x_i) \propto \exp\left(-\frac{E(x_i)}{T}\right)$. Their exact values depend on the ***Partition Function*** Z that satisfies the normalization $\sum_i \pi(x_i) = \frac{1}{Z}\sum_i \exp\left(-\frac{E(x_i)}{T}\right) = 1$ in a usually complex state-space, rendering its evaluation a combinatorially hard (**NP-Complete**) problem

Using the ***Metropolis Algorithm*** we generate a time-reversible ***MC*** as a random walk of random variables X_n satisfying the ***detailed balance equations*** that after n transistions (steps) lead to state x_i. Using a random variable generator we subsequently produce a state x_j of another process Y_n with symmetric transition probabilities $X_n \to Y_n$:

$$P\left(Y_n = x_j \middle| X_n = x_i\right) = P\left(Y_n = x_i \middle| X_n = x_j\right)$$

This transition would result into ***energy change*** $\Delta E = E_j - E_i$

- If $\Delta E < 0$ the transition leads to an acceptable lower energy state $(Y_n = x_j)$: $X_{n+1} := Y_n$
- If $\Delta E > 0$ the transition to $(Y_n = x_j)$ is accepted with probability $\exp(-\frac{\Delta E}{T})$, with T the "temperature" (or external control) of the environment and $X_{n+1} := Y_n$. Else $X_{n+1} := X_n$. The evolution of the random walk depends on independent ***Bernoulli Trials*** (accept transition or not) is based on the (pseudo)random number ξ uniformly distributed in $(0,1)$:

$$\text{If } \xi < \exp(-\tfrac{\Delta E}{T}),\ X_{n+1} := Y_n;\ \text{else } X_{n+1} := X_n$$

STOCHASTIC PROCESSES & OPTIMIZATION IN MACHINE LEARNING

Evolution of Systems to Thermal Equilibrium:
Markov Chain Monte Carlo (MCMC) Simulation Algorithms of Metropolis (5/5)

Choice of Transition Probabilities

- Generate a new ***MC*** with ***arbitrary symmetric transition probabilities*** τ_{ij} from $X_n = x_i$ to a new state $Y_n = x_j$:

$$\tau_{ij} = P\left(Y_n = x_j \middle| X_n = x_i\right) = P\left(Y_n = x_\iota \middle| X_n = x_j\right) = \tau_{ji}$$

$$\tau_{ij} \geq 0,\ \forall i,j \text{ and } \sum_j \tau_{ij} = 1, \forall i,$$

- The ***MC*** X_n will follow transition probabilities p_{ij} related to τ_{ij} as follows:

$$p_{ij} = P\left(X_{n+1} = x_j \middle| X_n = x_i\right) = \begin{cases} \tau_{ij}\frac{\pi_j}{\pi_i} = \tau_{ij}\ \exp(-\frac{E_j - E_i}{T}) & \text{for } \frac{\pi_j}{\pi_i} < 1 \\ \tau_{ij} & \text{for } \frac{\pi_j}{\pi_i} \geq 1 \end{cases}, \quad i \neq j$$

and with p_{ii} consistent with the normalization equation $\sum_j p_{ij} = 1, \forall i$:

$$p_{ii} = \tau_{ii} + \sum_{j \neq i} \tau_{ij}\left(1 - \frac{\pi_j}{\pi_i}\right)$$

- All p_{ij} above abide with the ***detailed balance equations***, yielding to ***Gibbs target*** probabilities $\pi_j = \frac{1}{Z}\exp\left(-\frac{E_j}{T}\right)$

The transition probabilities p_{ij} above depend only on **ratios** $\frac{\pi_j}{\pi_i} = \exp\left(-\frac{E_j - E_i}{T}\right) = \exp\left(-\frac{\Delta E}{T}\right)$ bypassing the need of ***partition function*** Z evaluation (often an **NP-Complete Problem**)

STOCHASTIC PROCESSES & OPTIMIZATION IN MACHINE LEARNING

Generalization of the Metropolis Algorithm: The Metropolis-Hastings Algorithm (1/2)

