Bayesian Methods in Machine Learning

Dmitry Vetrov

Research professor at faculty of Compter Sciences HSE

Head of Bayesian methods research group

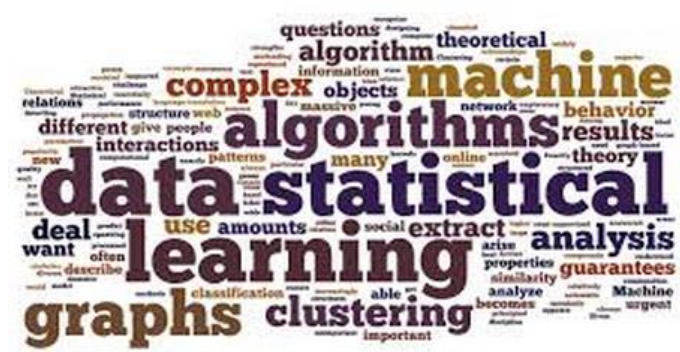
http://bayesgroup.ru

Outline

- Intro to mathematics of big data
- Bayesian framework
- Latent variable models
- Deep Bayes

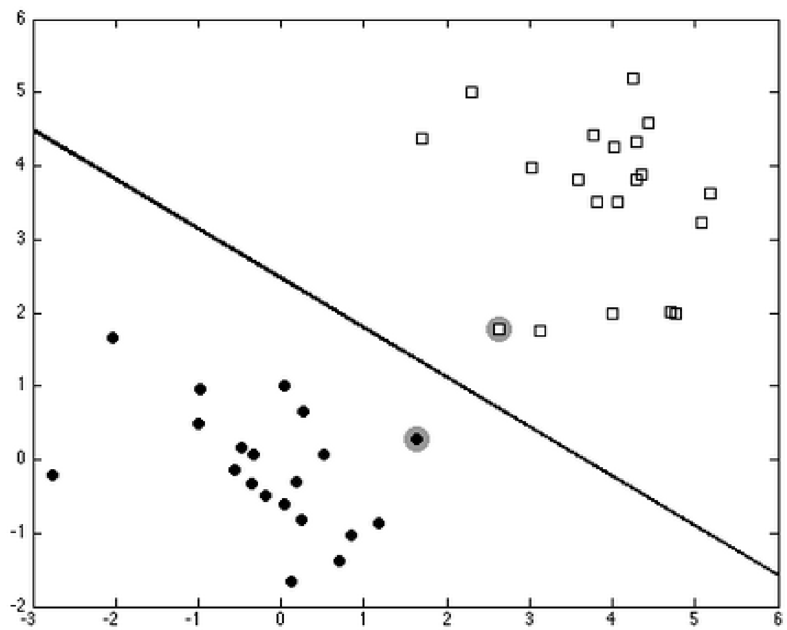
What is machine learning?

- 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 by adjusting the weights W of decision rule.



Simple example

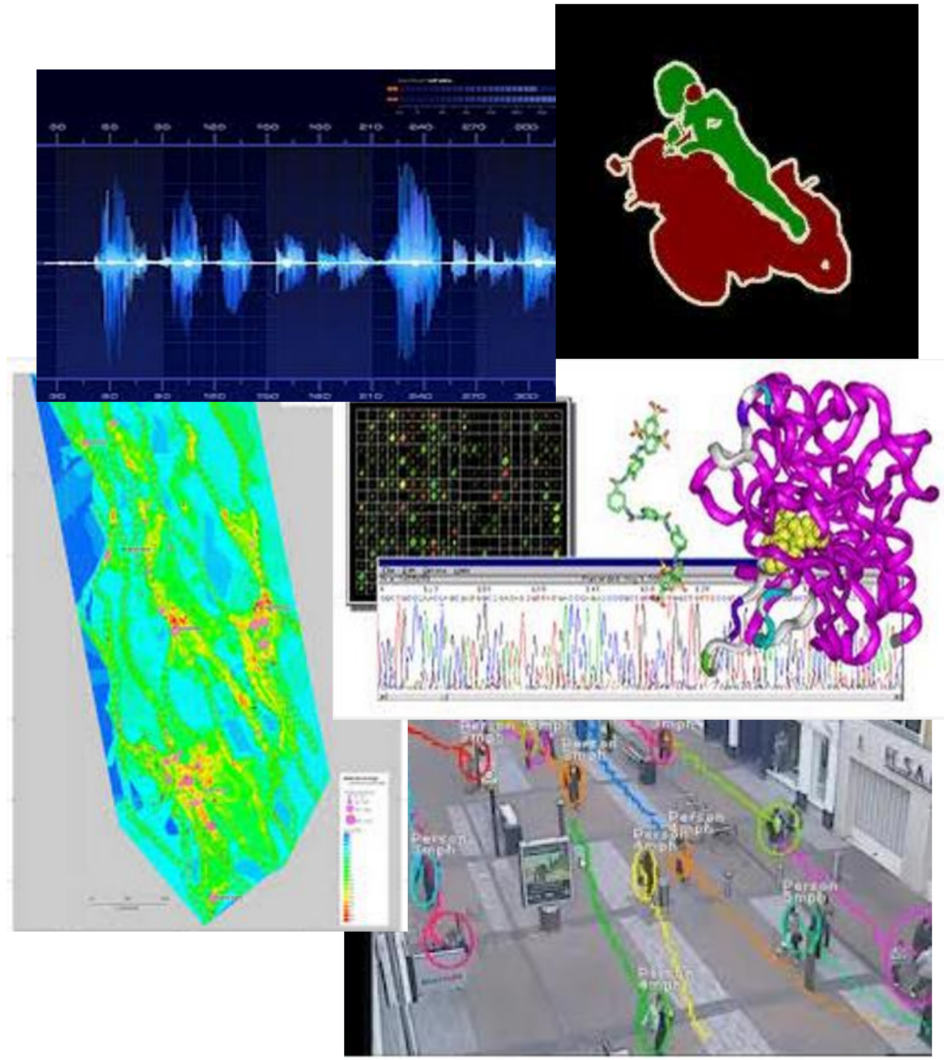
- 2-class Classification problem
- Observed variables are objects' features $X = \{x_i\}_{i=1}^n$, $x_i \in \mathbb{R}^2$
- Hidden variables are binary labels $T = \{t_i\}_{i=1}^n$, $t_i \in \{-1, 1\}$
- Weights W define separating hyperplane: $\hat{t}(x) = \text{sign}(W^T x) + w_0$



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 depostis search
- Bioinformatics
- Web-search
- Sells forecasts
- Behaviour analysis
- Social studies
- etc.



Milestones

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

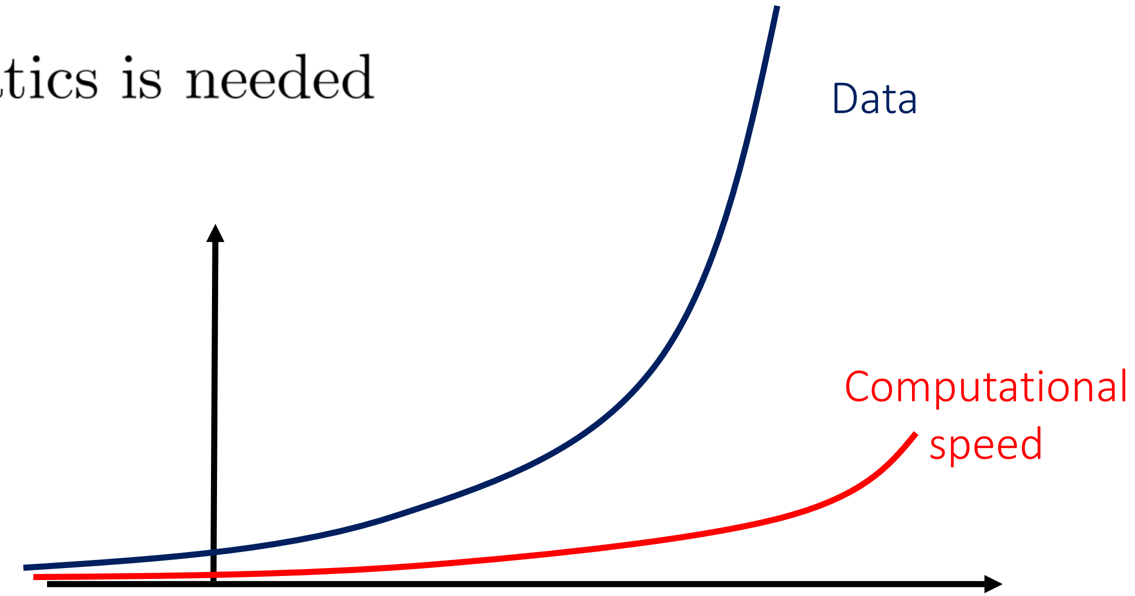
10s. Big Data. . . .

20s. Artificial intelligence?..

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

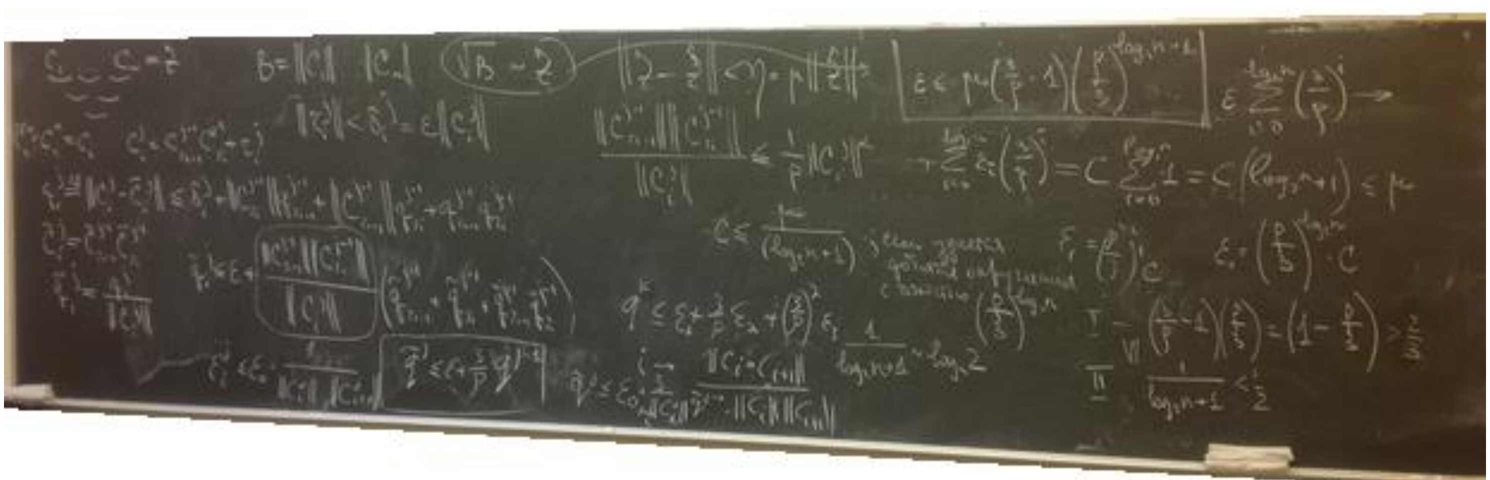
Entering the Age of Big Data

- The amount of data available for analysis grows several orders faster than the computational resources
- Difficult even to keep it not saying about processing
- Old methods simply do not work
- New mathematics is needed



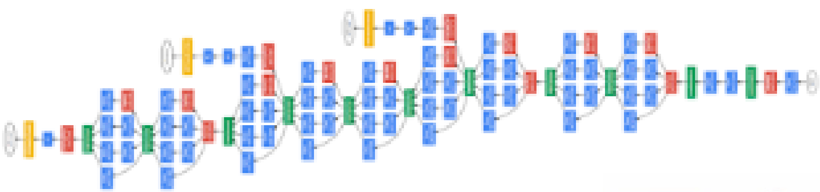
First Steps towards Mathematics of Big Data

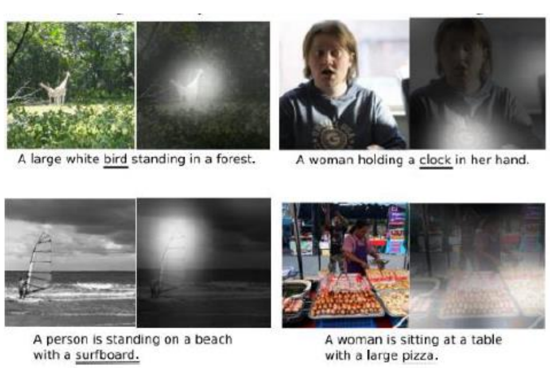
- Bayesian Inference & Graphical Models (Koller09)
- Latent Variable Modeling (Bishop06)
- Deep Learning (Bengio14)
- Tensor Calculus & Decomposition Techniques (No good book published yet)
- Stochastic Optimization (No good book published yet)

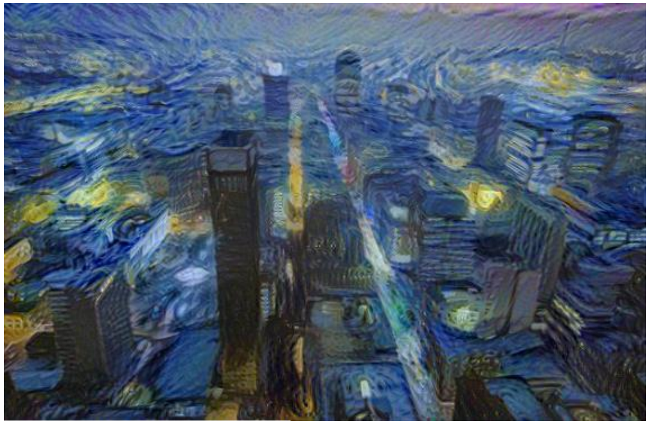


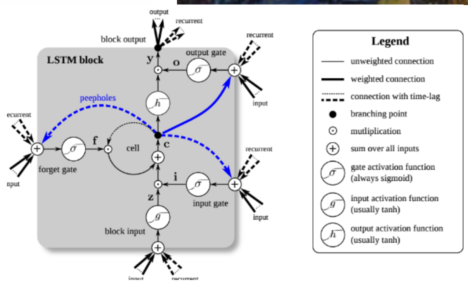
Deep learning: why now?

- Processing huge datasets makes training procedure robust
- Outperforms all existing approaches when dealing with big data
- Multiple yet equivalent local extrema
- Efficient GPU implementations allow to construct very deep networks









Conditional and marginal distributions

Just to remind...

