Bayesian Networks and Low Rank Structures (Part 1)

Dmitry P. Vetrov

Head of Bayesian methods research group, MSU, HSE.

Outline

Part 1

- General information
- Probabilistic modeling in machine learning
- Exponential family of distributions
- Probabilistic graphical models

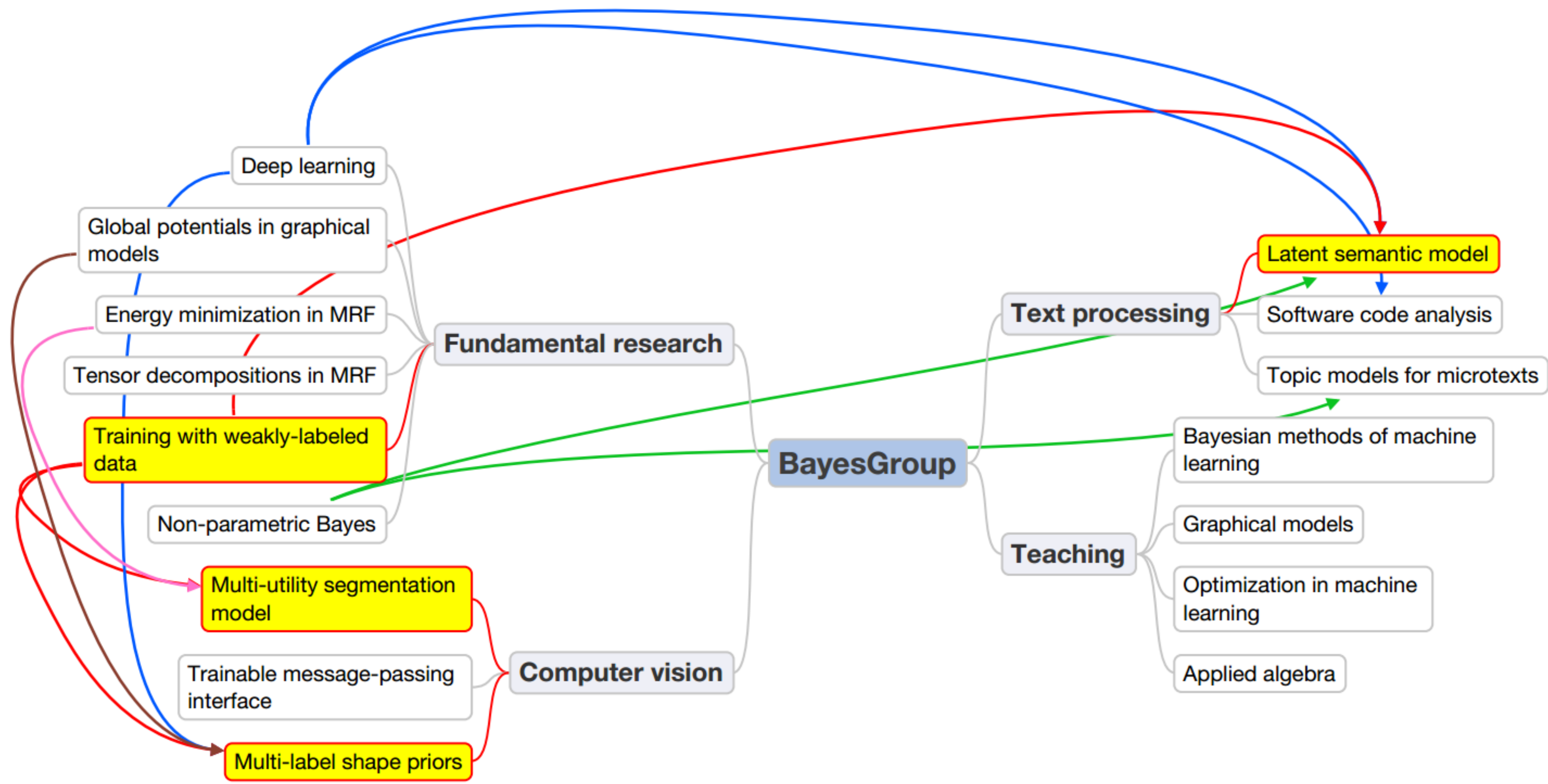
Part 2

- Markov random fields
- Tensor decomposition of MRF energy
- Partition function estimation via TT

Bayesian methods research group

Founded in 2007. Currently consists of 8 students, 5 PhD students, 1 researcher and 1 associate professor.

Bayesian methods research group



What is machine learning?

- ML tries to find regularities within the data
- Data is a set of objects (users, images, signals, RNAs, chemical compounds, credit histories, etc.)
- Each object is described by a set of observed variables X and a set of hidden (latent) variables T
- It is assumed that the values of hidden variables are hard to get and we have only limited number of objects with known hidden variables, so-called training set
- The goal is to find the way of predicting the hidden variables for a new object given the values of observed variables