- The ***MCMC*** algorithm (by ***Nicholas Metropolis*** et.al., ***1953***) assumes symmetric transitions and leads to ***time reversible*** chains that satisfy ***detailed balance equations.*** It converges to ***Gibbs*** (***Boltzmann***) state probabilities $\pi_i = \frac{1}{Z}\exp\left(-\frac{E_i}{T}\right)$ with proper definitions of E_i
- It was generalized by ***W.K. Hastings*** in 1970 to the ***Metropolis-Hastings*** (***MH***) algorithm
  - ***Proposal Conditional Density***:
    No symmetry is required in the (arbitrary) selection of $Q_{Y_n}(x|X_n = x_i)$
    $$P\left(Y_n = x_j \middle| X_n = x_i\right) \neq P\left(Y_n = x_i \middle| X_n = x_j\right)$$
  - ***Probability of Acceptance of Transition*** $X_{n+1} = x_j$:
    - If $\tilde{\pi}\left(x_j\right) \times P\left(Y_n = x_i \middle| X_n = x_j\right) \geq \tilde{\pi}\left(x_i\right) \times P\left(Y_n = x_j \middle| X_n = x_i\right)$ transition $X_n \to Y_n$ is accepted and $X_{n+1} = x_j$
    - Else, generate a random number ξ uniformly distributed in $(0,1)$
      - ✓ If $\xi < \frac{\tilde{\pi}\left(x_j\right)}{\tilde{\pi}\left(x_i\right)} \times \frac{P\left(Y_n = x_i \middle| X_n = x_j\right)}{P\left(Y_n = x_j \middle| X_n = x_i\right)}$ the choice $X_n \to Y_n$ is accepted & $X_{n+1} = x_j$
      - ✓ If not X_n remains in its current value and $X_{n+1} = x_i$

- The ***Metropolis*** algorithm is a special case of ***MH*** with symmetric $Q_{Y_n}(x|X_n = x_i)$

STOCHASTIC PROCESSES & OPTIMIZATION IN MACHINE LEARNING

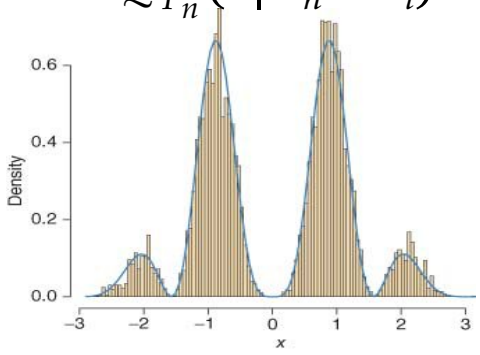
Generalization of the Metropolis Algorithm: The Metropolis-Hastings Algorithm (2/2)

Shortcoming of the *MH* Algorithm

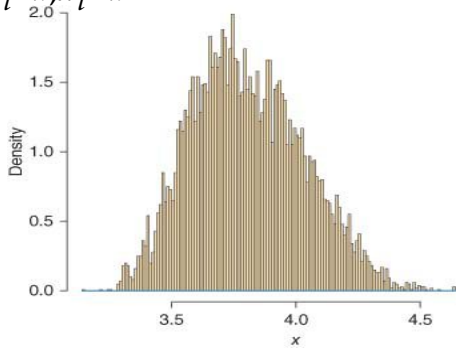
- The ***MH*** algorithm generates a **Markov Chain** with ergodic probabilities as specified in a given ***target histogram*** but the random variables of the sample are ***correlated***
- If the requirement is to generate multi-dimensional ***sample random vectors*** $\mathbf{X}_n$, the algorithm may suffer by the ***curse of dimensionality***
- The selection of ***Proposal Conditional Density*** $Q_{Y_n}(x|X_n = x_i)$ and ***initial state*** $X_0 = x_i$ may be of paramount importance for correct and fast convergence of the ***MH*** algorithm
  https://onlinelibrary.wiley.com/doi/full/10.1002/9781118445112.stat07834
- The influence of X_0 diminishes if we ignore transient states, e.g. for steps $n \leq 1000$

Example: Simulation of X_n conforming to a ***target distribution*** $\pi(x)$ proportional to $\tilde{\pi}(x) = \sin^2(x) \times \sin^2(2x) \times \varphi(x)$ with $\varphi(x)$ ***Normal*** $\mathbb{N}(0,1)$