- Conditional distribution

$$\texttt{Conditional} = \frac{\texttt{Joint}}{\texttt{Marginal}}, \quad p(x|y) = \frac{p(x,y)}{p(y)}$$

- Product rule: Any joint distribution can be expressed as a product of one-dimensional conditional distributions

$$p(x,y,z) = p(x|y,z)p(y|z)p(z) = p(z|x,y)p(x|y)p(y)$$

- Sum rule: Any marginal distribution can be obtained from the joint distribution by **intergrating out** unnessesary variables

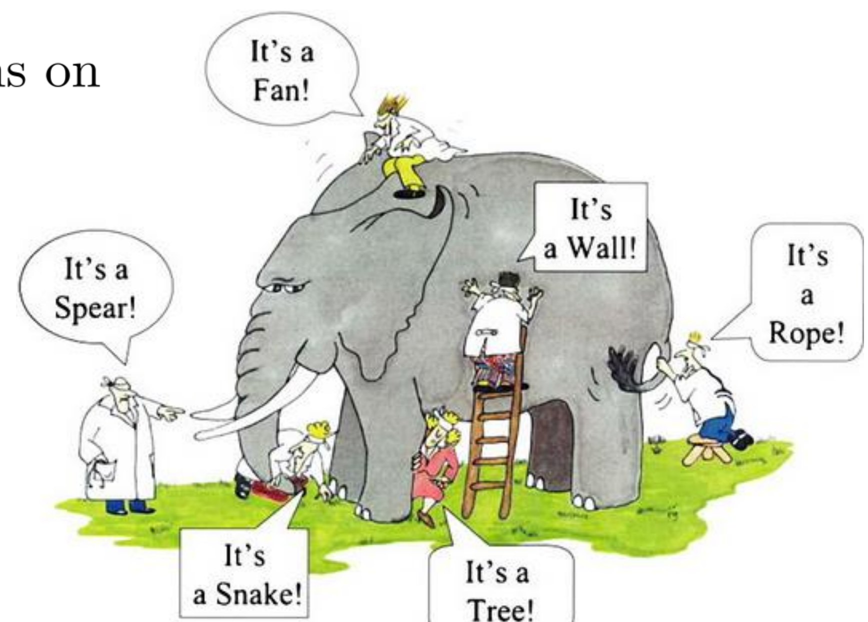
$$p(y) = \int p(x,y)dx = \int p(y|x)p(x)dx = \mathbb{E}_x p(y|x)$$

Bayesian framework

- Encodes ignorance in terms of distributions
- Makes use of **Bayes Theorem**

$$\texttt{Posterior} = \frac{\texttt{Likelihood} \times \texttt{Prior}}{\texttt{Evidence}}, \quad p(\theta|X) = \frac{p(X|\theta)p(\theta)}{\int p(X|\theta)p(\theta)d\theta}$$

- Posteriors may serve as new priors, i.e. may combine multiple models!
- **BigData:** we can process data streams on an update-and-forget basis
- Support distributed processing

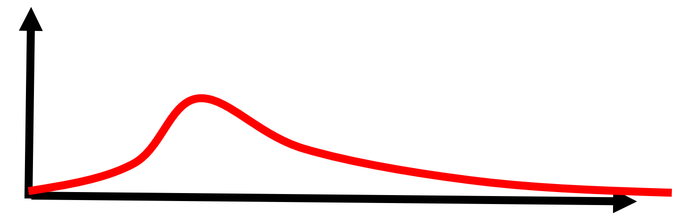


Frequentist vs. Bayesian frameworks

	Frequentist	Bayesian
Randomness	Objective indefiniteness	Subjective ignorance
Variables	Random and Deterministic	Everything is random
Inference	Maximal likelihood	Bayes theorem
Estimates	ML-estimates	Posterior or MAP-estimates
Applicability	$n \gg 1$	$\forall n$

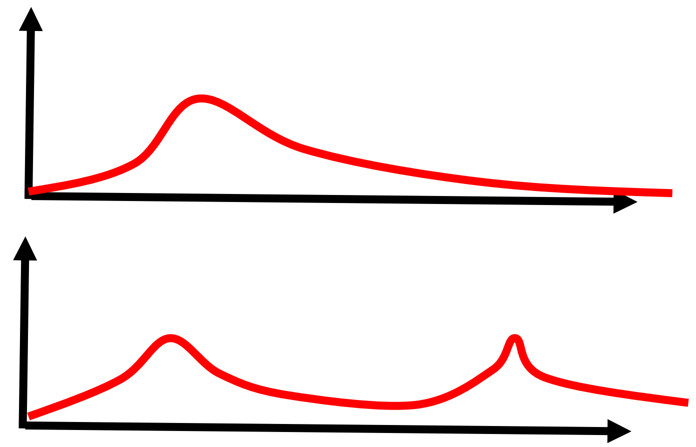
Bayesian inference

- Consider blind wisdomers who try to understand the mass of an elephant using their tactile measurements.
- They start with common knowledge about animals typical masses $p(\theta)$



Bayesian inference

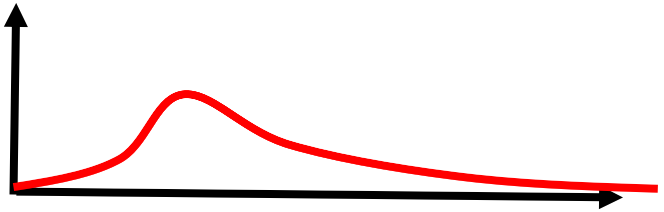
- Consider blind wisdomers who try to understand the mass of an elephant using their tactile measurements.
- They start with common knowledge about animals typical masses $p(\theta)$
- The first wisdomer touches a tail



$$p(\theta|x_1) = \frac{p_1(x_1|\theta)p(\theta)}{\int p_1(x_1|\theta)p(\theta)d\theta}$$

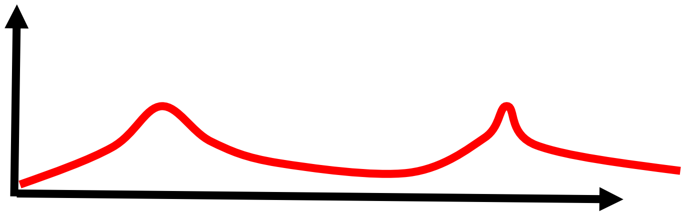
Bayesian inference

- Consider blind wisdomers who try to understand the mass of an elephant using their tactile measurements.



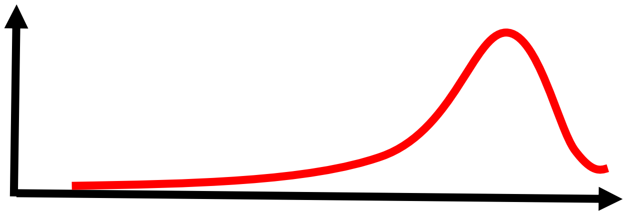
- They start with common knowledge about animals typical masses $p(\theta)$

- The first wisdomer touches a tail



$$p(\theta|x_1) = \frac{p_1(x_1|\theta)p(\theta)}{\int p_1(x_1|\theta)p(\theta)d\theta}$$

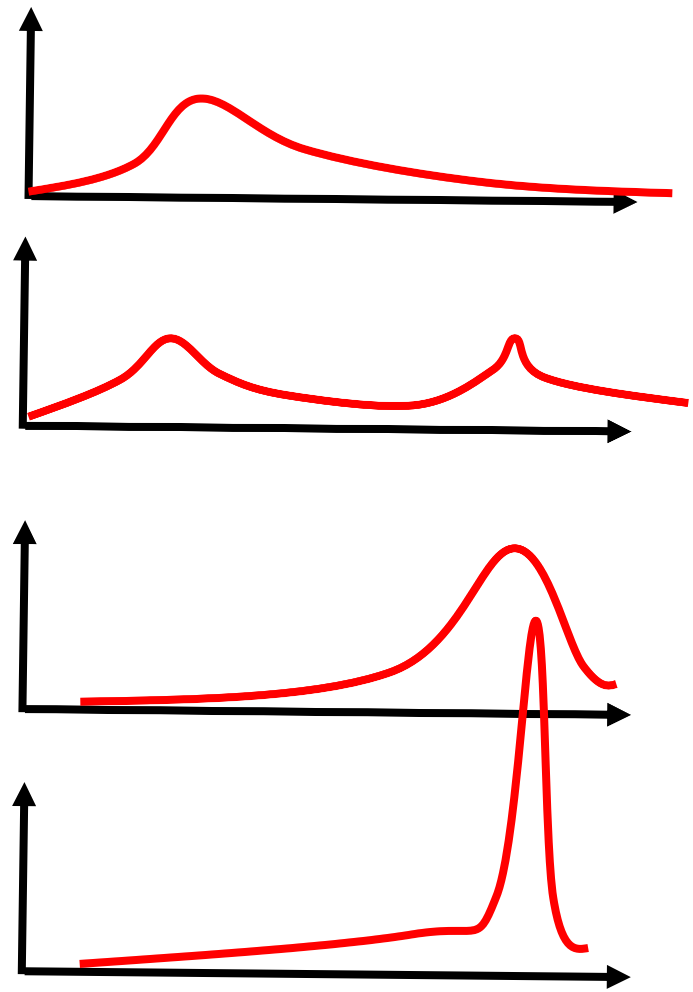
- The second wisdomer touches a leg and uses $p(\theta|x_1)$ as **his new prior**



$$p(\theta|x_1, x_2) = \frac{p_2(x_2|\theta)p(\theta|x_1)}{\int p_2(x_2|\theta)p(\theta|x_1)d\theta}$$

Bayesian inference

- Consider blind wisdomers who try to understand the mass of an elephant using their tactile measurements.



- They start with common knowledge about animals typical masses $p(\theta)$
- The first wisdomer touches a tail

$$p(\theta|x_1) = \frac{p_1(x_1|\theta)p(\theta)}{\int p_1(x_1|\theta)p(\theta)d\theta}$$

- The second wisdomer touches a leg and uses $p(\theta|x_1)$ as **his new prior**

$$p(\theta|x_1, x_2) = \frac{p_2(x_2|\theta)p(\theta|x_1)}{\int p_2(x_2|\theta)p(\theta|x_1)d\theta}$$

- ...
- At the end they form sharp distribution $p(\theta|x_1, \dots, x_m)$

Bayesian Learning and Inference

- Establishes joint distribution $p(X, T, W)$ on hidden variables T, observed variables X and parameters of decision rule W
- Learning: given labeled **training data** (X_{tr}, T_{tr}) find posterior on W:

$$p(W|X_{tr}, T_{tr}) = \frac{p(T_{tr}, X_{tr}|W)p(W)}{\int p(T_{tr}, X_{tr}|W)p(W)dW}$$

- Prior knowledge about W serves as **regularization** term
- Inference: given observed variables X of **new objects** find the distribution on hidden variables

$$p(T|X, X_{tr}, T_{tr}) = \int p(T|X, W)p(W|X_{tr}, T_{tr})dW$$

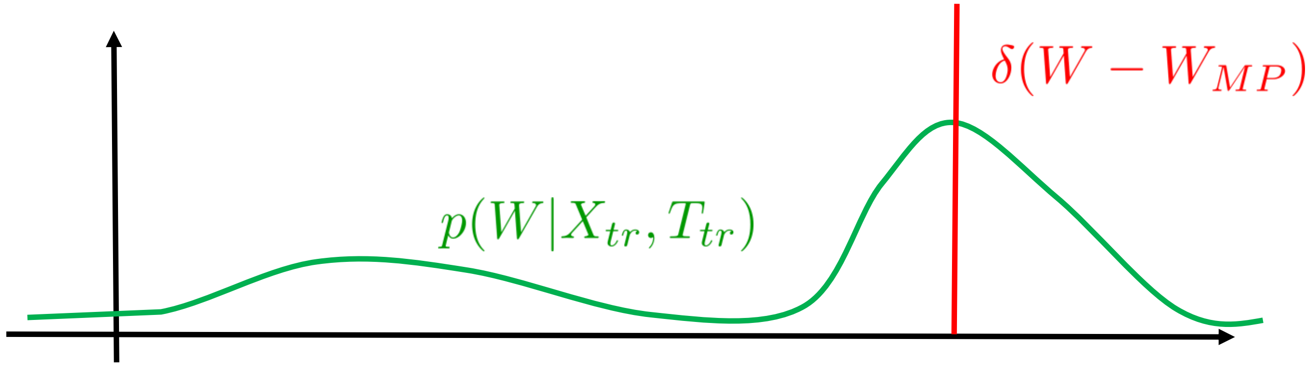
Poor man's Bayes

- Simplified probabilistic modeling
- Approximate posteior $p(W|X_{tr}, T_{tr})$ with a delta function $\delta(W - W_{MP})$
- Corresponds to point estimate of W:

$$W_{MP} = \arg\max p(W|X_{tr}, T_{tr}) = \arg\max p(T_{tr}, X_{tr}|W)p(W)$$

- Inference is more simple

$$p(T|X, X_{tr}, T_{tr}) = \int p(T|X, W)p(W|X_{tr}, T_{tr})dW \approx p(T|X, W_{MP})$$



Regularization

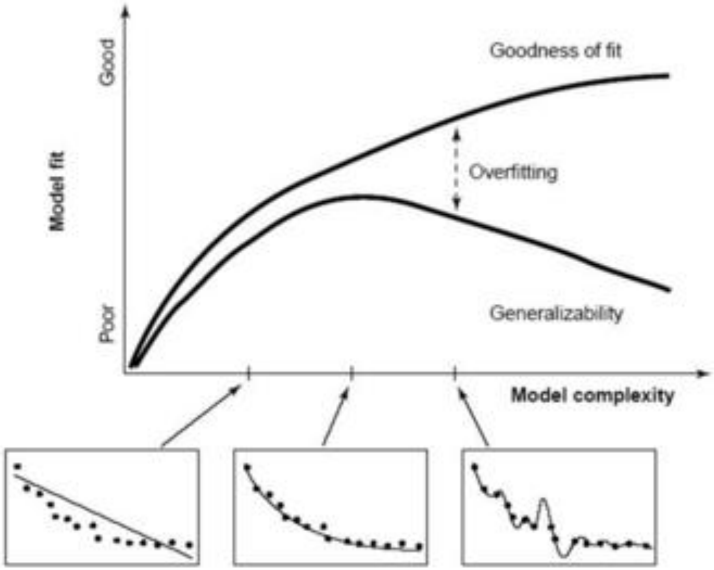
- By establishing priors over the weights θ we may **regularize** maximum likelihood estimates

Prior term

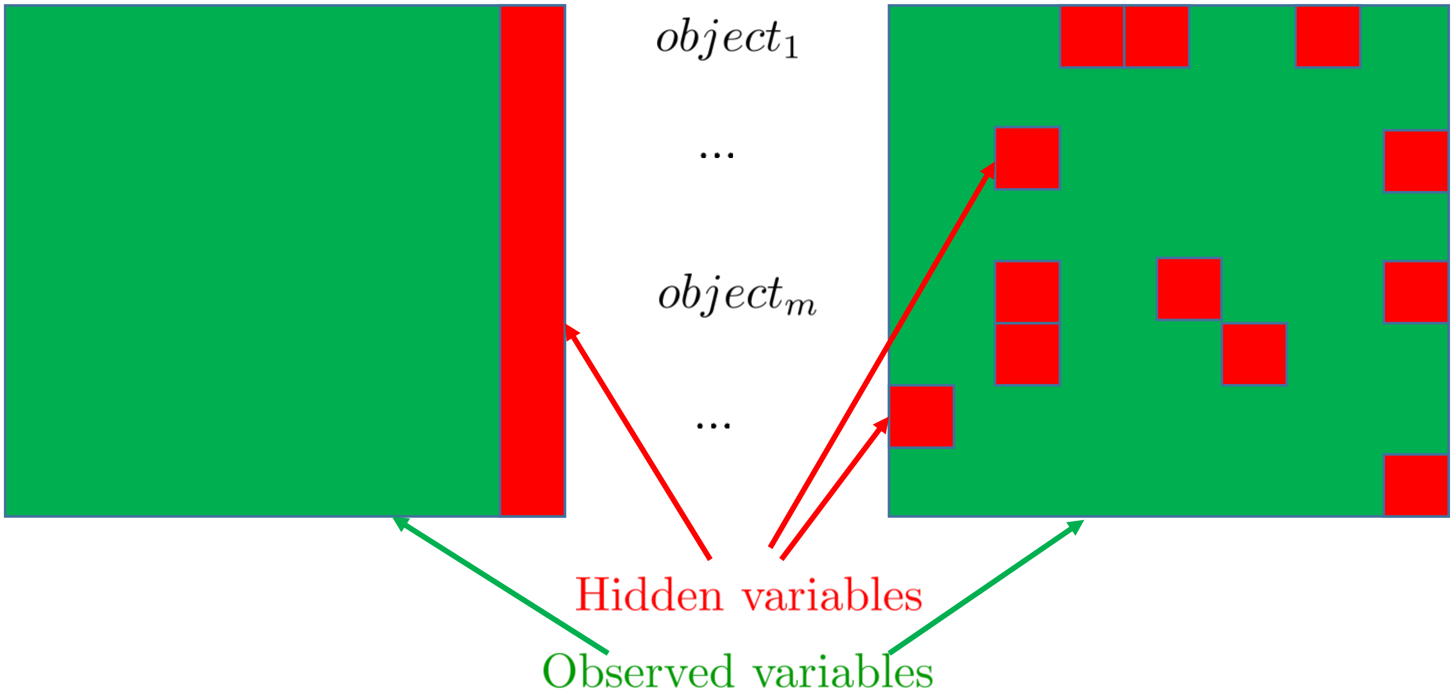
$$\cancel{p(Data|\theta) \to \max_{\theta}} \qquad p(\theta|Data) = \frac{p(Data|\theta)\boxed{p(\theta)}}{\int p(Data|\theta)p(\theta)d\theta}$$