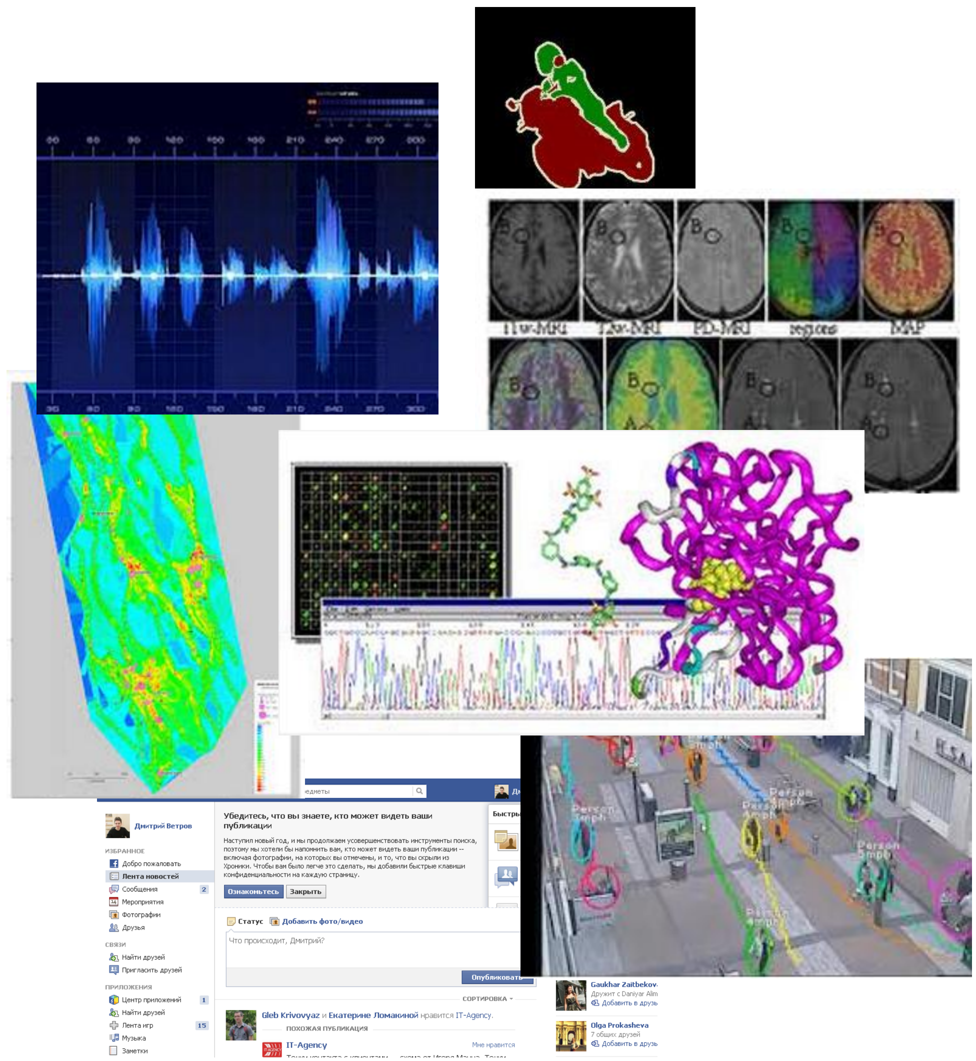
Example: Credit scoring

- Objects: clients in bank
- Observed variables: gender, age, income, family status, eduction, credit history, etc.
- Hidden variables: credit limit, to give or not to give credit.
- Training set: history of our credit operations from past

Areas of application

With the spread of information technologies ML has been used in more and more domains

- Computer vision
- Speech recognition
- Credit scoring
- Mineral deposits search
- Bioinformatics
- Web-search
- Sells forecasts
- Behaviour analysis
- Social networks
- etc.



Stages

90s. Support vector machines. Linear methods for constructing non-linear decision rules

90-00s. Bayesian framework. Encodes prior knowledge about the concrete problem into the model

00s. Probabilistic graphical models. Construct complex models using simple Bayesian models as building blocks

00-10s Deep revolution. 2^{nd} reincarnation of neural networks. This time a successful one

10s. Big Data. ...

20s. Artificial intelligence?..

Today we have a boosting development of ML techniques due to the unprecedented amounts of available data and computational resourses

Overfitting effect

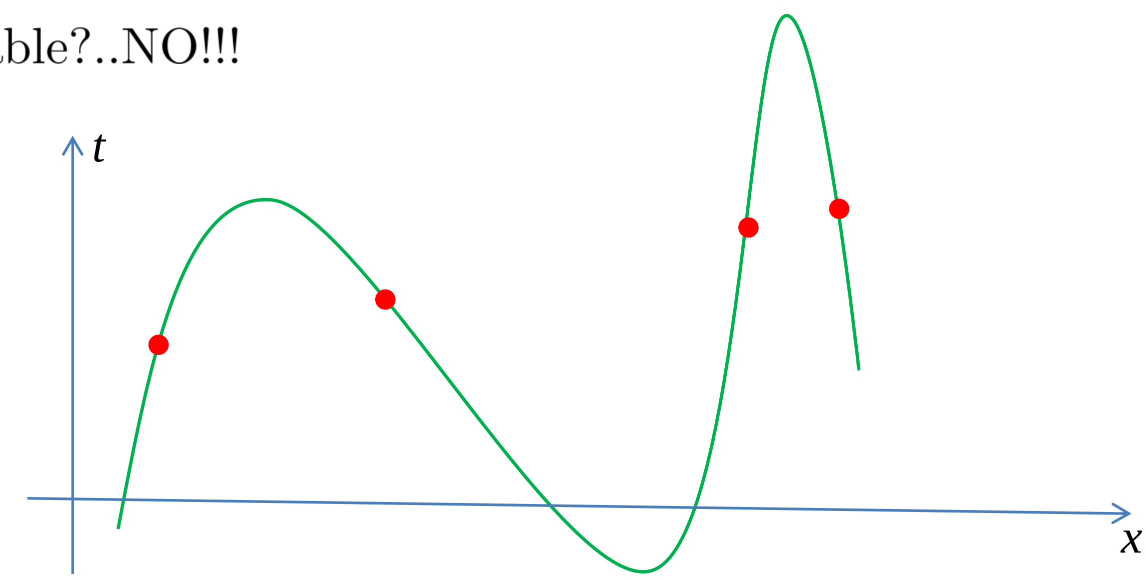
- Imagine we are given a trainig set $(X_{tr}, T_{tr}) = \{(x_i, t_i)\}_{i=1}^n$ and a parameterized set of possible prediction algorithms $\{f(x, \theta) \mid \theta \in \Theta\}$
- We select

$$\theta^* = \arg\min_{\theta \in \Theta} \sum_{i=1}^{n} \rho(t_i, f(x_i, \theta)),$$

  and use $f(x, \theta^*)$ for predicting the value of hidden variable for object x
- Seems reasonable?..

Overfitting effect

- Imagine we are given a trainig set $(X_{tr}, T_{tr}) = \{(x_i, t_i)\}_{i=1}^n$ and a parameterized set of possible prediction algorithms $\{f(x, \theta) \mid \theta \in \Theta\}$
- We select
$$\theta^* = \arg\min_{\theta\in\Theta} \sum_{i=1}^n \rho(t_i, f(x_i, \theta)),$$
and use $f(x, \theta^*)$ for predicting the value of hidden variable for object x
- Seems reasonable?..NO!!!



Occam's razor

- In 14th century William Occam fomulated his famous principle: among all explainations of the event you need to seek for the simplest one
- Occam razor has become the methodological basis of modern scientific method
- We use this principle informally in everyday's life
- PROBLEM: Computer can't distiguish between simple and complex explainations of training set

What is simple?

- From psychology: ”Complex explaination” = ”Unexpected explaination”
- From information theory: ”Unexpected” = ”Less probable”
- Shannon theorem provides an explicit way of formalizing our surprize in terms of a distribution
- The more complex the dependency is the less probable it should be
- We may now use probabilistic language to formalize Occam razor!