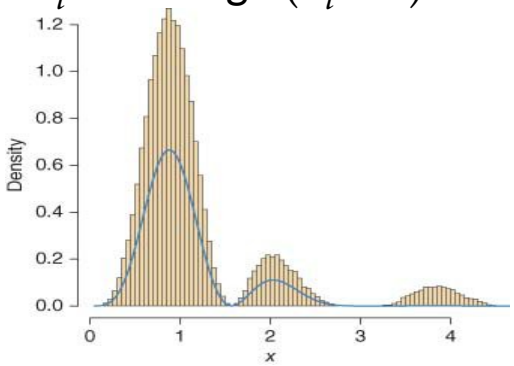
Select $Q_{Y_n}(x|X_n = x_i) = \frac{1}{2a}\mathbb{1}_{x_i-a,x_i+a}(x)$, uniform with average x_i and range $(x_i \pm a)$



$a = 1, X_0 = 3.14$
Correct convergence in 10^4steps



$a = 0.1, X_0 = 3.14$
Failure in 10^4 steps
(State trapped in 1 lobe)



$a = 0.2, X_0 = 3.14$
Partial failure in 10^4steps
(State trapped in positive values)

STOCHASTIC PROCESSES & OPTIMIZATION IN MACHINE LEARNING

Gibbs Sampling (1/2)

- Generates via ***Monte Carlo*** simulations sample vectors of **large dimensionality** exhibiting ***Gibbs*** ergodic probabilities. However, vectors are correlated due to ***Markovian*** transitions
- The ***Gibbs Sampling*** method is a variant of the ***Metropolis*** iterative algorithm eventually leading to lower energy states, but allowing some opposite direction transitions with diminishing probability as convergence is being approached
- Every iteration proceeds along each coordinate of the vector sample space. Transition probabilities for a specific coordinate depend on the **most recent values** of all other coordinates evaluated at the present iteration, **excluding the coordinate under consideration**
- It converges to ***Gibbs distributed samples***, by estimating ***joint*** or ***marginal*** ergodic probabilities as relative frequency of visits to states during the simulation time horizon

Some Uses of Gibbs Sampling

- Efficient generation of sample vectors with ***Gibbs target distribution***
- Complementing vectors containing non-observed or distorted coordinates assuming ***Gibbs*** distributions (e.g. ***hidden variables*** in ***Deep Neural Networks***)
- Estimation of functions of a variable that depends on ***Gibbs*** sample vector **single** coordinate, e.g. on the mean of values in a specific coordinate

STOCHASTIC PROCESSES & OPTIMIZATION IN MACHINE LEARNING

Gibbs Sampling (2/2)

- **Sample Space**: ***Random Vectors*** of dimension K at ***Iteration Step*** n of the ***Random Walk***

$$\mathbf{X}(n) = [X_1(n)\ \ X_2(n)\ \ \ldots\ X_K(n)]^{\mathrm{T}}$$

- **State Vector**: Values $x_i(n)$ assigned to the K ***Random Variables*** $X_i(n)$ of the coordinates of the random vector $\mathbf{X}(n)$

$$\mathbf{x}(n) = [x_1(n)\ \ x_2(n)\ \ \ldots\ x_K(n)]^{\mathrm{T}}$$

Algorithm (***Stuart*** & ***Donald Geman***, 1984)

- $n = 0$: Start from an arbitrary initial state vector $\mathbf{x}(0) = [x_1(0)\ \ x_2(0)\ \ \ldots\ x_K(0)]^{\mathrm{T}}$
- $n \to n+1$: For $i = 1,\ldots,K$ generate state values $x_i(n+1)$ for $X_i(n+1)$ with pseudo-random algorithm based on the conditional probability

$$P\left[X_i(n+1)\,|\{x_1(n+1)\ldots x_{i-1}(n+1)\ \ x_{i+1}\ (n)\ \ \ldots\ x_K(n)\}\right]$$

- **Note 1**: The condition involves coordinates $j \neq i$ evaluated up to this step and not $X_i(n)$
- **Note 2**: The ***conditional probabilities*** can be inferred from joint probabilities (***Bayes*** rule) and the system (environment) specifications

https://www2.stat.duke.edu/~rcs46/modern_bayes17/lecturesModernBayes17/lecture-7/07-gibbs.pdf