- Prevents overfitting

- We can set the best prior automatically



Learning from incomplete data



We can learn from incomplete, weakly-labeled and unlabeled data in a correct way using EM-framework and its numerous extensions

Advantages of Bayesian framework

- Regularization
  - Incorporates specifics of particular problem
- Extendibility
  - Builds complex model from simpler ones
- Latent variable modeling
  - Learns from incomplete data
- Ensembling
  - Performs weighted voting across multiple algorithms
- Scalability (new!)
  - Applicable to large datasets when combined with deep neural networks

Exponential class of distributions

- Distribution $p(y|\theta)$ belongs to exponential class if it can be expressed as follows
$$p(y|\theta) = \frac{f(y)}{g(\theta)} \exp\left(\theta^T u(y)\right),$$
where $f(y) \geq 0$, $g(\theta) > 0$

- Function $g(\theta)$ ensures that right-hand expression is a distribution $g(\theta) = \int f(y) \exp\left(\theta^T u(y)\right) dy$

- Functions $u(y)$ are **sufficient statistics** whose values contain all information that can be extracted from sample about distribution

- Function $f(y)$ can be **arbitrary** non-negative function

Log-concavity of exponential class

- Consider derivate of $\log g(\theta)$

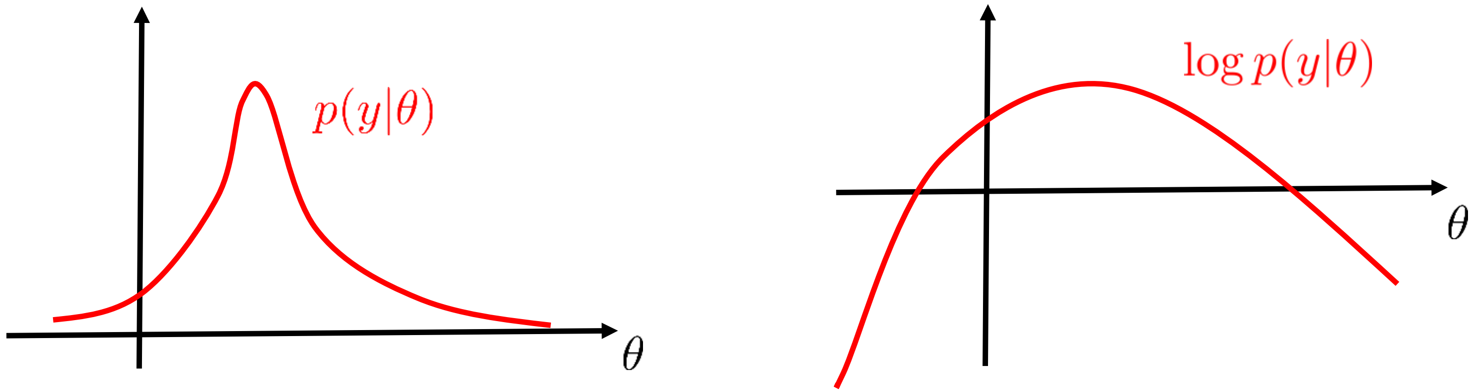
$$\frac{\partial \log g(\theta)}{\partial \theta_j} = \frac{1}{g(\theta)} \frac{\partial g(\theta)}{\partial \theta_j} = \frac{1}{g(\theta)} \frac{\partial}{\partial \theta_j} \int f(y) \exp(\theta^T u(y)) dy =$$
$$\frac{1}{g(\theta)} \int f(y) \exp(\theta^T u(y)) u_j(y) dy = \int p(y|\theta) u_j(y) dy = \mathbb{E}_y u_j(y)$$

- Analogously $\frac{\partial^2 \log g(\theta)}{\partial \theta_i \partial \theta_j} = \mathrm{Cov}(u_i(y), u_j(y))$
- Thus $\log g(\theta)$ is convex function, consequently

$$\log p(y|\theta) = \theta^T u(y) - \log g(\theta) + \log f(y)$$

is concave function of θ

Log-concavity of exponential class



- For log-concave distributions maximum likelihood estimation can be done in an efficient manner
- All discrete distributions and many continuous (Gaussian, Laplace, Gamma, Dirichlet, Wishart, Beta, Chi-squared, etc.) belong to exponential class

Example: Gaussian distribution

- Standard form of 1-dimensional Gaussian

$$p(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

- Natural form

$$p(x|\theta) = \frac{1}{\sqrt{-\frac{\pi}{\theta_1} \exp\left(-\frac{\theta_2^2}{4\theta_1}\right)}} \exp(\theta_1 x^2 + \theta_2 x),$$

where $\theta_1 = -\frac{1}{2\sigma^2}$ and $\theta_2 = \frac{\mu}{\sigma^2}$

- Hence x and x^2 are sufficient statistics and

$$g(\theta) = \sqrt{-\frac{\pi}{\theta_1} \exp\left(-\frac{\theta_2^2}{4\theta_1}\right)}$$

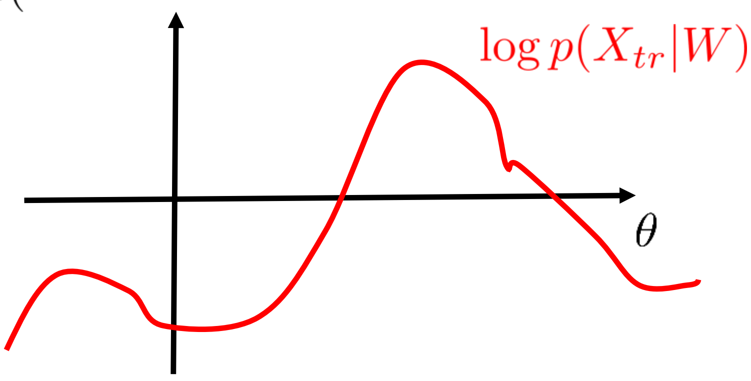
- Note that there is one-to-one correspondence between (θ_1, θ_2) and (μ, σ)

Incomplete likelihood

- Let our likelihood $p(X, T|W)$ belong to exponential class and $p(W)$ is log-concave w.r.t. W
- If we knew X_{tr}, T_{tr} we would find W_{MP} easily
- Suppose that only X_{tr} is known. Then we need to find

$$W_* = \arg\max p(W|X_{tr}) = \arg\max \log p(W|X_{tr}) =$$
$$\arg\max\left(\log p(X_{tr}|W) + \log p(W)\right) = \arg\max\left(\log \int p(X_{tr}, T|W)dT + \log p(W)\right)$$

- The first term is no longer concave :(



Variational lower bound

$$\log p(X_{tr}|W) = \int \log p(X_{tr}|W) q(T) dT = \int \log \frac{p(X_{tr}, T|W)}{p(T|X_{tr}, W)} q(T) dT =$$

$$= \int \log \frac{p(X_{tr}, T|W) q(T)}{p(T|X_{tr}, W) q(T)} q(T) dT = \int \log \frac{p(X_{tr}, T|W)}{q(T)} q(T) dT +$$

$$+ \int \log \frac{q(T)}{p(T|X_{tr}, W)} q(T) dT = \mathcal{L}(q, W) + KL(q(T)||p(T|X_{tr}, W))$$

- $KL(q||p)$ stands for **Kullback-Leibler divergence** that is a pseudo-distance between distributions.
- KL-divergence is always non-negative and equals to zero iff both arguments coinside almost everywhere
- Hence $\mathcal{L}(q, W)$ is **variational lower bound** for the log of incomplete likelihood
- Idea! Let us maximize $\mathcal{L}(q, W)$ iteratively w.r.t. to W and $q(T)$ instead of maximizing $\log p(X_{tr}|W)$

EM-algorithm

- E-step: $\mathcal{L}(q, W_{t-1}) \to \max_q$. Equivalent to KL-divergence minimization. Can be done in an explicit manner

$$q_t(T) = \arg\min_q KL(q(T)||p(T|X_{tr}, W_{t-1})) = p(T|X_{tr}, W_{t-1})$$

- M-step: $\mathcal{L}(q_t, W) \to \max_W$. Note that

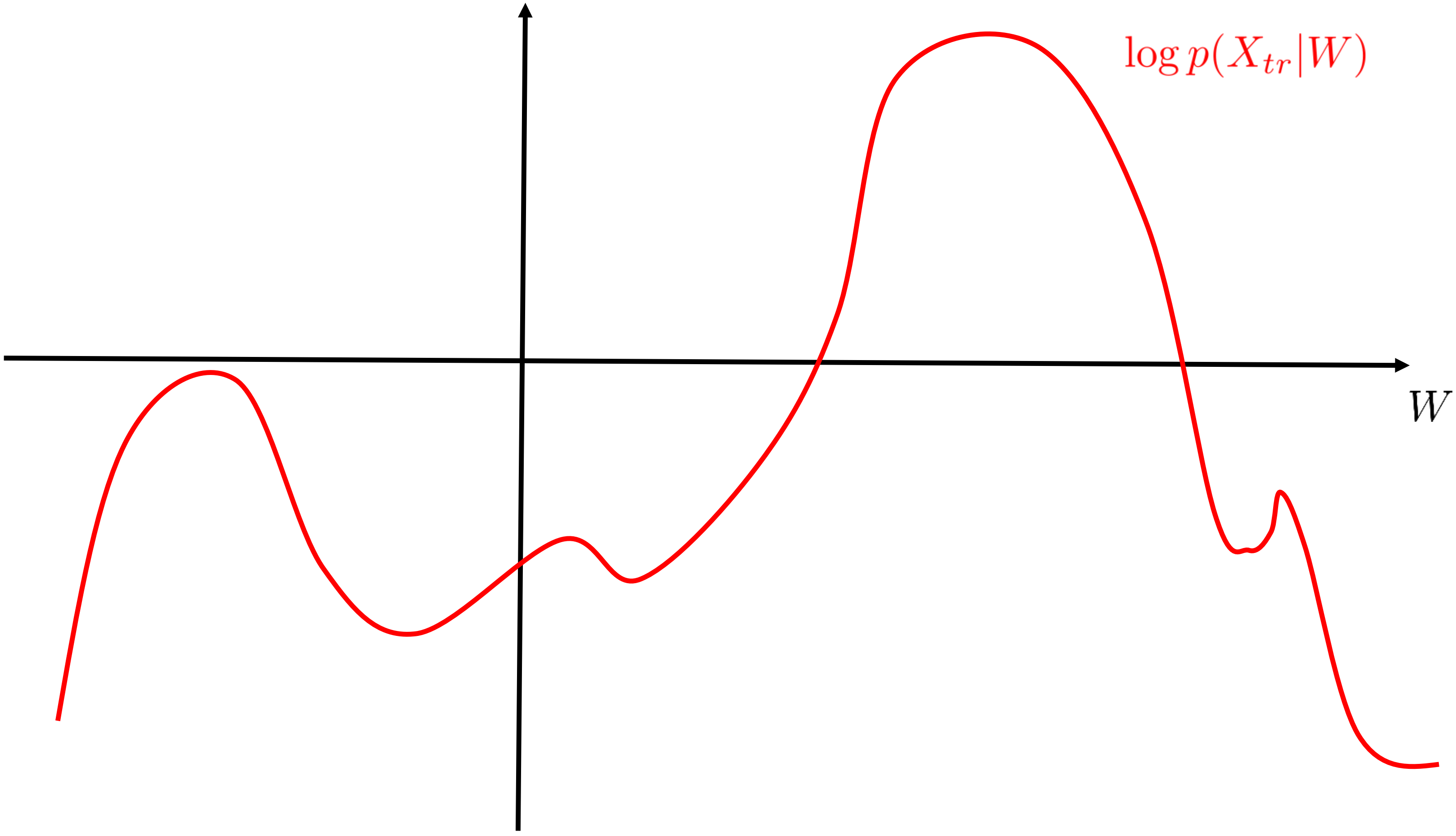
$$W_t = \arg\max_W \mathcal{L}(q_t, W) = \arg\max_W \int q_t(T) \log \frac{p(X_{tr}, T|W)}{q_t(T)} dT =$$
$$\arg\max_W \int q_t(T) \log p(X_{tr}, T|W) dT$$

  corresponds to maximizing convex combination of concave functions, i.e. concave function