Bayes theorem

- In probabilistic setting we try to recover $p(t|x,\theta)$ wrt training set
- Maximum likelihood estimation

$$\theta^* = \arg\max_\theta p(T_{tr}|X_{tr},\theta) = \arg\max_\theta \prod_{i=1}^n p(t_i|x_i,\theta)$$

  tends to overfit

- We may encode the complexity of dependence in terms of prior distribution $p(\theta)$
- Famous Bayes theorem (1763) provides a correct way of transforming our knowledge from prior to posterior form

$$p(\theta|X_{tr},T_{tr}) = \frac{p(T_{tr}|X_{tr},\theta)p(\theta)}{\int p(T_{tr}|X_{tr},\theta)p(\theta)d\theta}$$

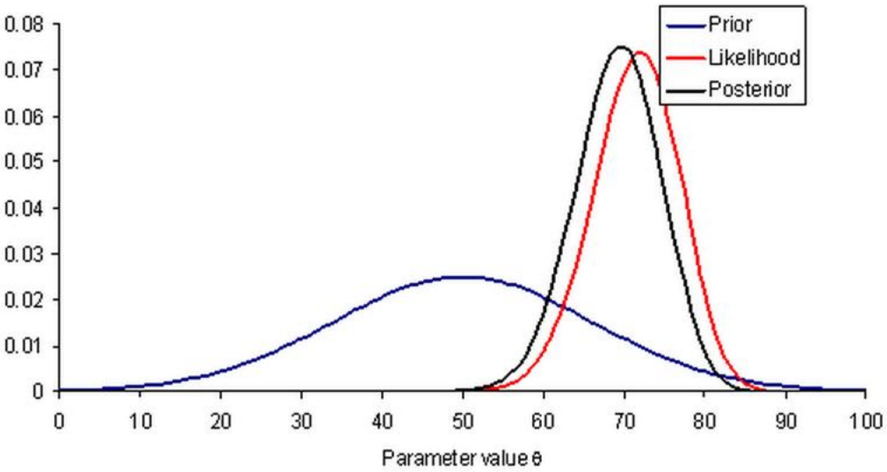
- See "Harry Potter and the Methods of Rationality", Chapter 20

Bayesian world

- In Bayesian world everything is random!

$$\texttt{Posterior} = \frac{\texttt{Likelihood} \times \texttt{Prior}}{\texttt{Evidence}}$$

- New interpretation of randomness: "Objective uncertainty" $\rightarrow$ "Subjective ignorance"
- In Bayesian modeling we estimate $p(T, \theta | X)$ instead of $p(T | X, \theta)$
- Rather than getting point estimate for θ we obtain posterior $p(\theta | X_{tr}, T_{tr})$ that **can be used as a prior** in next model
- Thus we can construct complex models from simpler ones



Example: logistic regression

- Training set: $(X_{tr}, T_{tr}) = \{(x_i, t_i)\}_{i=1}^n$, where $x_i \in \mathbb{R}^d$, $t_i \in \{0, 1\}$
- Discriminative model

$$p(T, \theta | X) = \prod_{i=1}^{n} p(t_i | x_i, \theta) p(\theta)$$

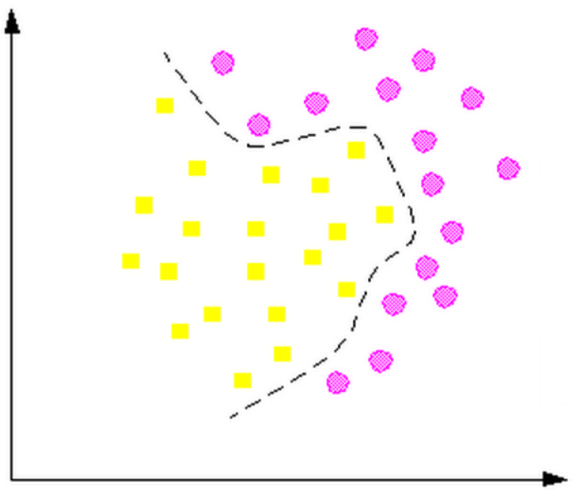
- Likelihood term is defined as follows

$$p(t_i | x_i, \theta) = \frac{1}{1 + \exp(-t_i \theta^T x_i)}$$

- Prior usually penalizes large weights, e.g.

$$p(\theta) \sim \mathcal{N}(0, \lambda^{-1}),$$

where $\lambda > 0$ is **regularization coefficient**



Exponential family of distributions

- Plays important role in Bayesian (and not only!) regularization and learning
- Functional rather than parametric family

$$p(x|\theta) = \frac{1}{h(\theta)} f(x) \exp(\theta^T u(x))$$

- Key observation: log-linear model
- Factorization criteria: if

$$p(x|\theta) = f_1(x) f_2(\theta) f_3(u(x), \theta)$$

  then (and only then) $u(x)$ are **sufficient statistics** of $p(x|\theta)$

- Sufficient statistics conatin **all** information that is nessesary for estimating θ

Examples

- The most of "table distributions" are from exponential family: Gaussian, Gamma, Beta, Chi-squared, Wishart, Von Mises, **all discrete**, etc.

- Sometimes is it not easy to see that a distribution can be reduced to the form $p(x|\theta) = h^{-1}(\theta)f(x)\exp(\theta^T u(x))$

- Consider 1-dimensional Gaussian distribution

$$\mathcal{N}(\mu, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-0.5\sigma^{-2}x^2 + \sigma^{-2}\mu x - 0.5\sigma^{-2}\mu^2\right)$$

Denoting $\theta_1 = -0.5\sigma^{-2}$ and $\theta_2 = \sigma^{-2}\mu$ we get

$$\sqrt{\frac{-2\theta_1}{2\pi}}\exp\left(\theta_2^2\theta_1^{-1}\right)\exp\left(\theta_1 x^2 + \theta_2 x\right)$$

- Hence $u_1(x) = x$ and $u_2(x) = x^2$ are sufficient statistics. Parameters θ are called **natural parameters**

Normalization constant

- Function $h(\theta)$ ensures that $p(x|\theta)$ is normalized. i.e.