- The sequence $x_i(n)$, $i = 1,\ldots,K$ forms a ***Markov Chain*** that converges (geometrically) to the i^{th} coordinate of state $\mathbf{x} = [x_1\ \ x_2\ \ \ldots\ x_K]$ at thermal equilibrium, with ***marginal*** probability

$$\lim_{n\to\infty} P\left(X_i(n) = x_i(n)|x_i(0)\right) = P(X_i = x_i)$$

The state vector $\mathbf{X}(n)$ converges to and ***Gibbs*** joint probabilities

$$\lim_{n\to\infty} P(\mathbf{X}_n = \mathbf{x}_n) = P(X_1 = x_1, \ldots, X_K = x_K) = P(\mathbf{X} = \mathbf{x}) = \frac{1}{Z}\exp\left(-\frac{E(\mathbf{x})}{T}\right)$$

Machine Learning Applications: ***GenAI*** Models - ***Boltzmann Machine*** (BM), ***Restrictive*** BM

STOCHASTIC PROCESSES & OPTIMIZATION IN MACHINE LEARNING

Some Applications of MCMC Simulation Methods

Sampling of Random Walks Leading to Thermal Equilibrium

The ***Metropolis*** algorithm generates ***correlated random walks*** across states X_n of a ***time reversible*** Markov Chain with ergodic state probabilities specified by a target ***Gibbs*** (***Boltzmann***) distribution, without using the ***partition function*** Z. As mentioned earlier it is often the case that states cannot be fully specified or accounted, especially for sample spaces of multi-dimensional vectors, and hence normalization of state probabilities is a non tractable problem

Computations of Integrals and Statistical Parameters

The ***MCMC*** methods can be used for numerical evaluation of ***integrals*** and ***probability moments*** of multi-dimensional random variables (vectors) with missing elements or distorted values due to noise or other limitations in exact specification of sample elements

Search for Global Extremal Points, Simulated Annealing

The ***MCMC*** iterative algorithms enable state visits that do not appear attractive in a myopic view. Such behavior is typical in ***deepest descent*** algorithms that could be trapped in local minima and miss global optimality. In ***MCMC*** such unlikely states are not fully ignored but are registered with some reduced chance. This forms the basis of ***Simulated Annealing*** applied in complex problems of combinatorial optimization, e.g. the ***Travelling Salesman Problem***

SEARCH FOR GLOBAL EXTREMAL POINTS

In searching for an extremal point (e.g. highest ***hill climbing***), we allow for some steps to what myopically appears as the wrong direction, but eventually may lead to ***global*** minima or maxima

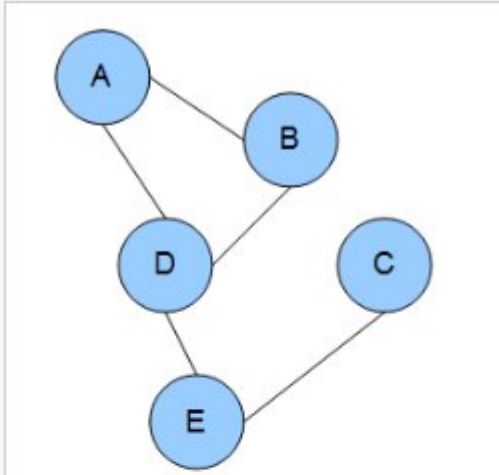
https://en.wikipedia.org/wiki/File:Hill_Climbing_with_Simulated_Annealing.gif

STOCHASTIC PROCESSES & OPTIMIZATION IN MACHINE LEARNING

Markov Random Fields - The *Ising* Model (1/2)

https://en.wikipedia.org/wiki/Markov_random_field

- A ***Markov Random Field*** (***MRF***) is an undirected graphical model of joint probability distribution with nodes representing random variables and edges denoting bi-directional dependence among neighbors. Applications of ***MRF*** in image processing include (i) texture classification models and (ii) in ***belief networks*** to estimate probability distribution for a subset of nodes (***hidden neurons***) given values of another subset (***visible neurons***)



An example of a Markov random field. Each edge represents dependency. In this example: A depends on B and D. B depends on A and D. D depends on A, B, and E. E depends on D and C. C depends on E.