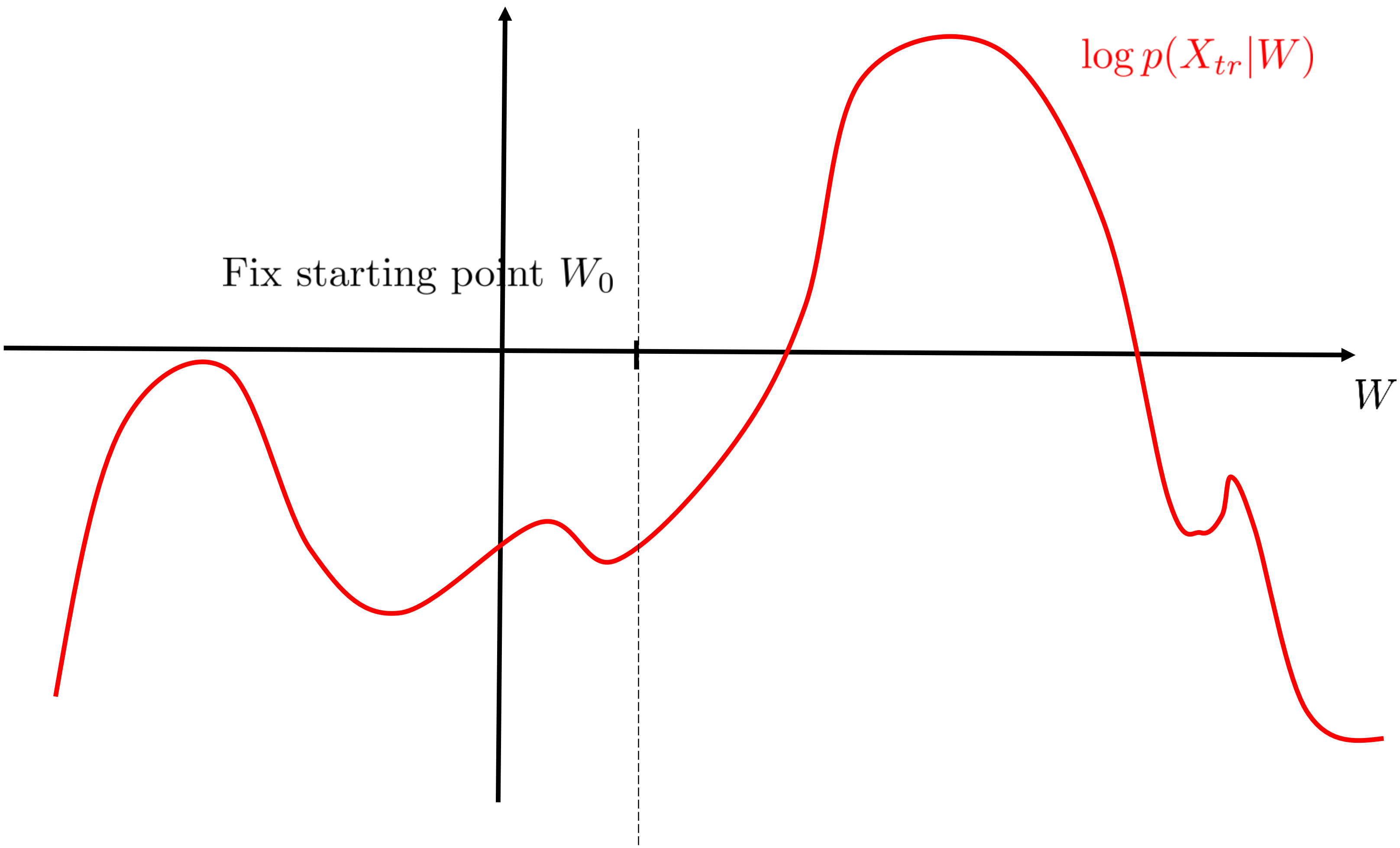
- Iterate until convergence

- $\mathcal{L}(q, W)$ monotonically increases

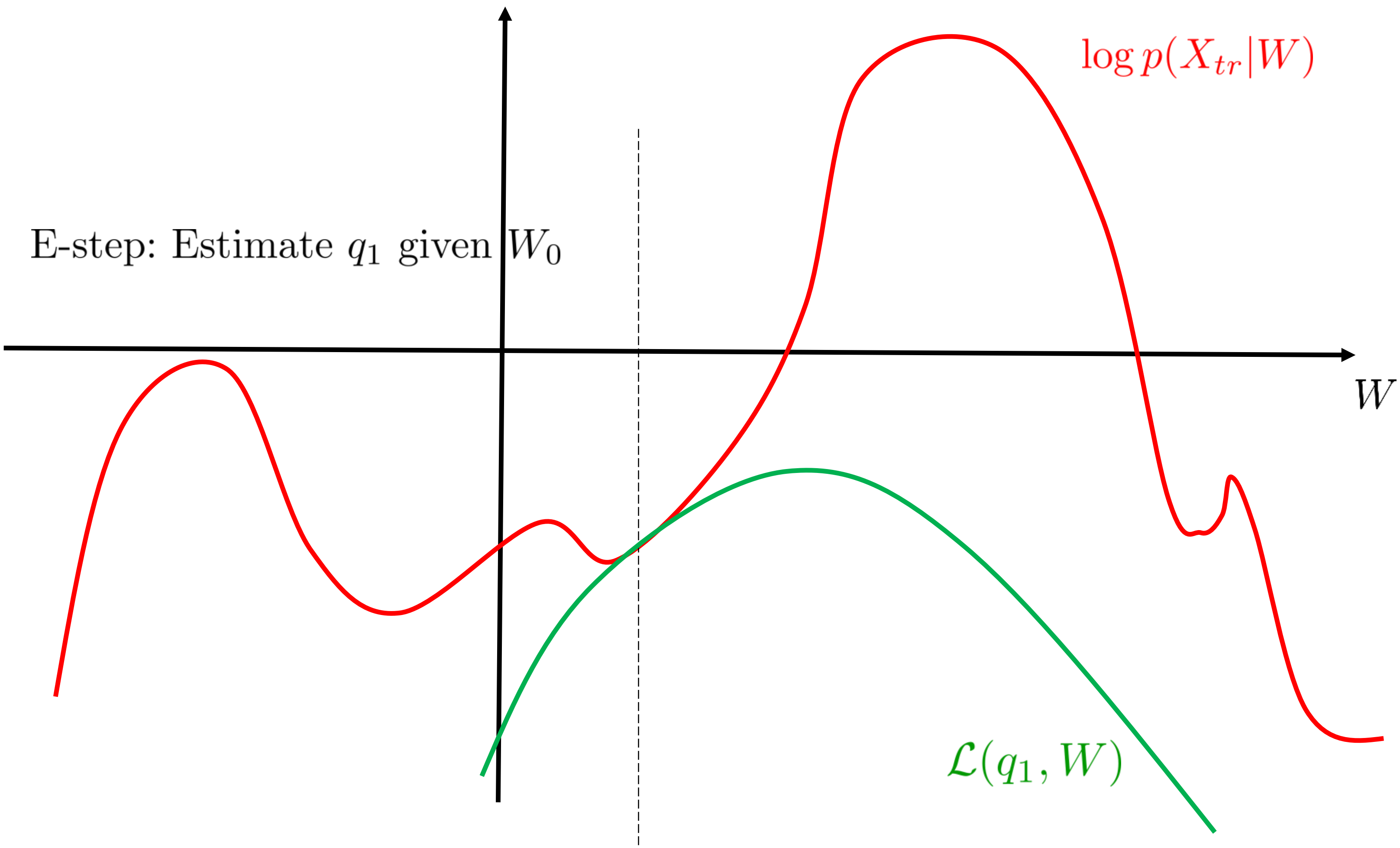
EM-algorithm



EM-algorithm

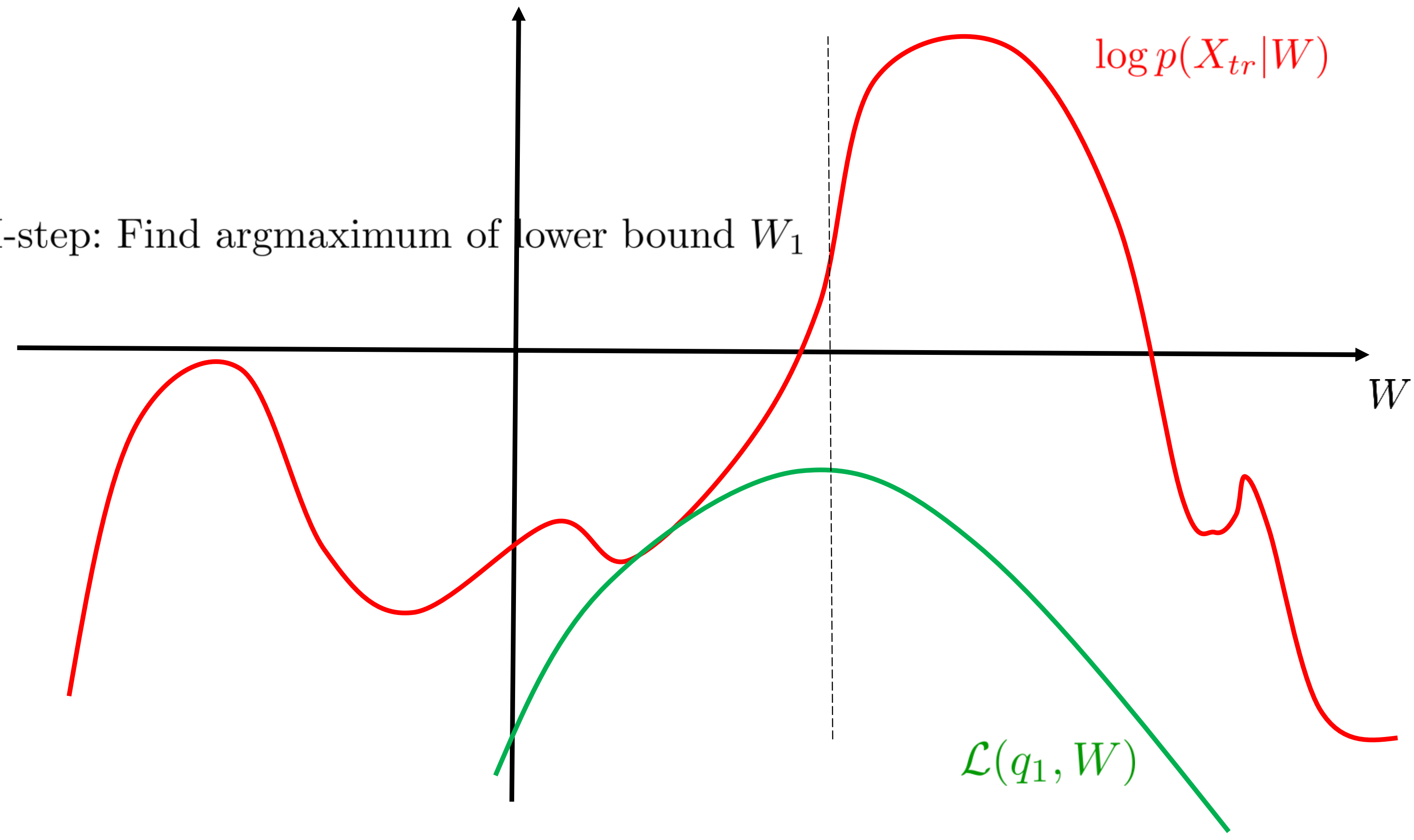


EM-algorithm



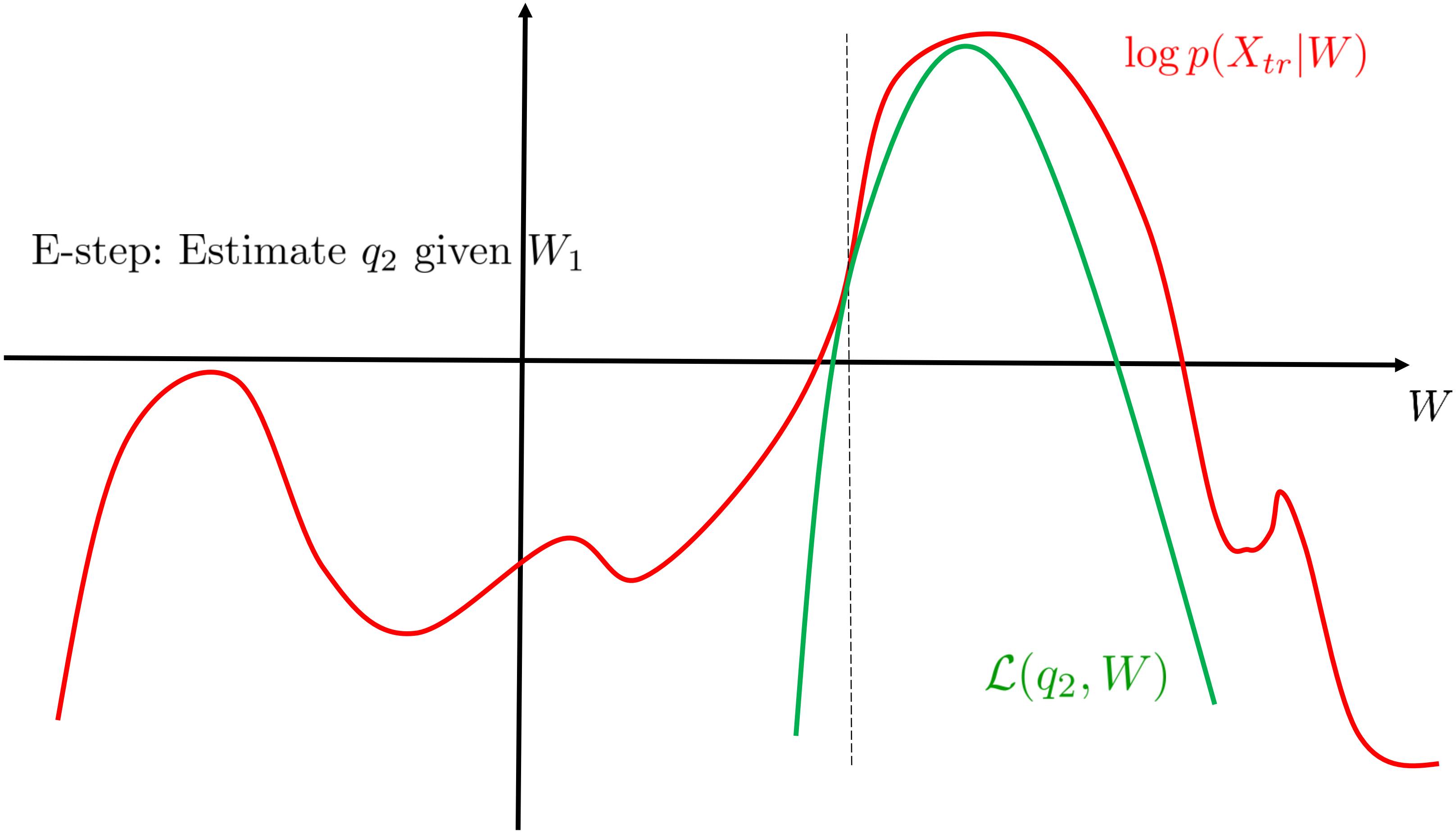
EM-algorithm

M-step: Find argmaximum of lower bound W_1



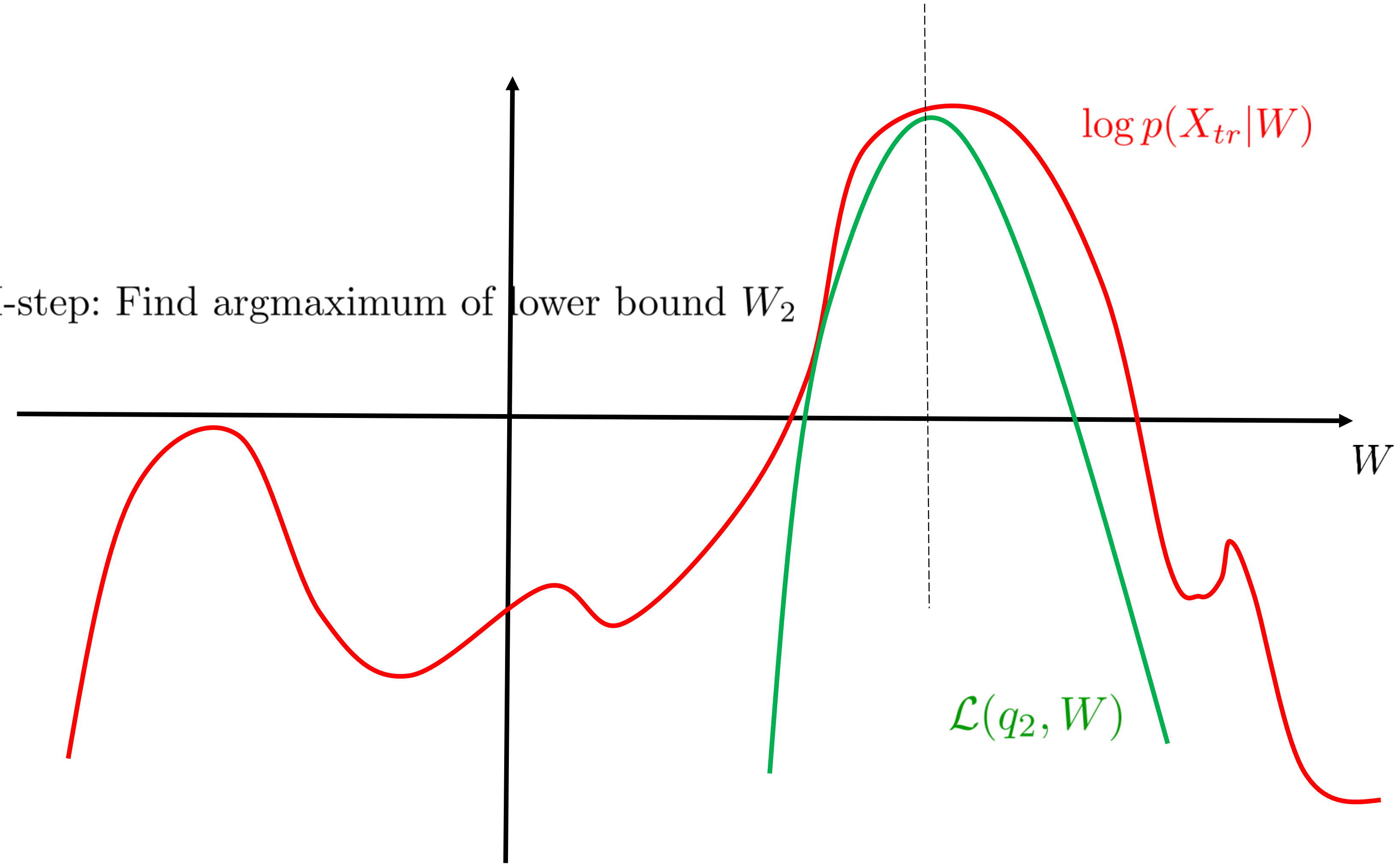
EM-algorithm

E-step: Estimate q_2 given W_1



EM-algorithm

M-step: Find argmaximum of lower bound W_2

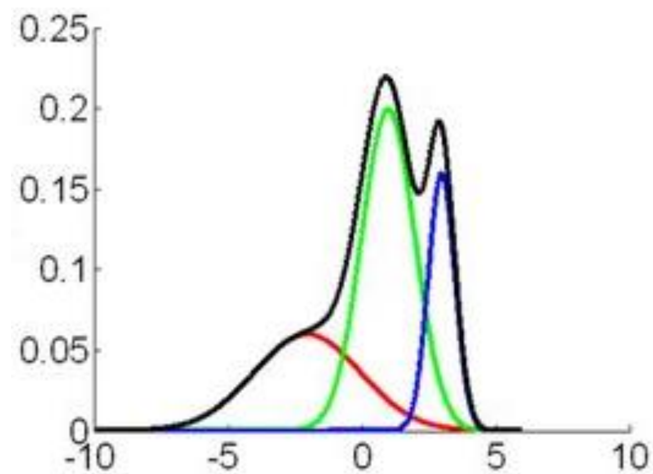


Discrete T