$$h(\theta) = \int f(x)\exp(\theta^T u(x))dx$$

- Explicit knowledge of $h(\theta)$ is very useful

$$\frac{\partial h(\theta)}{\partial \theta_j} = \frac{\partial}{\partial \theta_j}\int f(x)\exp(\theta^T u(x))dx = \int f(x)\frac{\partial}{\partial \theta_j}\exp(\theta^T u(x))dx =$$
$$\int f(x)u_j(x)\exp(\theta^T u(x))dx = h(\theta)\int \frac{1}{h(\theta)}f(x)u_j(x)\exp(\theta^T u(x))dx = h(\theta)\mathbb{E}u_j(x)$$

- Equivalently

$$\frac{\partial \log h(\theta)}{\partial \theta_j} = \mathbb{E}u_j(x)$$

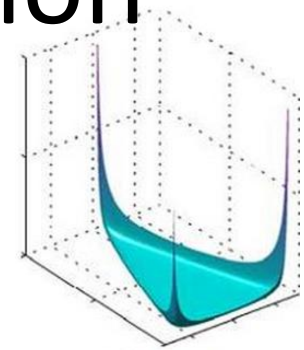
- Similarly $\frac{\partial^2 \log h(\theta)}{\partial \theta_j^2} = \mathbb{D}u_j(x)$

Example: Dirichlet distribution

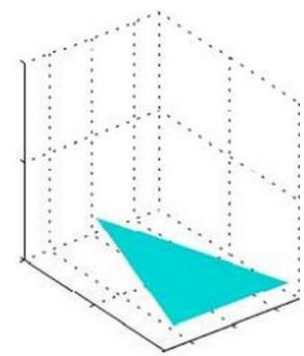
- Distribution over simplex $\{x \in \mathbb{R}^d \mid x_j \geq 0, \sum_{j=1}^d = 1\}$
- Good for setting priors on discrete probabilities
- Density function

$$p(x|\theta) = \frac{\Gamma\left(\sum_{j=1}^d \theta_j\right)}{\prod_{j=1}^d \Gamma(\theta_j)} \prod_{j=1}^d x_j^{\theta_j - 1}$$

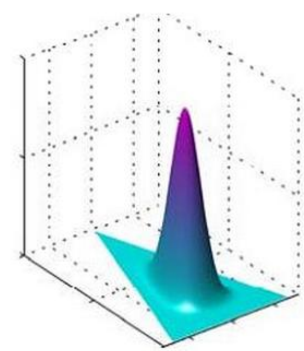
- Sufficient statistics $u_j(x) = \log x_j$
- We may compute $\mathbb{E} \log x_j$ by differentiating normalization constant



$\{\alpha_k\} = 0.1$



$\{\alpha_k\} = 1$



$\{\alpha_k\} = 10$

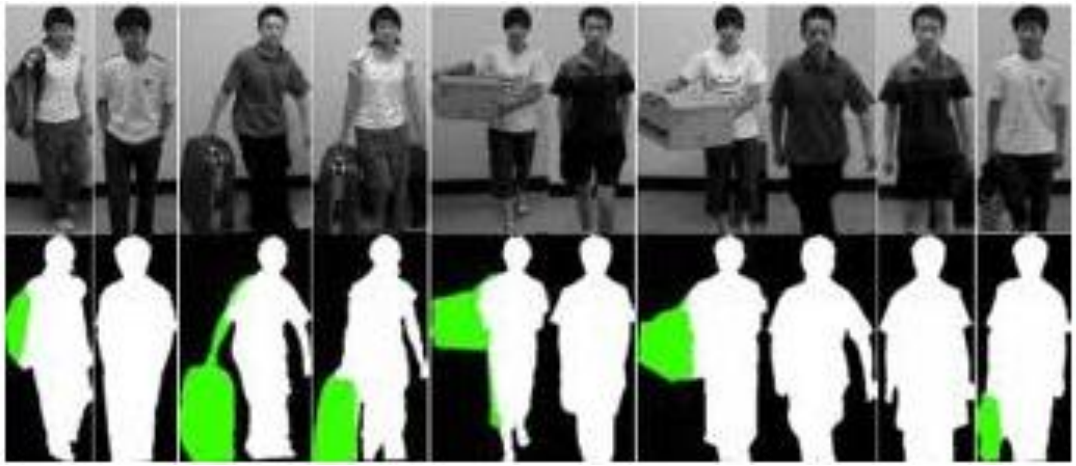
Interdependencies between objects

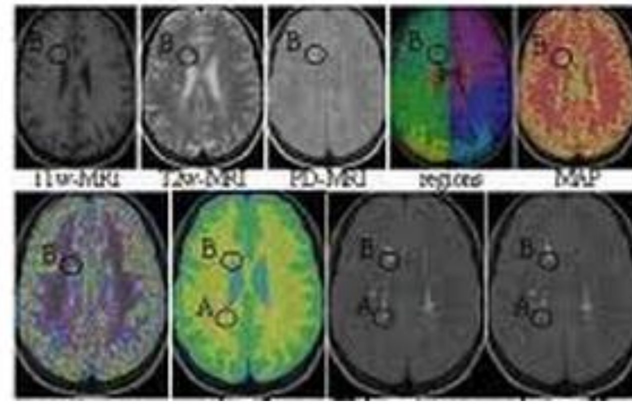
- Up to now we assumed that hidden variables for each object depend only on the observed variables of that object

$$p(T, \theta | X) = \prod_{i=1}^{n} p(t_i | x_i, \theta) p(\theta)$$

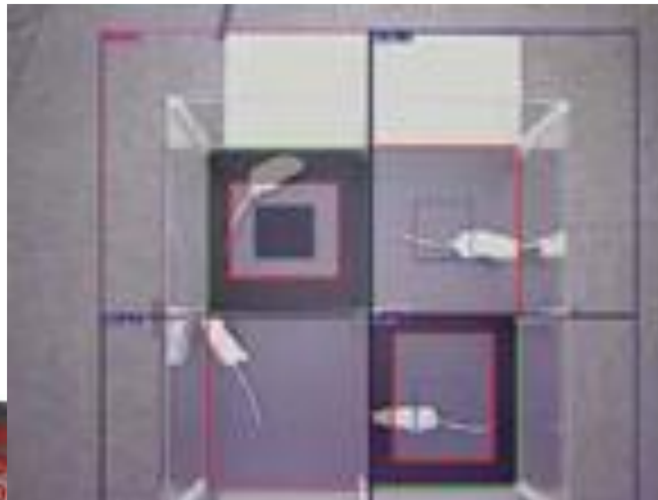
- But what happens if objects are interdependent?
- We need to model the joint distribution $p(T, \theta | X)$ directly even for very large n
- Excellent way for smarter regularization
- This is often the case in practice