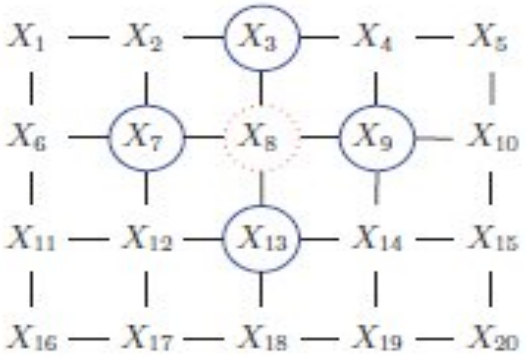
- The ***Ising*** model is an ***MRF*** case that analyzes ferromagnetism in statistical mechanics; it can potentially predict phase transitions. The model consists of binary discrete random variables representing magnetic dipole moments of "spins" in one of two states $\{-1, +1\}$. The spins interact according to a graph allowing each spin to interact with its neighbors. It was first developed for chain configurations by ***Ernst Ising*** in his 1925 Thesis, with no phase transitions identified. These were later identified in 1944 for ***lattice*** two-dimensional configurations (commonly used as a graph topology since then)

STOCHASTIC PROCESSES & OPTIMIZATION IN MACHINE LEARNING

Markov Random Fields - The *Ising* Model (2/2)

- The ***Ising*** model consists of a graph G of bipolar nodes and edges representing mutual interactions. A nodes s is in a binary state (***spin***) $y_s \in \{-1, +1\}$. If $s \leftrightarrow t$, nodes s, t are neighbors with interaction $J_{s,t}$: $J_{s,t} = +1$ if it pushes y_s και y_t to the same spin (***ferromagnetic***) or $J_{s,t} = -1$ if it pushes to opposite spins (***antiferromagnetic***)
- In the lattice ***MRF*** G, the state of s directly depends on its neighbors $t \in N(s) \subset G$. Then the equilibrium state vector is $\boldsymbol{y}(G) = [y_1\ y_2 ... y_s\ y_t ...]^{\mathrm{T}}$ for feasible ***configurations*** of node states y_s as they result from transition probabilities $P(y_s | \boldsymbol{y}(G)) = P(y_s | \boldsymbol{y}(N(s)))$
  https://en.wikipedia.org/wiki/Ising_model

The state of X_8 depends only on its neighbors X_3, X_7, X_9, X_{13}



- In thermal equilibrium $\boldsymbol{y}(G)$ converges to ***Gibbs*** (***Boltzmann***) ergodic probabilities, possibly with external magnetic influence h_t on nodes t:

$$P(\boldsymbol{y}(G)) = \frac{1}{Z}\exp\left(-\frac{E(\boldsymbol{y}(G))}{T}\right) \text{με } E(\boldsymbol{y}(G)) \approx -\sum_{s\leftrightarrow t} J_{s,t} y_s y_t - \mu \sum_t h_t y_t$$

- If $J_{s,t} = J$ for all neighboring pairs $s \leftrightarrow t$ then $E(\boldsymbol{y}(G)) \approx -J\sum_{s\leftrightarrow t} y_s y_t - \mu \sum_t h_t y_t$

  Convergence to equilibrium and approximation of the partition function Z may be handled via the ***Metropolis-Hastings*** algorithm (if $J_{s,t} = J > 0$ convergence leads to the ***ferromagnetic*** phase)

STOCHASTIC PROCESSES & OPTIMIZATION IN MACHINE LEARNING

Simulated Annealing (1/2)

Motivation - Physics Case Study

Simulated Annealing was introduced in modeling of multiple particle physical system behavior as a ***Markov Chain*** convergence to equilibrium with lowering the environment temperature to a target low temperature T

- The system converges to ***Gibbs*** probabilities $p_i = \frac{1}{Z}\exp\left(-\frac{E_i}{T}\right)$ as $T \to 0$, with lower energy states exhibiting higher frequency of occurrence

The ***Metropolis*** algorithm is ***quick*** to converge towards equilibrium states at high temperature T but ***slow*** at low temperatures. This motivated its variant via the ***Simulated Annealing*** algorithm that comprises two modules:

- A ***cooling schedule*** towards the target low temperature $T \to 0$ via iterative cooling $T_0, T_1,\ldots,T_k,\ldots,T$ at diminishing steps (e.g. $T_k = \alpha T_{k-1}$, $0.8 < \alpha < 0.99$)
- Identification of equilibrium states via the ***Metropolis*** algorithm, initially for a high temperature T_0 that would entail a large range of state transitions and a reasonable number of transitions (e.g. 10 iterations) to convergence. For the next round at a lower $T_k \to T_{k+1}$ start at an initial state equal to the final state of the previous round T_k