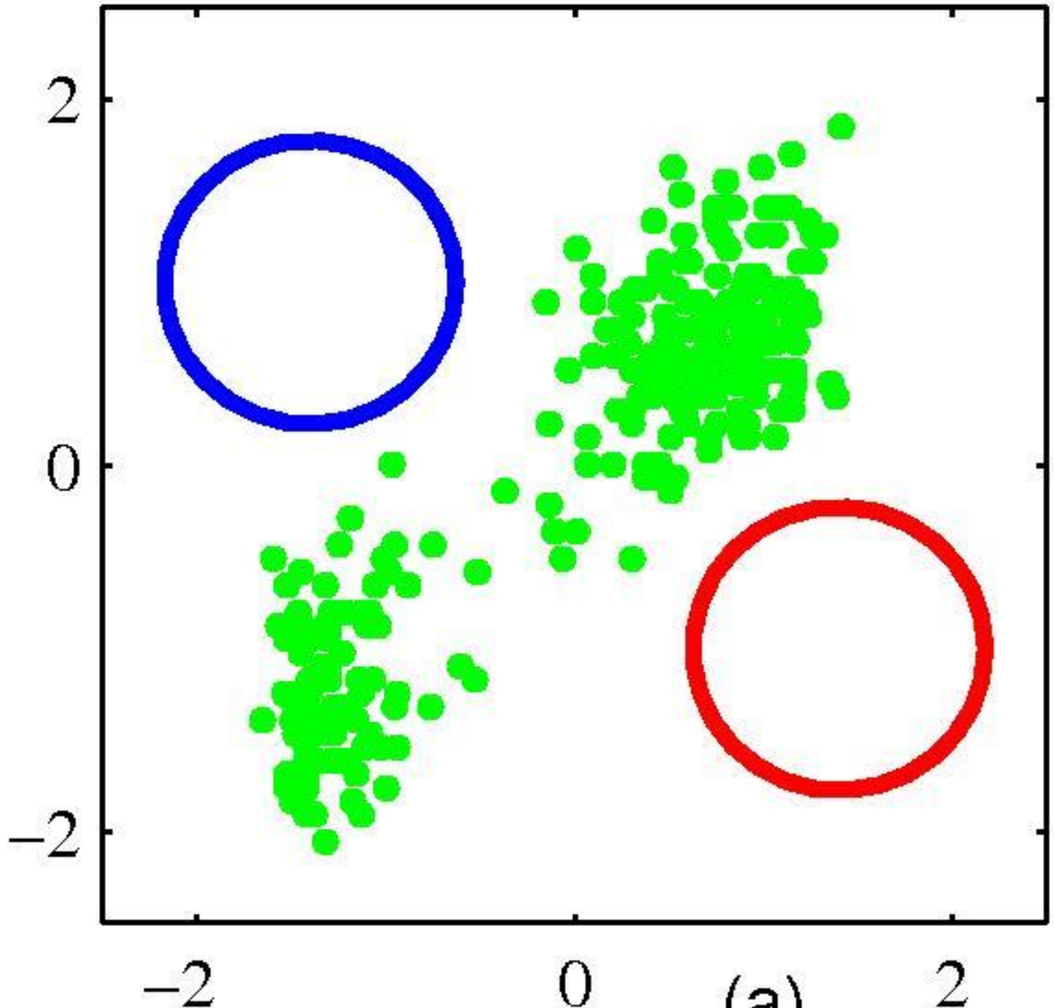
- Let $t \in \{1, \dots, K\}$, then

$$p(x|W) = \sum_{k=1}^{K} p(x|k, W)p(t = k)$$

- If each $p(x|k, W)$ defines a distribution from exponential class we may restore a mixture of distributions
- Additionally we find to which component each object belongs to – useful for clustering problems
- Classical example: mixture of gaussians

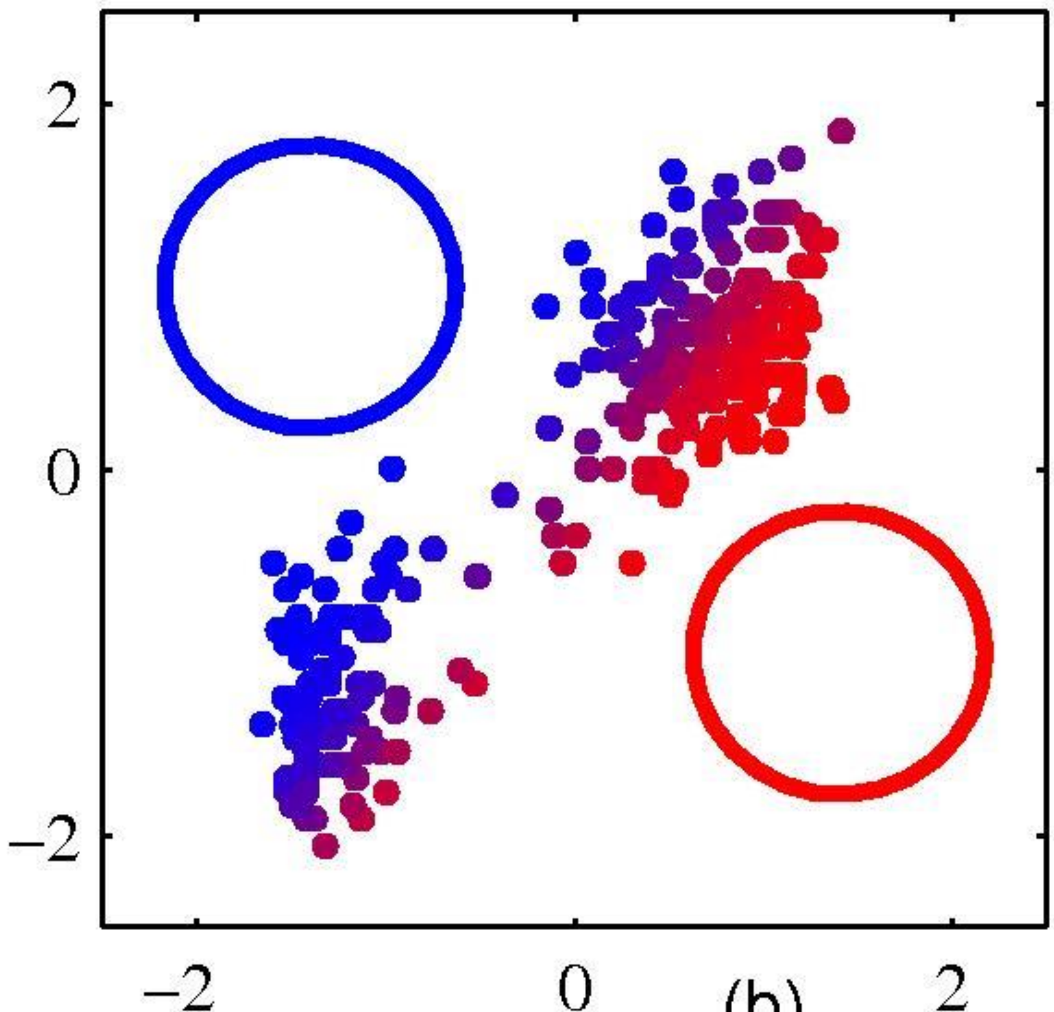


Mixture of gaussians



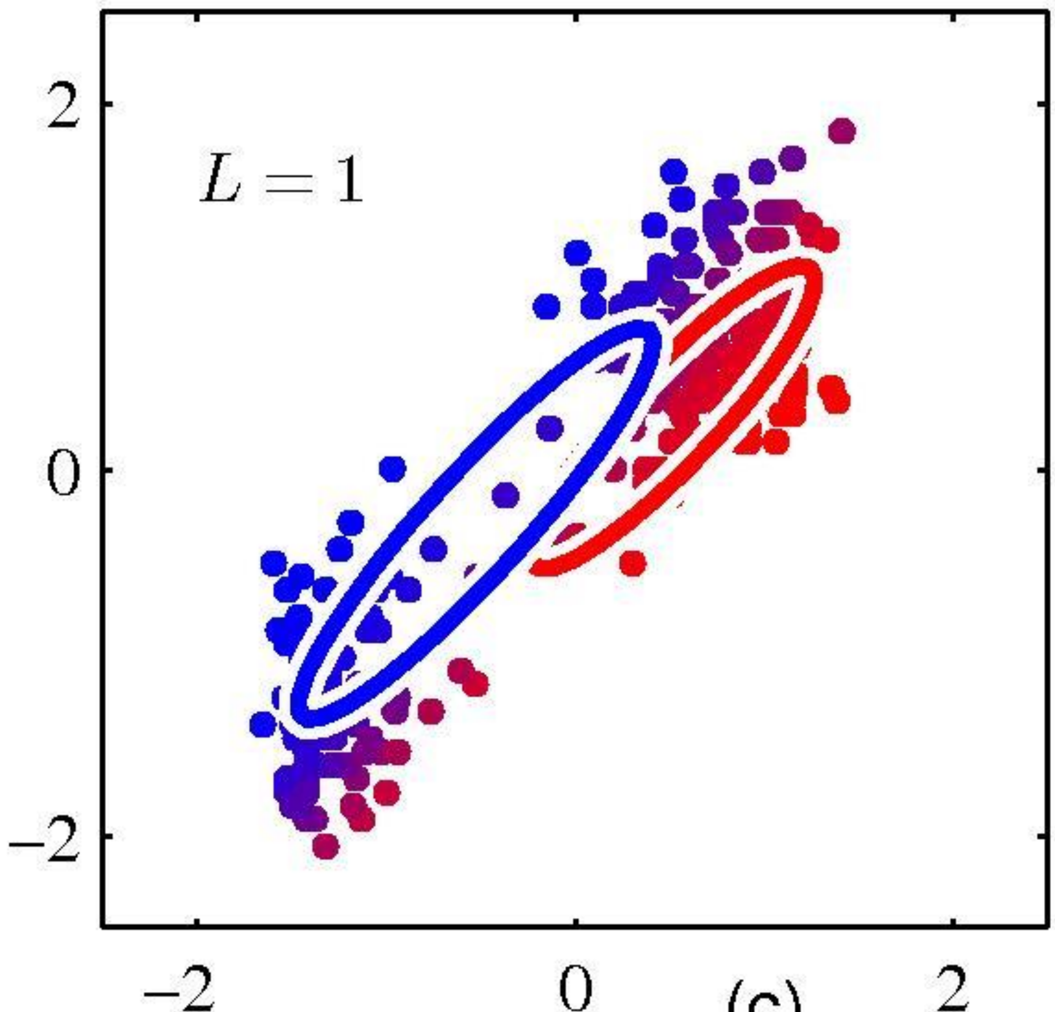
(a)

Mixture of gaussians



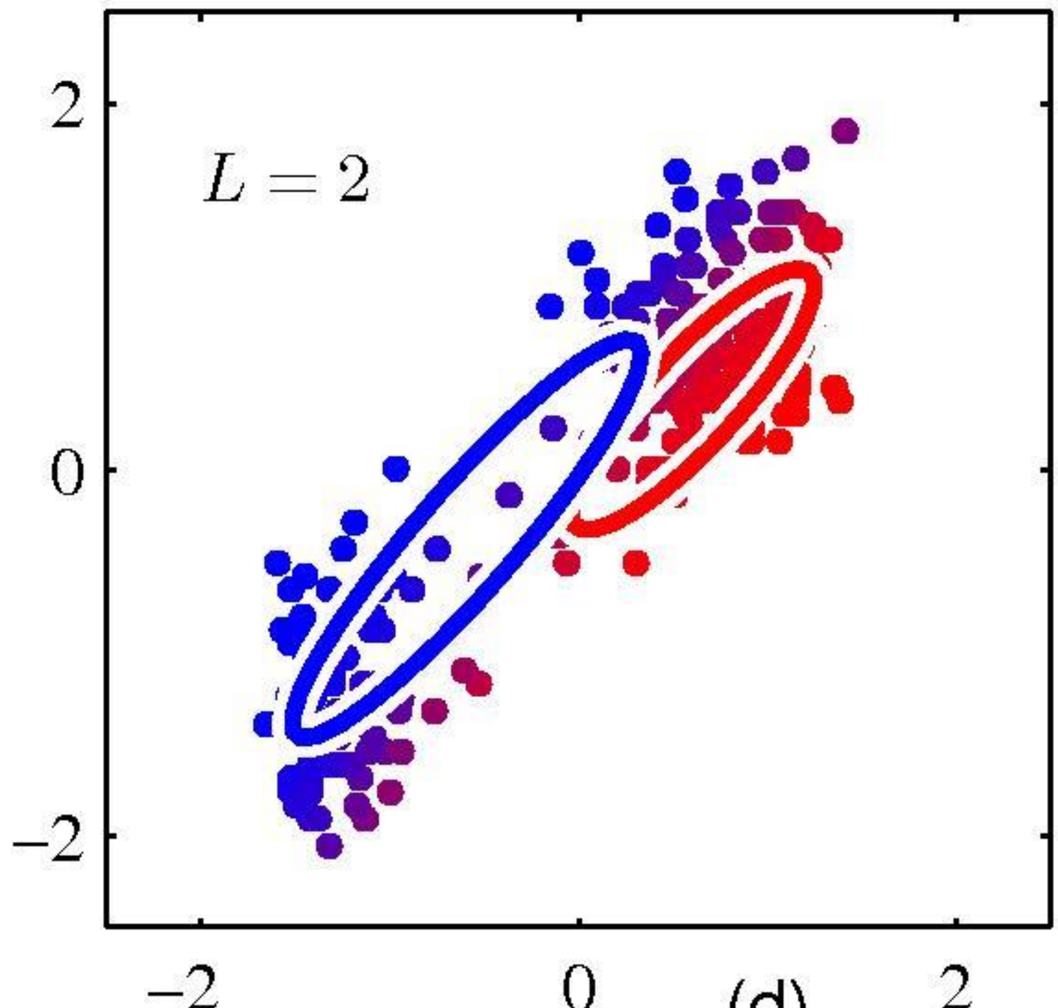
(b)