Image segmentation

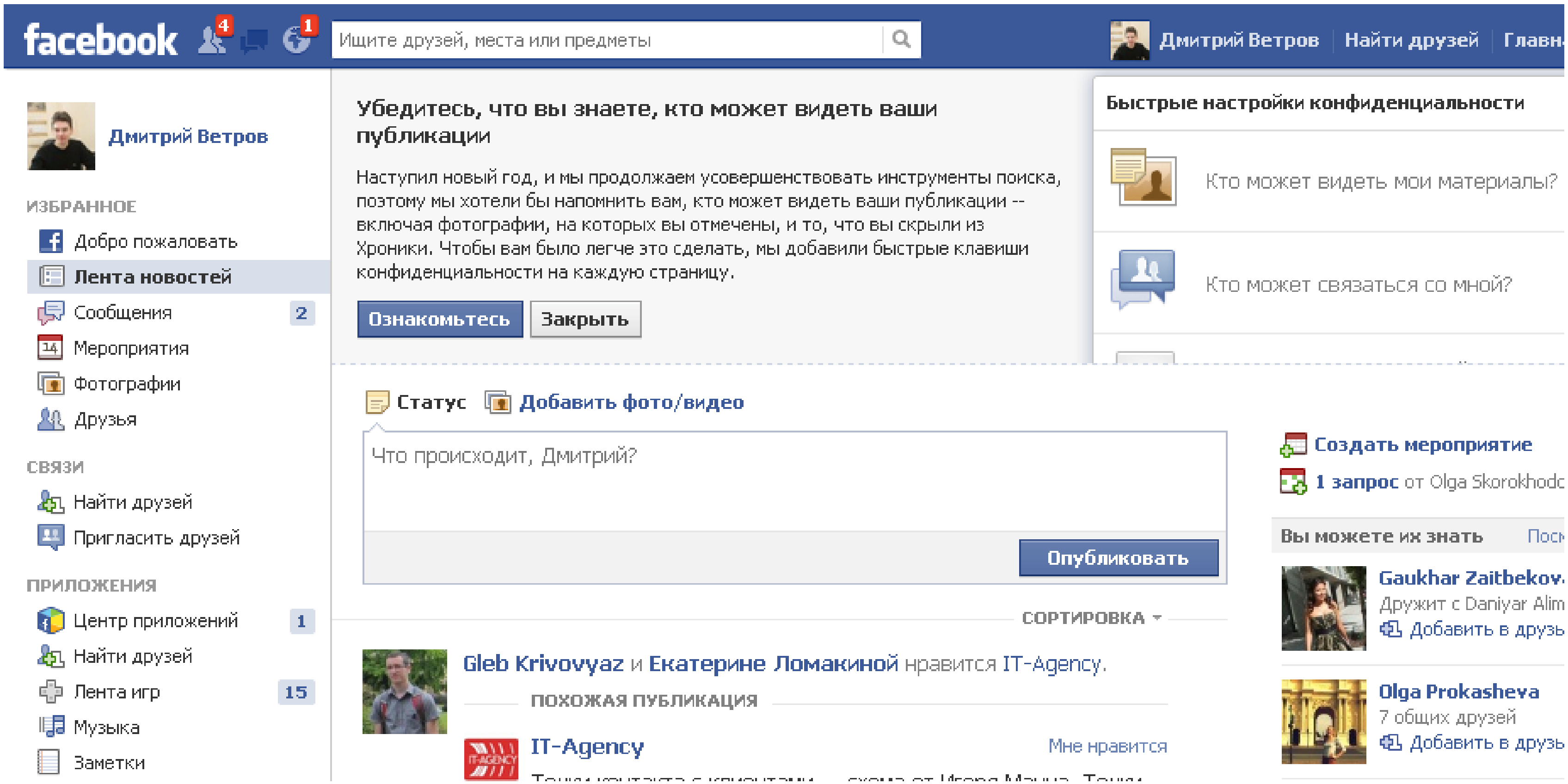




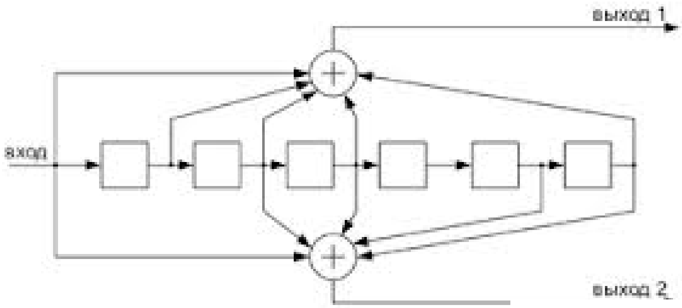
Multiple videotracking

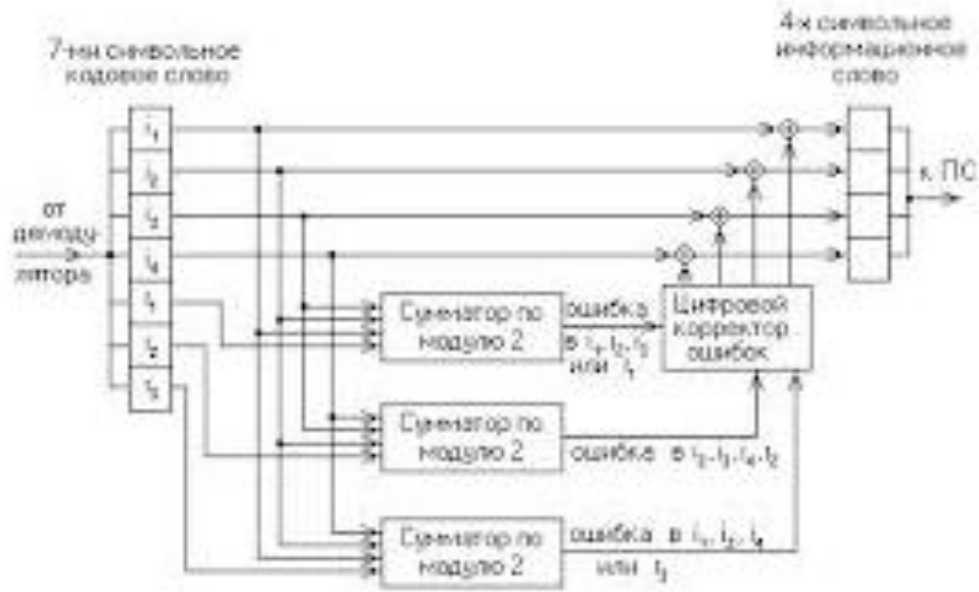


Social network analysis

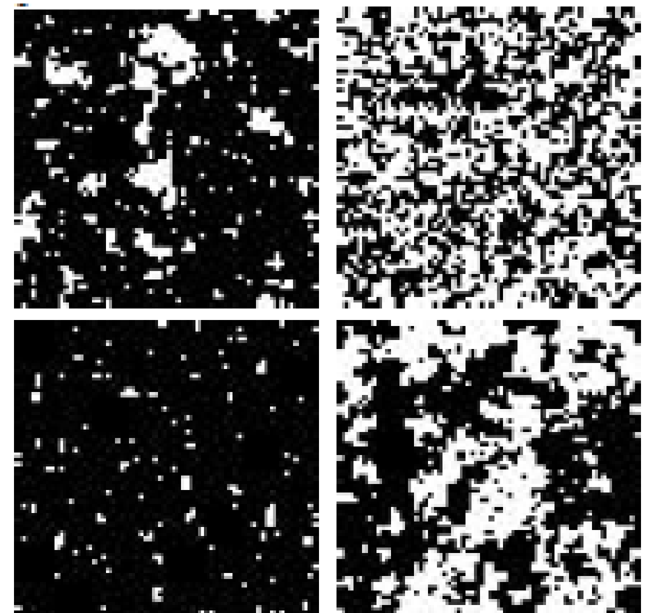


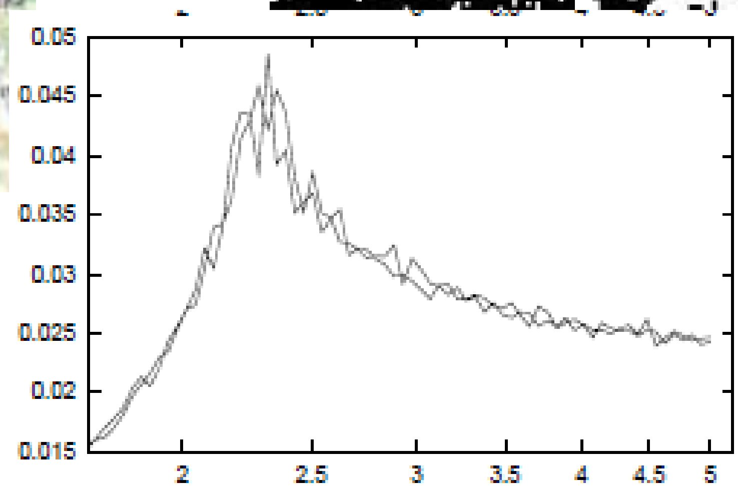
Decoding of noisy messages



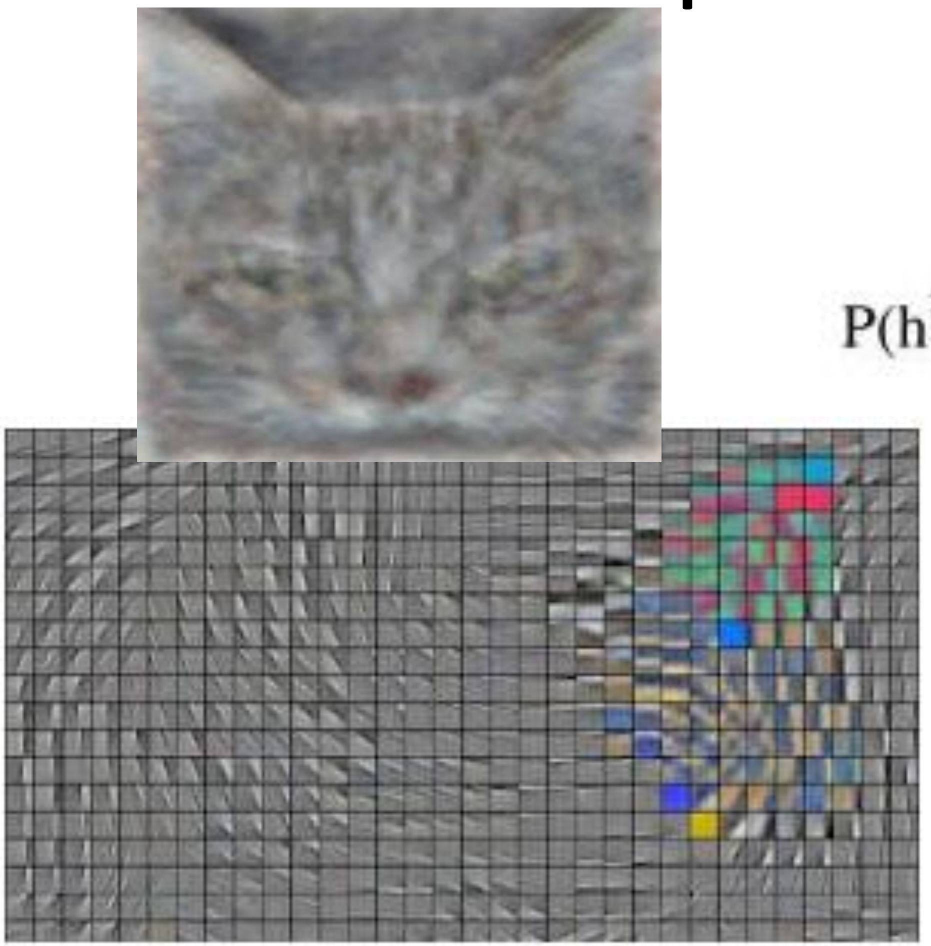


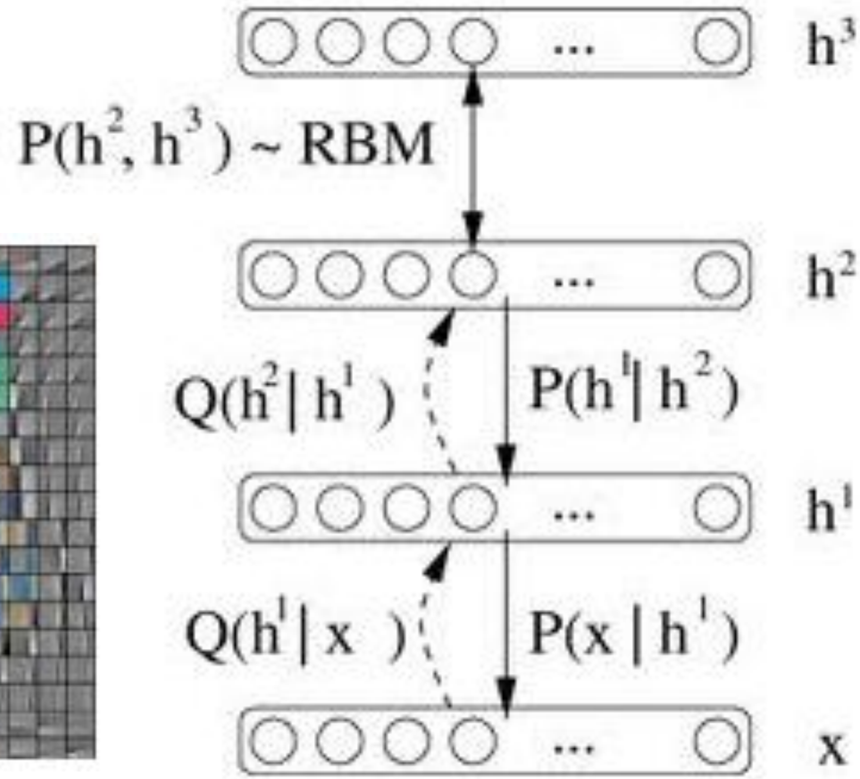
Multi-agent modeling



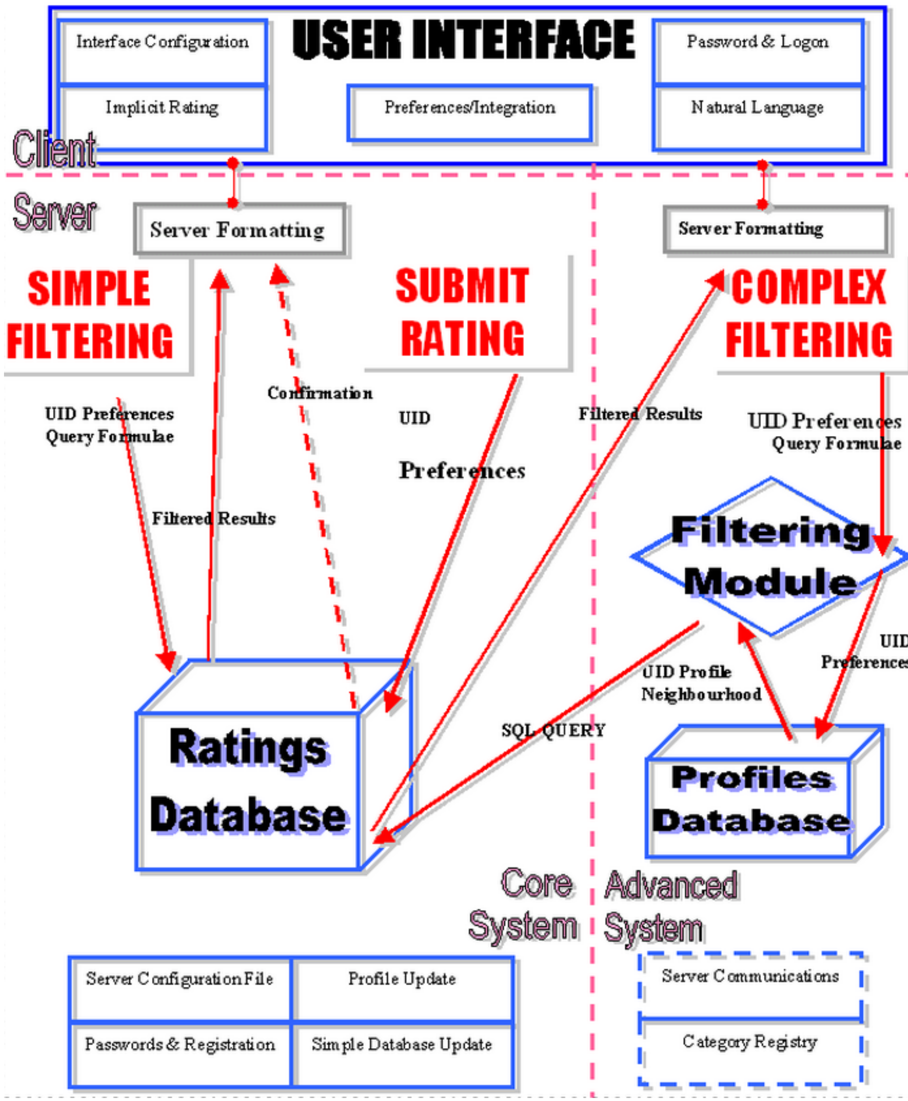


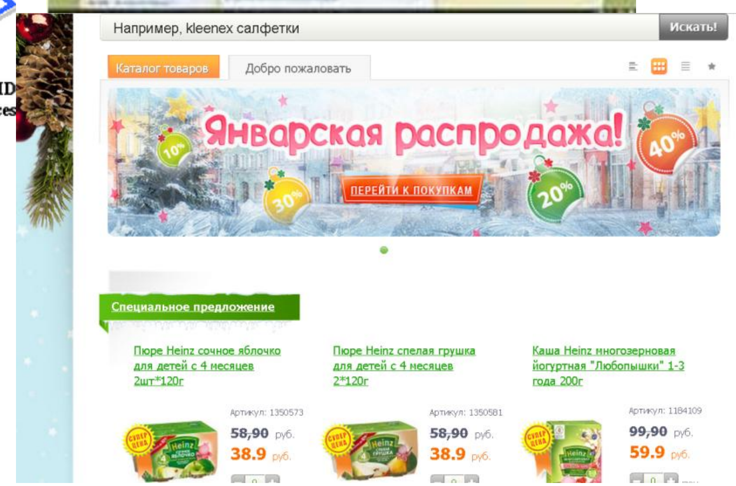
Deep learning





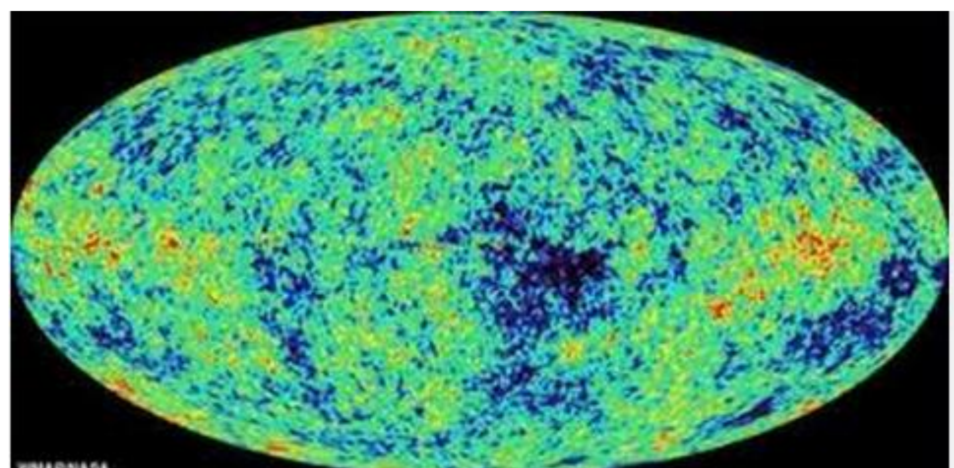
Collaborative filtering





Discrete joint distributions

- Modern probabilistic models deal with the joint distributions of thousands/millions of discrete and continuous variables
- Assume we'd like to model the distribution of 30×30 binary image