The ***Simulated Annealing*** algorithm may be trapped In **local minima**. Convergence to the ***global minimum*** is not guaranteed for realistic cooling schedules and several repetitions may be required with different schedules and hyperparameters

STOCHASTIC PROCESSES & OPTIMIZATION IN MACHINE LEARNING

Simulated Annealing (2/2)

Application in Combinatorial Optimization

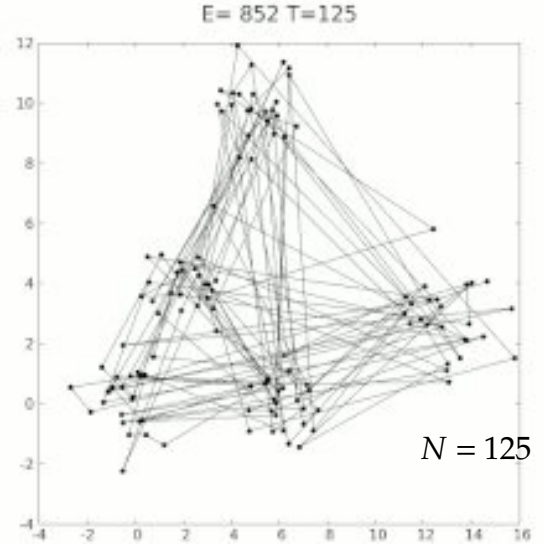
The quest for thermodynamic equilibrium at low "temperatures" is translated to cost minimization in complex problems of ***Combinatorial Optimization***, especially if they involve several local minima and a great number of states

- The state "energies" E_i correspond to the numerical cost of discrete states. Temperature T is an external control parameter that starts from a high initial value. In every subsequent round (epoch) the temperature diminishes $T \to \alpha T$ by a ***hyperparameter*** α, $0 < \alpha < 1$ until $T \cong 0$
- In every round initiated by a lower temperature T we run the ***Metropolis*** algorithm, starting from the final state of the previous round and searching for a new equilibrium state with ***Gibbs*** probabilities $p_i = \frac{1}{Z}\exp\left(-\frac{E_i}{T}\right)$ for E_i and T. Acceptance of some seemingly wrong transitions $i \to j$ to higher cost states $E_j > E_i$ with non-zero probability $\exp\left(-\frac{E_j - E_i}{T}\right)$, may avert **entrapment** to local **minima**. As $T \to 0$ this probability is being reduced and the algorithm **assumes** that it reached the ***global minimum***

Traveling Salesman Problem: Infamous NP-Complete Combinatorial Optimization Problem

Consider a graph of N nodes (cities) with travel cost c_{ij} between nodes i και j. Find the minimum total cost that a ***Traveling Salesman*** has to pay to visit all nodes just once

https://en.wikipedia.org/wiki/File:Travelling_salesman_problem_solved_with_simulated_annealing.gif



STOCHASTIC PROCESSES & OPTIMIZATION IN MACHINE LEARNING

Aanalogies of Statistical Physics & Combinatorial Optimization

Simulated Annealing and **Random Markov Fields** – **Ising Model** inspired ***Combinatorial Optimization*** algorithms, by seeking analogies with ***Statistical Physics*** terms (e.g. transitions to thermal equilibrium states ***Gibbs/Boltzmann*** as they evolve in ***Monte Carlo Markov Chain*** simulations e.g. ***Metropolis-Hastings***)

Correspondence of Statistical Physics and Combinatorial Optimization

- Search for multi-dimensional state vectors of ***minimum cost*** or ***energy*** that a model with stochastic transitions converges. Convergence can be accelerated by exercising external control parameters (e.g. ***Bias*** in Machine Learning Systems and Neural Networks, ***Temperature*** in Physical Systems)
- Optimization algorithms do not necessarily proceed to steps that may temporarily reduce cost (or increase rewards). They rather follow random trajectories (e.g. ***Markov Random Walk*** samples) and may visit a large range of states in order to bypass local minimum (or maximum)

TABLE 11.1 Correspondence between Statistical Physics and Combinatorial Optimization

Statistical physics	Combinatorial optimization
Sample	Problem instance
State (configuration)	Configuration
Energy	Cost function
Temperature	Control parameter
Ground-state energy	Minimal cost
Ground-state configuration	Optimal configuration