Mixture of gaussians



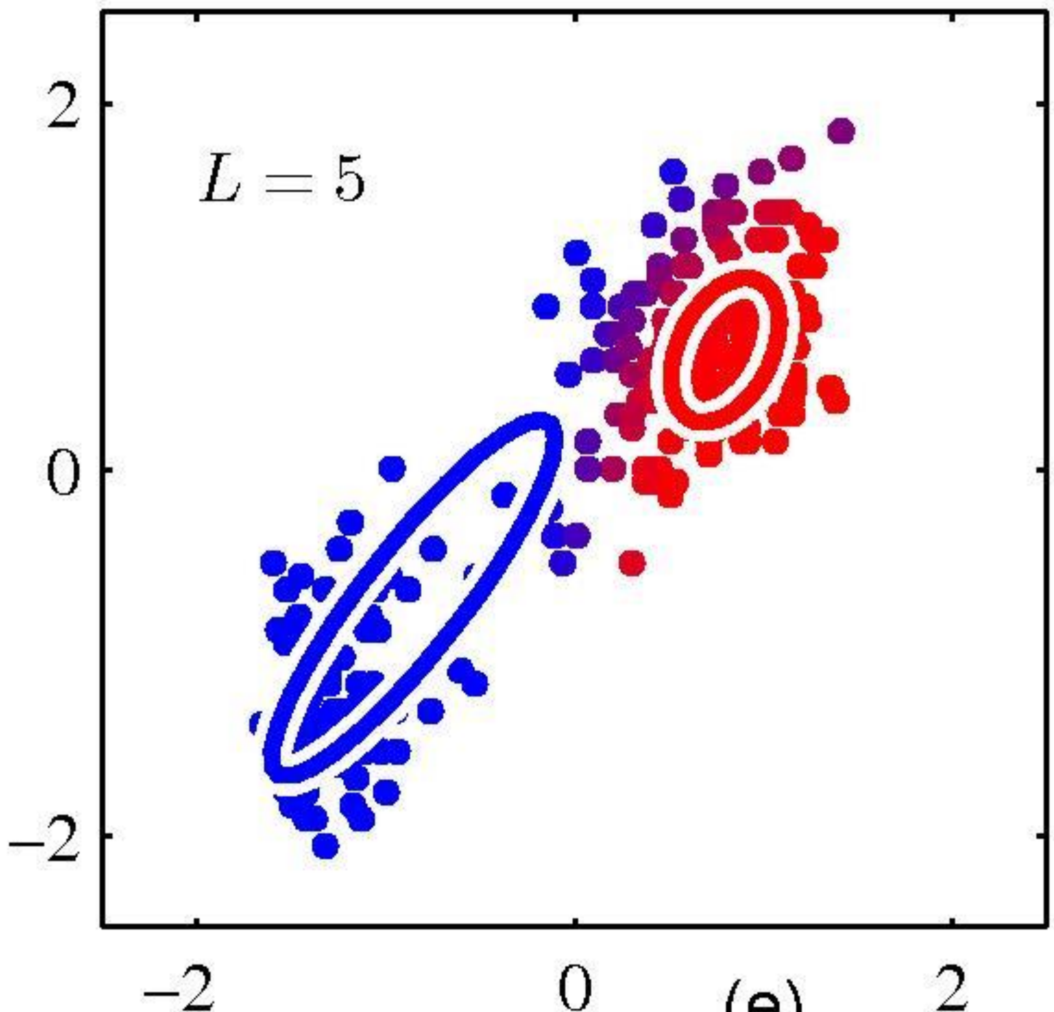
(c)

Mixture of gaussians



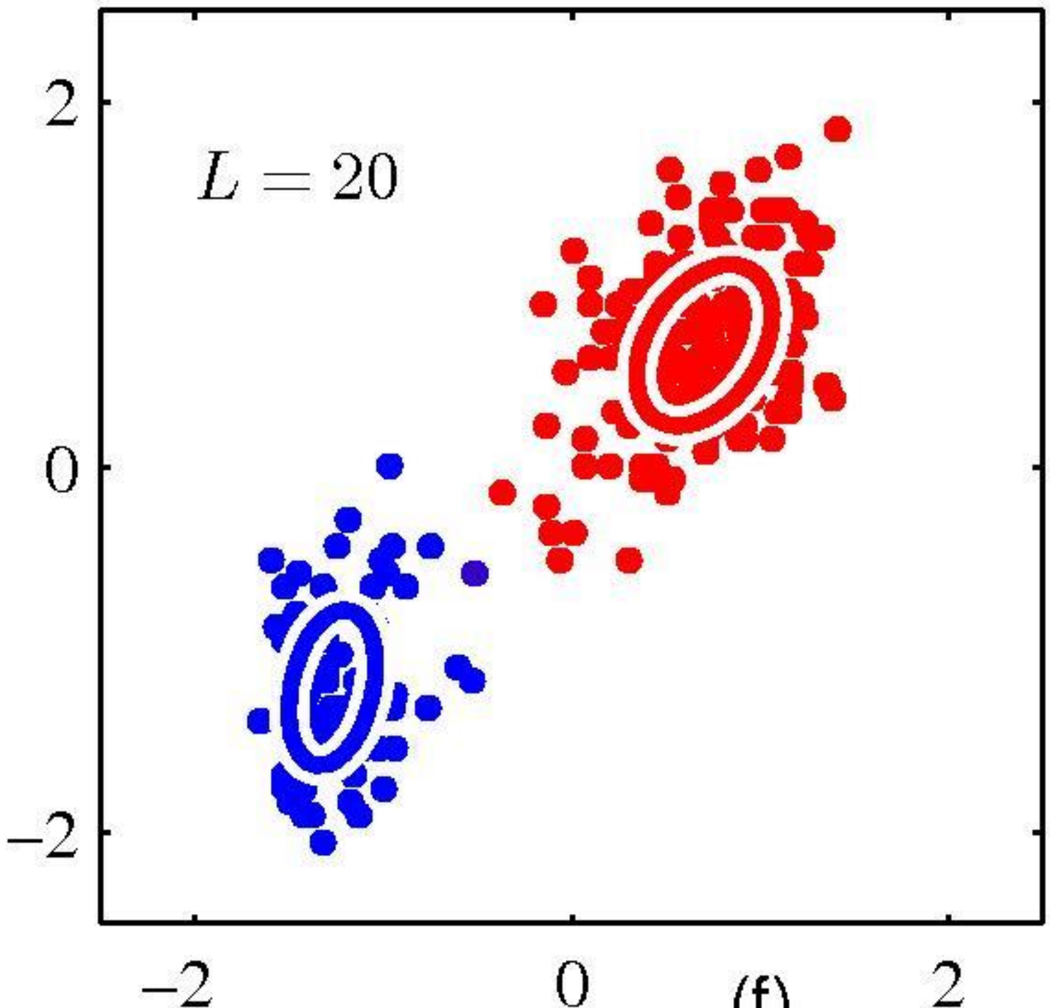
(d)

Mixture of gaussians



(e)

Mixture of gaussians



(f)

Mixture of gaussians: formal description

- Joint distribution

$$p(X,T|W) = \prod_{i=1}^{n} p(x_i|t_i,W)p(t_i|W) = \prod_{i=1}^{n} \mathcal{N}(x_i|\mu_{t_i},\Sigma_{t_i})\theta_{t_i},$$

  where θ is vector of probabilities $p(t_i = k) = \theta_k$ and (μ_k, Σ_k) are the parameters of k^{th} gaussian

- W consists of θ, $\{\mu_k\}$, $\{\Sigma_k\}$
- We may establish prior distributions on W if needed, e.g. penalizing too narrow gaussians
- We could still perform EM-algorithm for estimating $\arg\max p(W|X_{tr})$

EM-algorithm for mixture of gaussians

- Probabilistic model

$$p(X,T|W) = \prod_{i=1}^{n} p(x_i|t_i,W)p(t_i|W) = \prod_{i=1}^{n} \mathcal{N}(x_i|\mu_{t_i},\Sigma_{t_i})\theta_{t_i},$$

- Problem

$$p(X|W) = \sum_{T} p(X,T|W) \to \max_{W}$$

- E-step

$$\gamma_i(l) = \frac{\mathcal{N}(x_i|\mu_l,\Sigma_l)\theta_l}{\sum_{k=1}^{K}\mathcal{N}(x_i|\mu_k,\Sigma_k)\theta_k}$$

- M-step

$$n_k = \sum_{i=1}^{n}\gamma_i(k), \quad \mu_k = \frac{1}{n_k}\sum_{i=1}^{n}\gamma_i(k)x_i$$

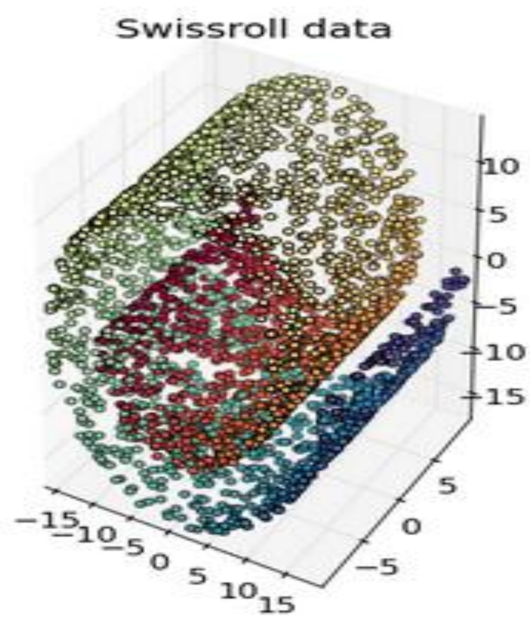
$$\Sigma_k = \frac{1}{n_k - 1}\sum_{i=1}^{n}\gamma_i(k)(x_i-\mu_k)(x_i-\mu_k)^T$$

Continuous T

- Continuous varuables can be regarded as a mixture of a continuum of distributions

$$p(x|W) = \int p(x,t|W)dt = \int p(x|t,W)p(t|W)dt$$

- They are more tricky to perform inference
- Need to check conjugacy property in order to perform E-step explicitly
- Typically used for dimension reduction

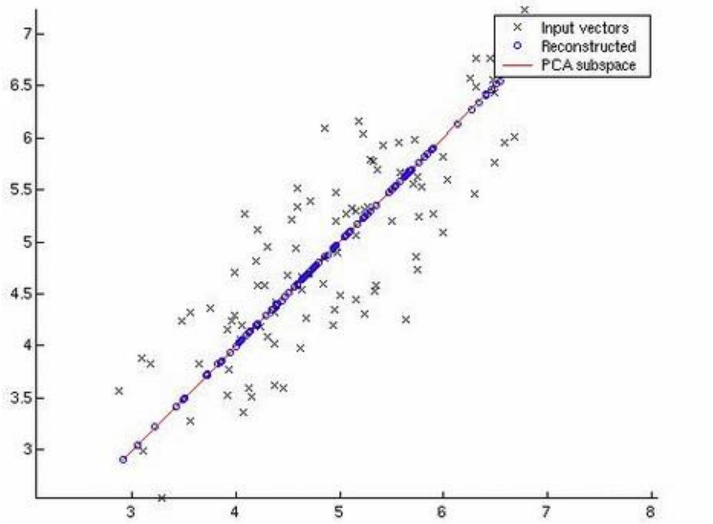


Example: PCA model

- Consider $x \in \mathbb{R}^D$, $t \in \mathbb{R}^d$, such that $D \gg d$
- Joint distribution

$$p(X, T|W) = \prod_{i=1}^{n} p(x_i|t_i, W)p(t_i|W) = \prod_{i=1}^{n} \mathcal{N}(x_i|Vt_i, \sigma^2 I)\mathcal{N}(t_i|0, I)$$

- W consists of $D \times d$ matrix V and scalar σ
- Can use EM-algorithm to find $\arg\max_W p(X_{tr}|W)$



Advantages of EM PCA

In PCA the explicit equation for W can be obtained analytically. Then why use EM?..

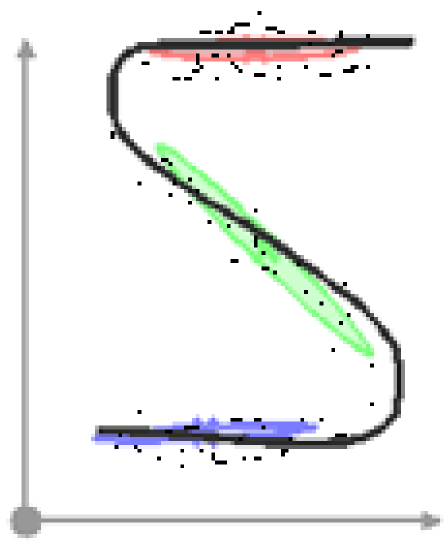
- EM updates have complexity $O(nDd)$ instead of $O(nD^2)$ in analytic solution
- Can process missing parts in X and present parts in T
- Can determine d if $p(W)$ is established
- Can be extended to more general models such as mixture of PCA and variational auto-encoders

Mixture of PCA

- Two types of latent variables: discrete $z \in \{1, \ldots, K\}$ and continuous $t \in \mathbb{R}^d$
- Joint distribution