- We'd need to set $2^{900} \approx 10^{270}$ probabilities - one for each possible configuration
- The number of atoms in the universe is just about 10^{90}!



Graphical models

- One way to work with such distributions is to make use of conditional independence properties (if any) and split it to factors

$$p(T) = \frac{1}{Z}\prod_{c\in\mathcal{C}}\psi_c(T_c) = \frac{1}{Z}\exp\left(\sum_{c\in\mathcal{C}}\phi_c(T_c)\right),$$

  where T_c are small **intersecting** subsets of T

- This is known as Markov random field (MRF) that is a particular case of graphical model

- The most important problems are to find

$$Z = \sum_T \prod_{c\in\mathcal{C}}\psi_c(T_c)$$

  and

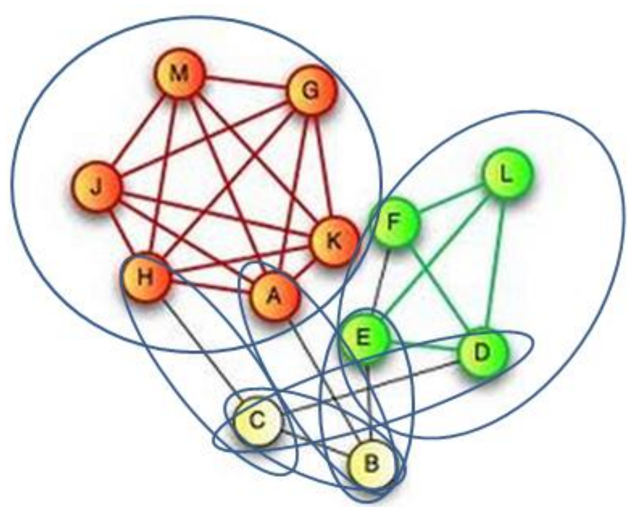
$$T_{MP} = \arg\max_T p(T)$$

  both are NP-hard problems

Graphs in graphical models