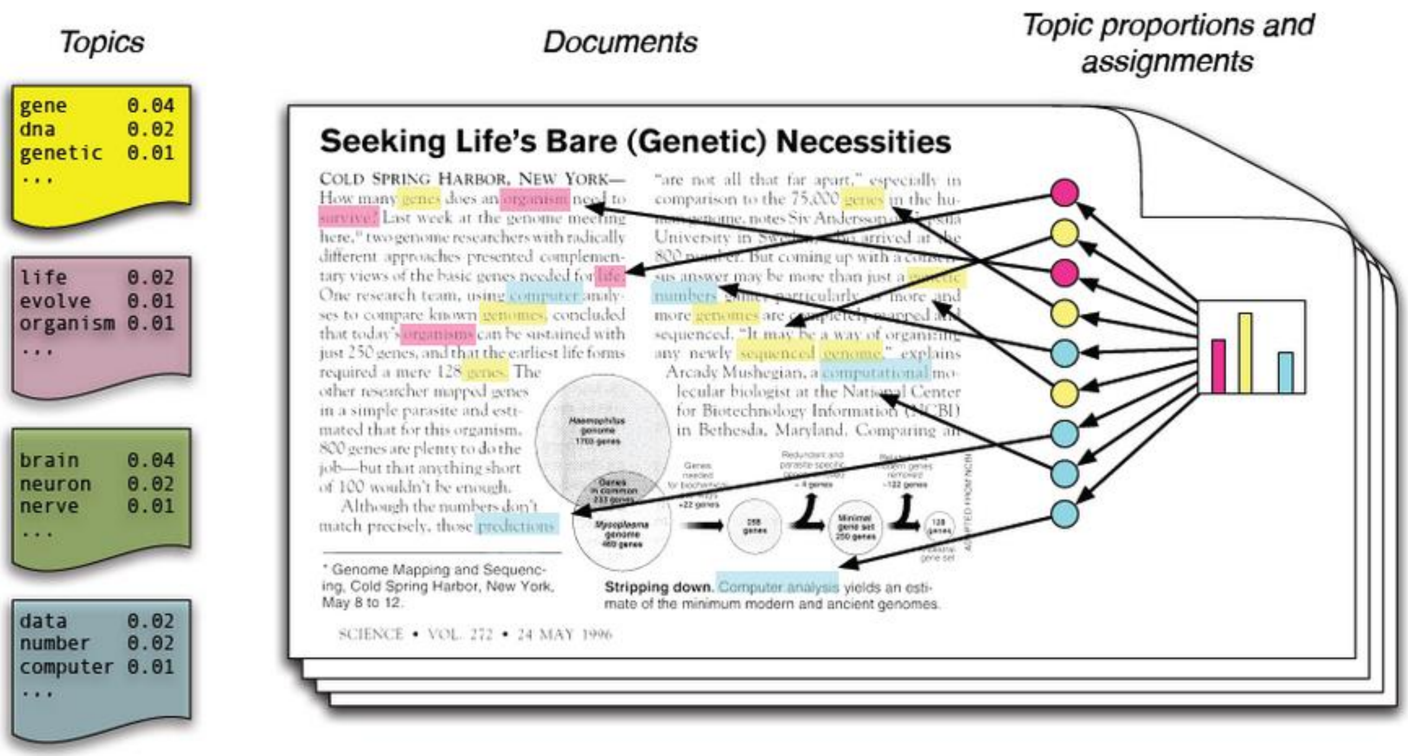
$$p(X, Z, T|W) = \prod_{i=1}^{n} p(x_i|t_i, z_i, W)p(t_i|W)p(z_i|W) = \prod_{i=1}^{n} \mathcal{N}(x_i|V_{z_i}t_i, \sigma_{z_i}^2 I)\mathcal{N}(t_i|0, I)\theta_{z_i}$$

- W consists of matrices $\{V_k\}$, scalars $\{\sigma_k\}$, and vector of probabilities θ such that $p(z_i = k) = \theta_k$
- Can be used for non-linear dimension reduction



Example: Latent Dirichlet Allocation

- Popular generative model for **texts**
- Each text is considered as a mixture of few **topics**
- Each topic is a **distribution** over words



LDA: formal description

$$p(X, Z, \Psi, \Phi) = \prod_{d=1}^{D} \left(p(\phi_d) \prod_{i=1}^{N_d} p(x_{di}|\psi_{z_{di}}) p(z_{di}|\phi_d) \right) \prod_{t=1}^{T} p(\psi_t)$$

$p(\psi_t) \sim \mathcal{D}(\psi_t|\alpha)$ Distribution of words in topic t

$p(\phi_d) \sim \mathcal{D}(\phi_d|\beta)$ Distribution of topics in document d

$p(z_{di}|\phi_d) = \phi_{d,z_{di}}$ Probability of ith word in document d belongs to topic z_{di}

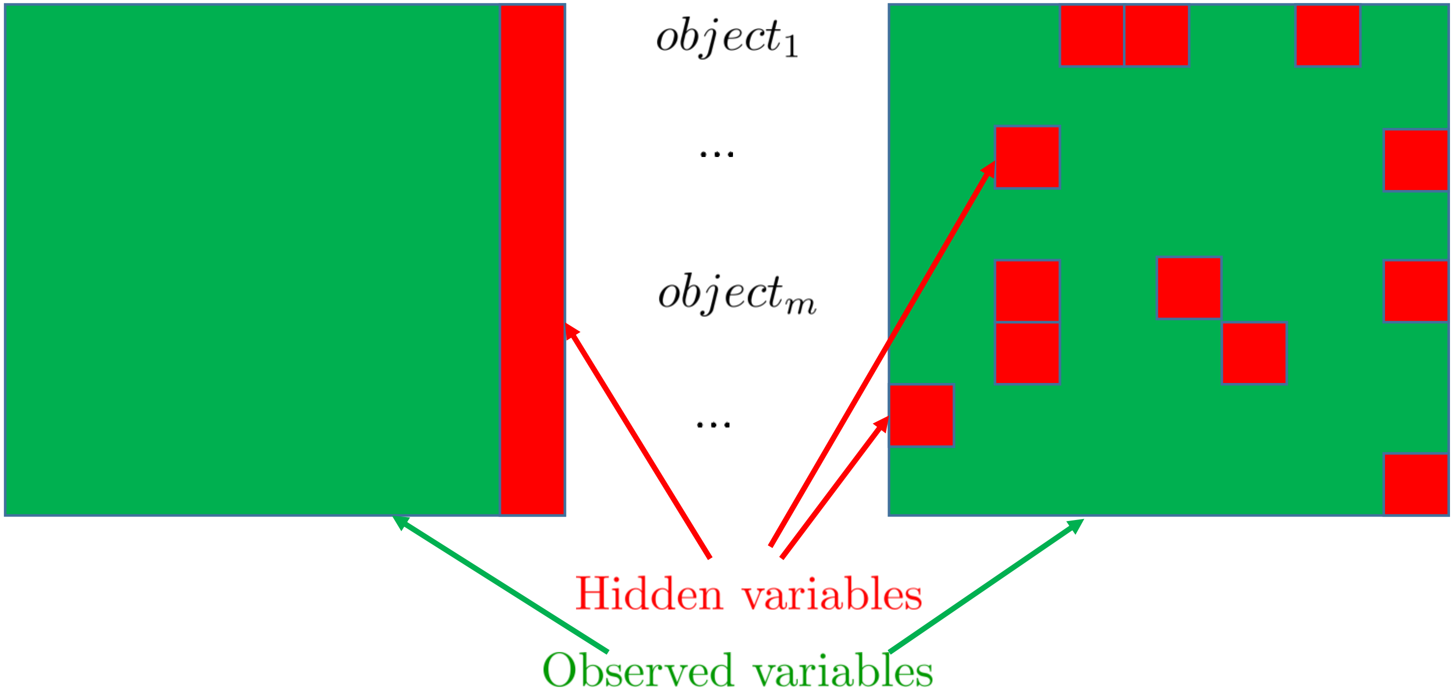
$p(x_{di}|\psi_{z_d i}) = \psi_{z_{di},x_{di}}$ Probability of word w_{di} belongs to topic z_{di}

Given: $\{X_d\}_{d=1}^{D}, \alpha, \beta, T$

Required: $p(\Psi|X) \to \max\limits_{\Psi}$

There exist multiple extensions of LDA model which take into account additional information about the problem (microtexts, sequential data, preferences on predefined words, etc.) and its modifications to **collaborative filtering**

General nature of EM-framework



- EM algorithm allows processing arbitrary missing data
- May deal with both discrete and continuous variables
- Always converges
- Allows multiple extensions

Variational inference: way to complex Bayesian models

- Inference becomes optimization
- Instead of computing

$$p(\theta|Data) = \frac{p(Data|\theta)p(\theta)}{\int p(Data|\theta)p(\theta)d\theta}$$

Variational inference: way to complex Bayesian models

- Inference becomes optimization
- Instead of computing

$$p(\theta|Data) = \frac{p(Data|\theta)p(\theta)}{\boxed{\int p(Data|\theta)p(\theta)d\theta}}$$

The most difficult part

Variational inference: way to complex Bayesian models

- Inference becomes optimization
- Instead of computing

$$p(\theta|Data) = \frac{p(Data|\theta)p(\theta)}{\boxed{\int p(Data|\theta)p(\theta)d\theta}}$$

The most difficult part

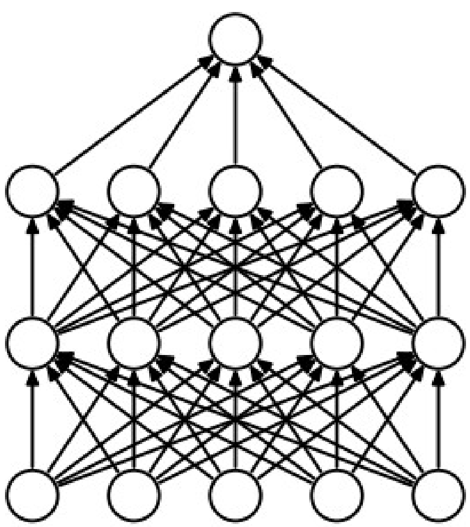
we optimize variational lower bound

$$\mathcal{L}(q) = \int q(\theta|\phi) \log \frac{p(Data, \theta)}{q(\theta|\phi)} dW \to \max_{\phi}$$

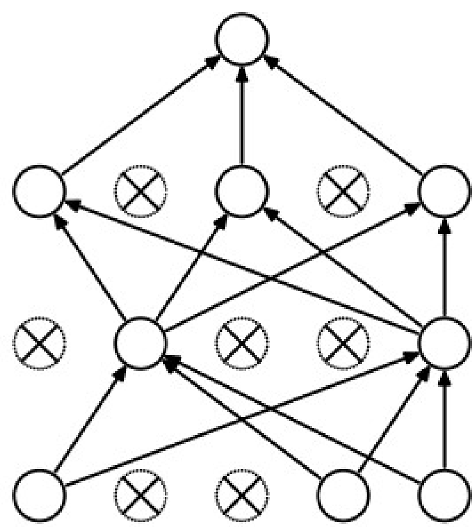
- Can use **deep neural networks** to model $q(\theta|\phi)$

Dropout

- Proposed by Geoffrey Hinton's group in 2012
- Nullifies the outputs of randomly selected neurons at each iteration of training
- Purely heuristic procedure for preventing overfitting
- Can be justified and generalized from Bayesian point of view



(a) Standard Neural Net



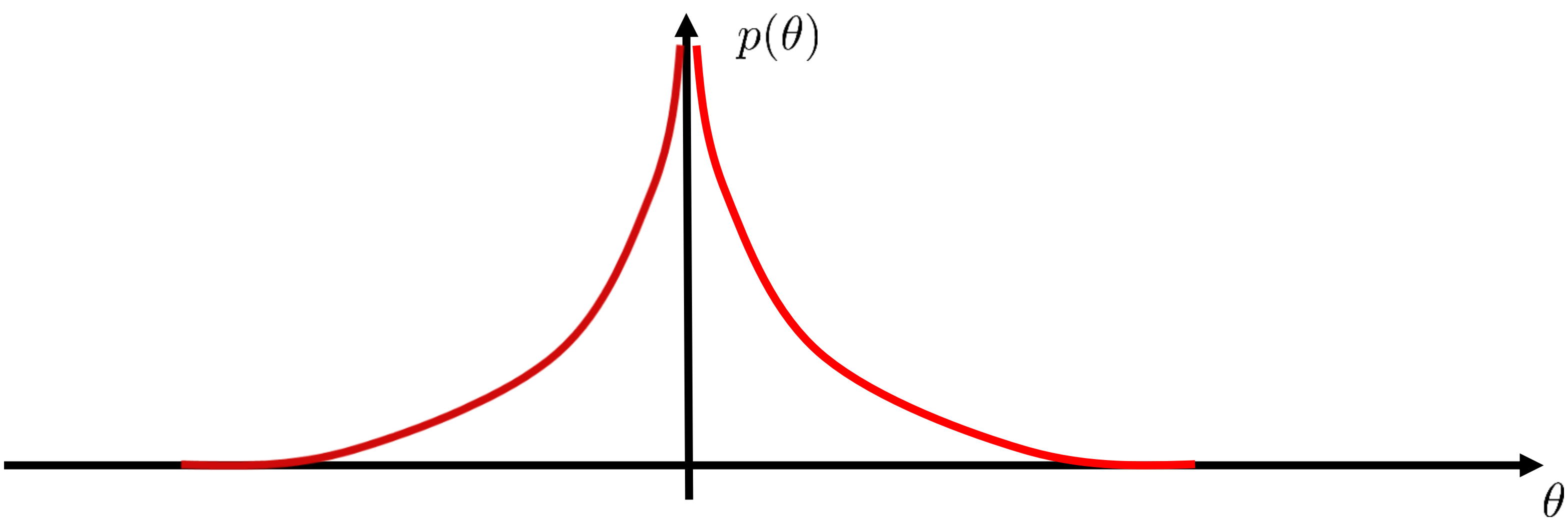
(b) After applying dropout.

Variational Dropout

- In December 2015 new techniques of variational dropout was suggested (Kingma15)
- It was shown that dropout correponds to Bayesian inference with special improper prior over the weights θ

$$p(\theta) \propto \frac{1}{|\theta|}$$

- This is so-called **scale-invariant prior** which penalizes the precision of θ



Diederik P. Kingma, Tim Salimans, Max Welling. Variational Dropout and the Local Reparameterization Trick. arXiv:1506.02557

Variational Dropout

- We approximate posterior as gaussian distribution

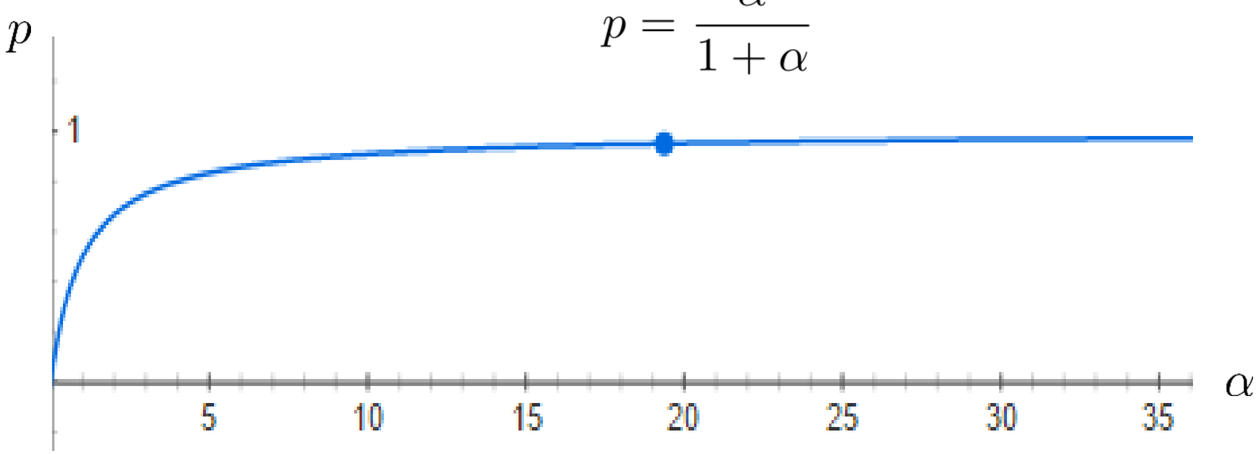
$$p(\theta|Data) \approx q(\theta) = \prod_{i,j} q(\theta_{ij}) = \prod_{i,j} \mathcal{N}(\theta_{ij}|\mu_{ij}, \alpha\mu_{ij}^2)$$

- Stochastic optimization of variational lower bound

$$\mathcal{L}(q) = \int q(\theta) \log \frac{p(Data|\theta)p(\theta)}{q(\theta)} \to \max_{\mu}$$

w.r.t. all μ_{ij} given α fixed correponds to standard droupout learning with dropout rate

$$p = \frac{\alpha}{1+\alpha}$$

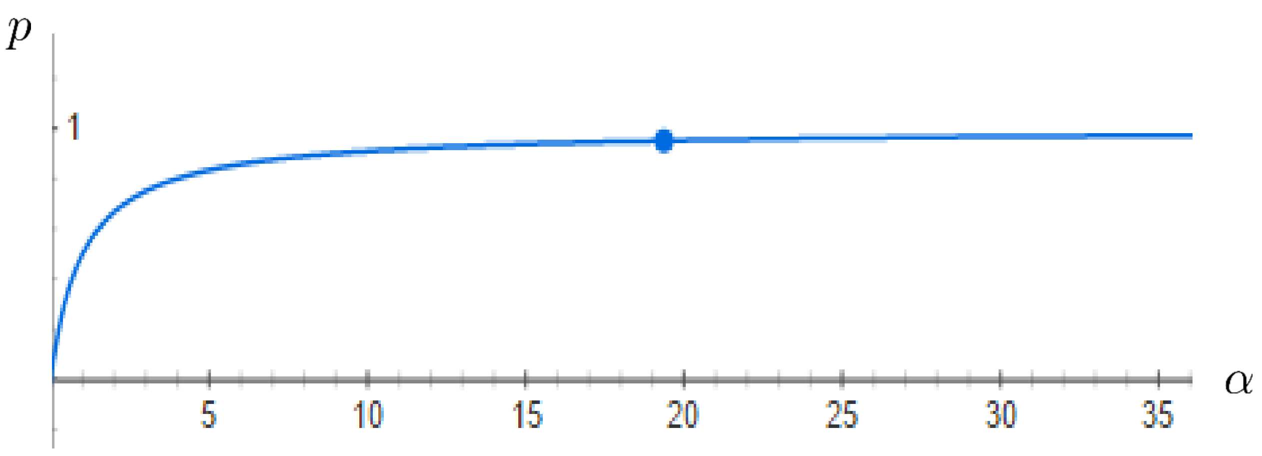


Adjusting dropout rates

- But we can also optimize $\mathcal{L}(q)$ w.r.t. α

$$\mathcal{L}(q) = \int q(\theta) \log \frac{p(Data|\theta)p(\theta)}{q(\theta)} \to \max_{\mu,\alpha}$$

- This would make an approximation of posterior even more accurate
- We obtained the proper way of setting dropout rate automatically!



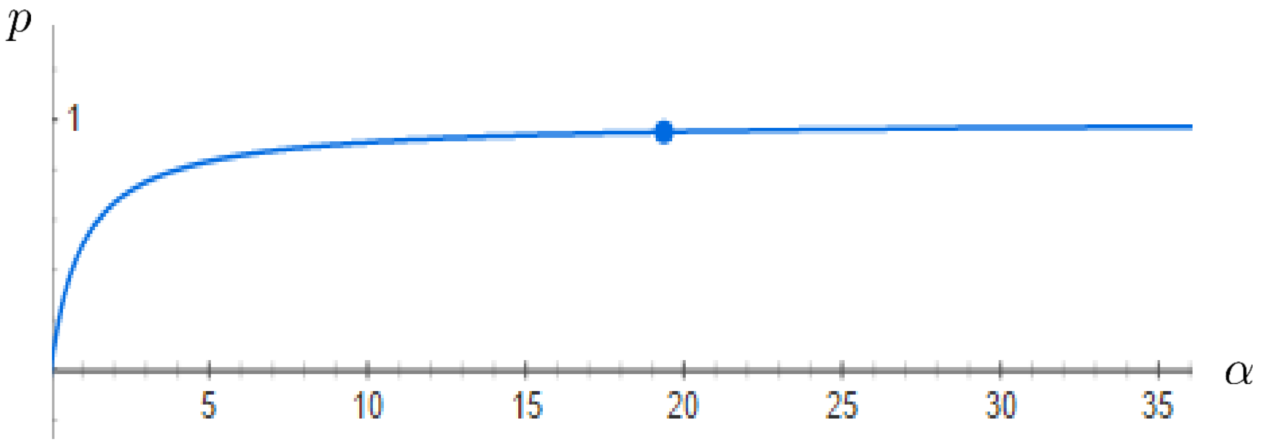
Individual dropout rate

- No we go even further and set variational family

$$q(\theta) = \prod_{i,j} \mathcal{N}(\theta_{ij} | \mu_{ij}, \alpha_{ij}\mu_{ij}^2)$$

  It corresponds to **individual** dropout rates for each weight

- The approximation becomes only more accurate
- We can show that if $\alpha_{ij} \to +\infty$ then $\mu_{ij} \to 0$ and $\alpha_{ij}\mu_{ij}^2 \to 0$



Alternative view on dropout

- Split lower bound on two parts

$$\mathcal{L}(q) = \int q(\theta) \log \frac{p(Data|\theta)p(\theta)}{q(\theta)} d\theta = \int q(\theta) \log p(Data|\theta) d\theta - KL(q(\theta)||p(\theta))$$

- Dropout can be viewed as regularization

Alternative view on dropout

- Split lower bound on two parts

$$\mathcal{L}(q) = \int q(\theta) \log \frac{p(Data|\theta)p(\theta)}{q(\theta)} d\theta = \underbrace{\int q(\theta) \log p(Data|\theta) d\theta}_{\text{Data term}} \underbrace{- KL(q(\theta)||p(\theta))}_{\text{Regularization}}$$

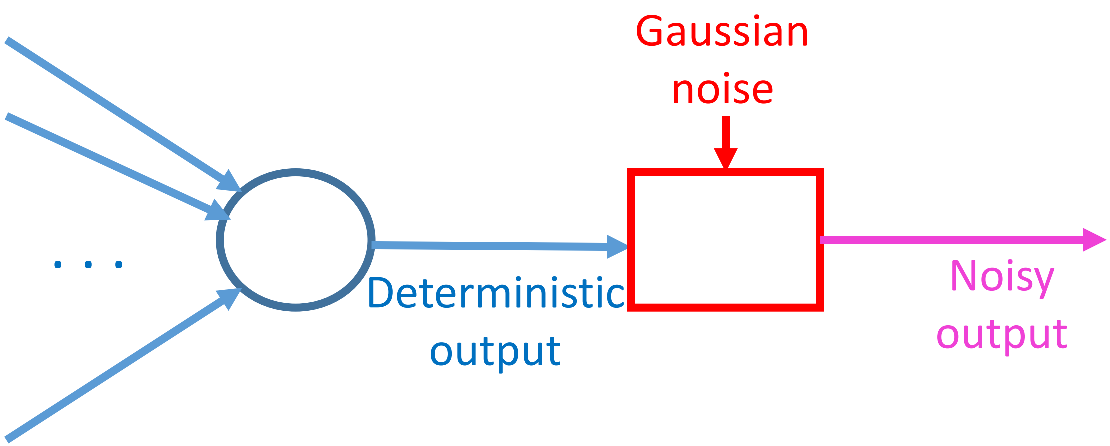
- Dropout can be viewed as regularization

Alternative view on dropout

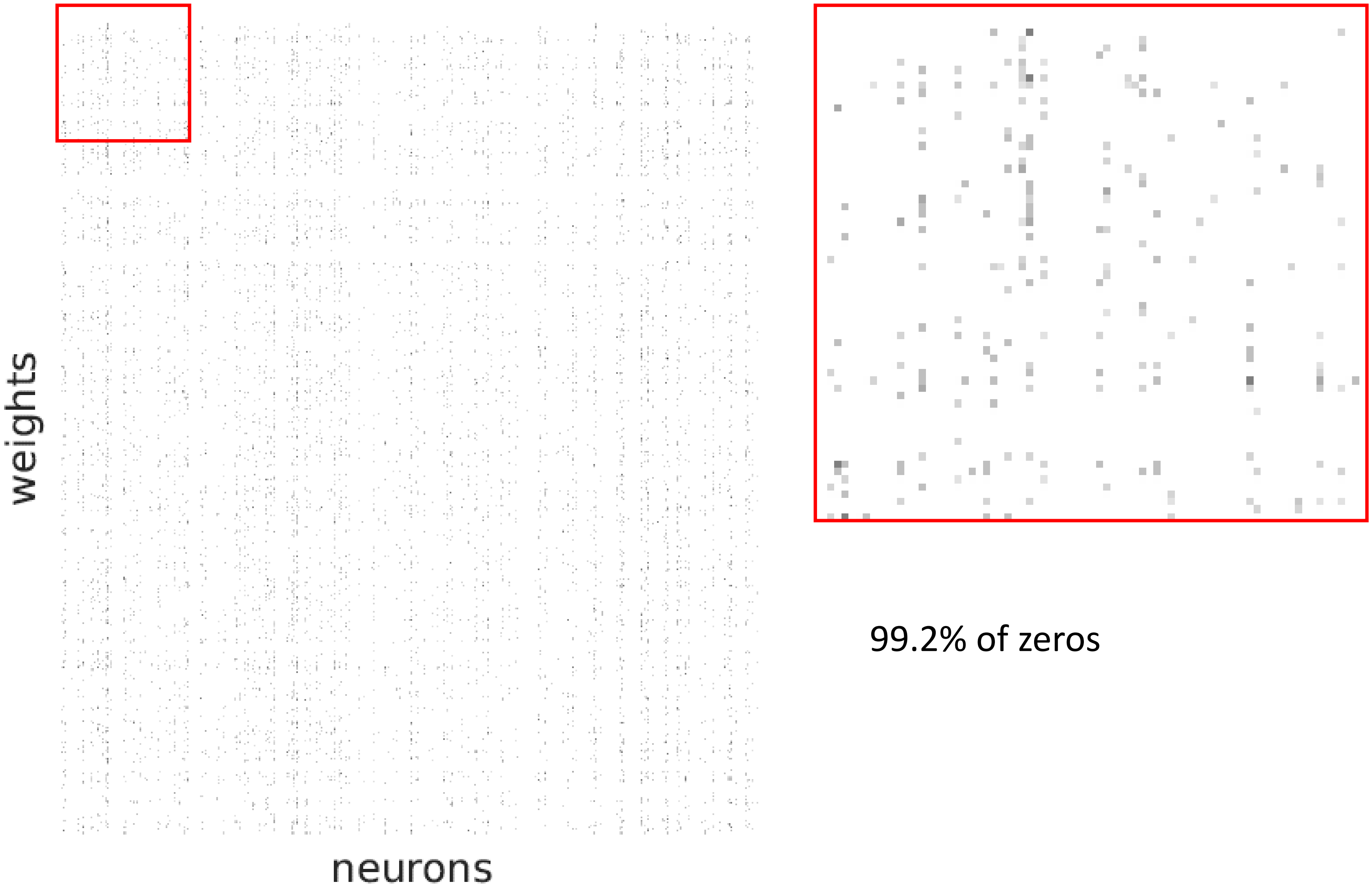
- Split lower bound on two parts

$$\mathcal{L}(q) = \int q(\theta) \log \frac{p(Data|\theta)p(\theta)}{q(\theta)} d\theta = \int q(\theta) \log p(Data|\theta) d\theta - KL(q(\theta)||p(\theta))$$

- Dropout can be viewed as regularization
- Alternative view is **noisification** which prevents overfitting on training data



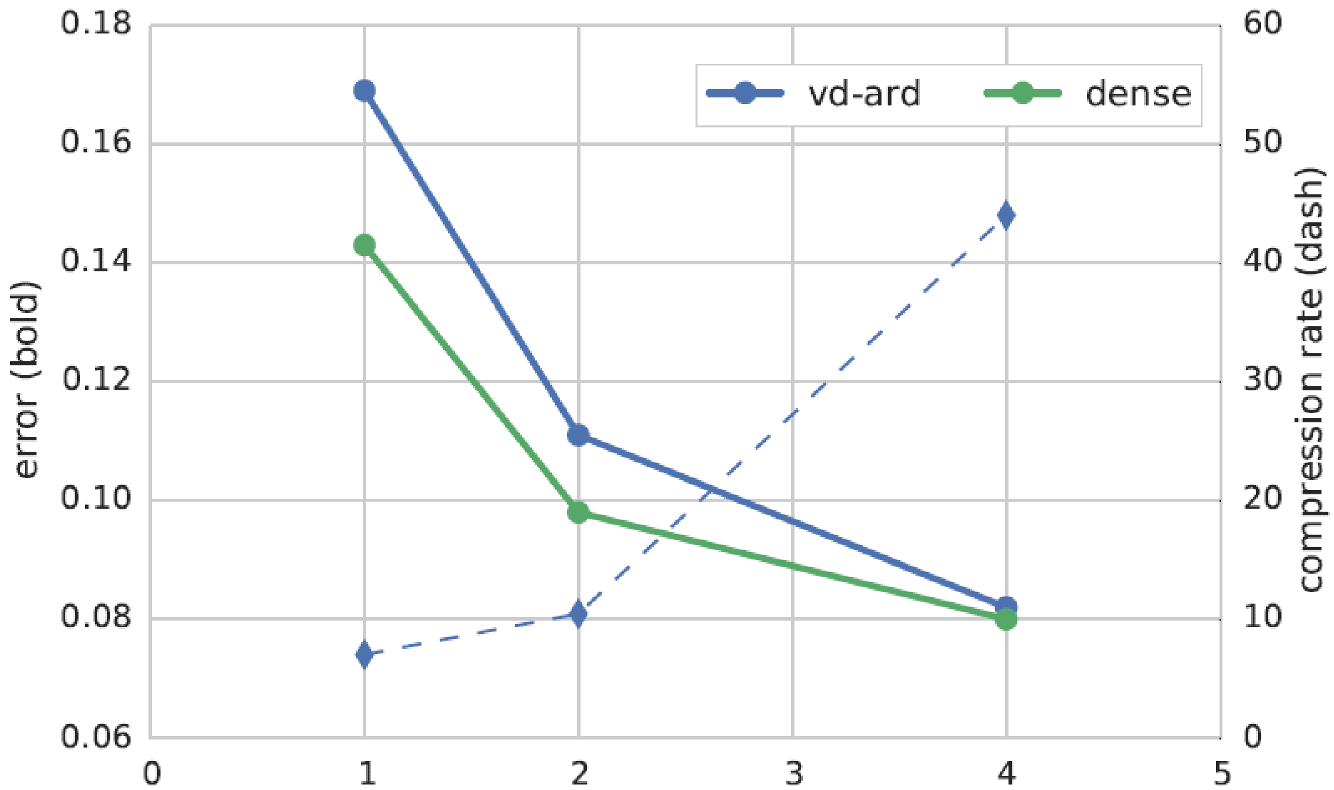
Sparsifying deep NNs



99.2% of zeros

Sparsifying deep NNs

- Large α_{ij} is equivalent to **elimination** of the corresponding weight from the model
- Surprizingly it turns out that the most of weights are not needed



Summer school

- In August we organize summer school on Deep Bayesian models
- Call for applications has been made. The deadline is 31st of March
- Hot topics: Attention models, variational auto-encoders, adversarial networks, deep reinforcement learning, stochastic optimization, normalization, etc.
- Visit our website if interested `http://DeepBayes.ru`

Conclusion

- Bayesian framework is extremely powerful and extends ML tools
- We do have scalable algorithms for approximate Bayesian inference
- Bayes + Deep Learning = 
- Even the first attempts of neurobayesian inference give impressive results