- Markov random field can be set by graph whose maximal cliques define factorization of the joint distribution
- Variables correspond to nodes, edges correspond to **direct dependencies**
- If there is no edge between t_i and t_j then these variables are **conditionally independent** given all other variables

$$p(t_i, t_j | T_{\setminus i,j}) = p(t_i | T_{\setminus i,j}) p(t_j | T_{\setminus i,j})$$



Example: image denoising

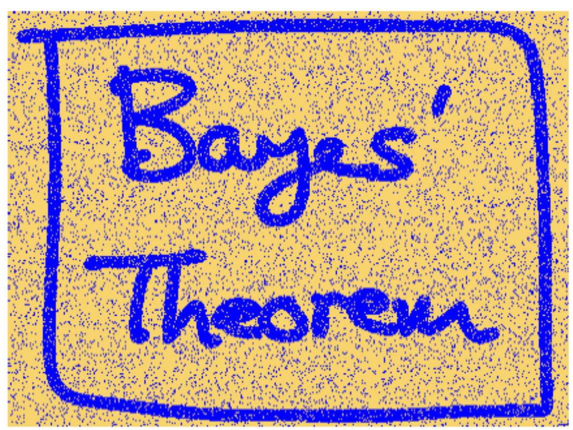
- Consider noised binary image X, $x_i \in \{-1, 1\}$ and its denoised version T, $t_i \in \{-1, 1\}$
- Define the energy of MRF as follows:

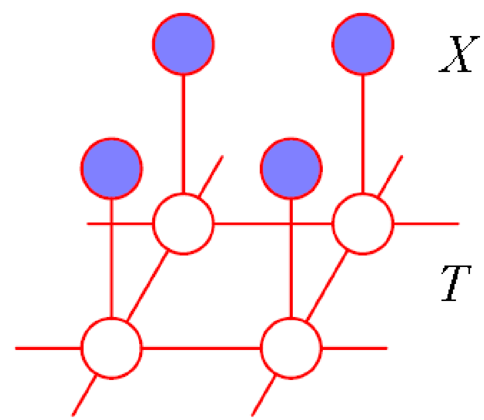
$$-\log p(X, T) = E(X, T) + Const = -\sum_{i \in \mathcal{V}} \theta_1 x_i t_i - \sum_{(i,j) \in \mathcal{E}} \theta_2 t_i t_j + Const,$$

  where $\theta_1, \theta_2 > 0$

- MAP estimate of T given X is

$$T^* = \arg\max_T p(T|X) = \arg\min_T E(X, T)$$





Tensor perspective

- Each discrete distribution $p(T)$, $t_j \in \{1, \dots, K\}$ can be treated as n-dimensional tensor A of length K

$$p(T = \tau) = p(t_1 = \tau_1, \dots, t_n = \tau_n) = A[\tau_1, \dots, \tau_n]$$

- We could use one of tensor decomposition techniques for keeping and processing the distributions

- Tensor train (TT) format (Oseledets11) provides a new framework for working with probabilistic models

$$A[\tau_1, \dots, \tau_n] = G_1[\tau_1] \cdot \ldots \cdot G_n[\tau_n],$$

where $G_j[\tau_j] \in \mathbb{R}^{r_{j-1} \times r_j}$

Tomorrow

- The application of TT to energy decomposition
- New algorithm for partition function estimation
- TT decomposition for global potentials
- Let us see how tensor train runs in a Markov random field :)