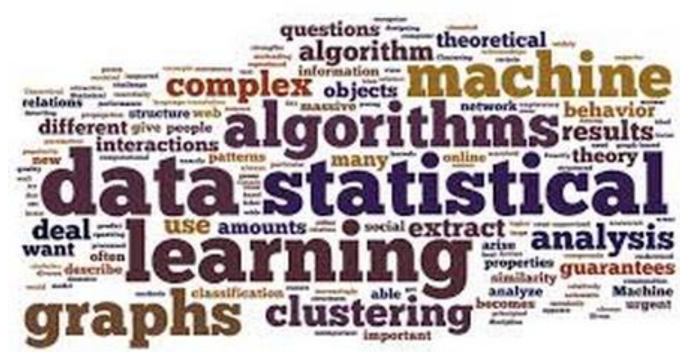
Bayesian Inference and Latent Variable Models in Machine Learning

Dmitry P. Vetrov
Head of Bayesian methods research group
http://bayesgroup.ru,
Faculty of Computer Science, HSE
Skoltech

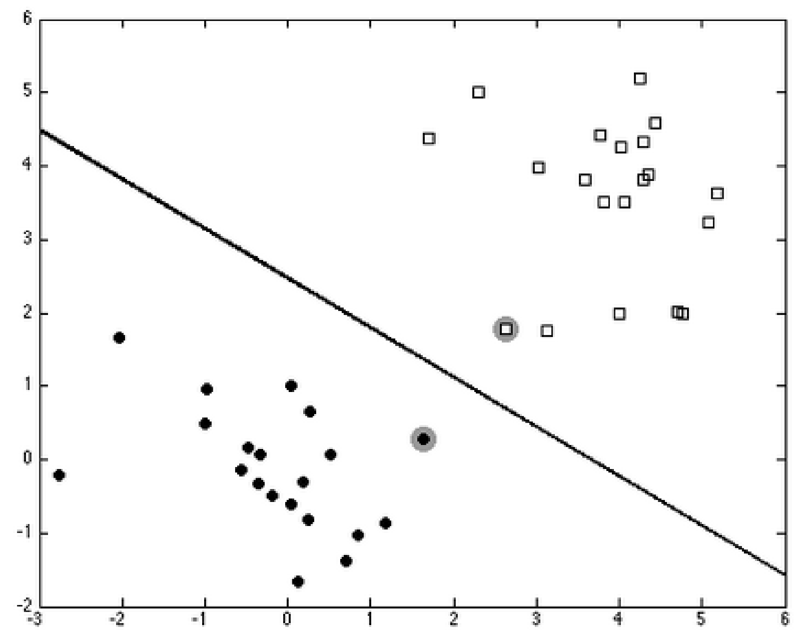
What is machine learning?

- 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 (X_{tr}, T_{tr})
- 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
- We know observed variables for the objects within the training set $X_{tr} = \{x_i\}_{i=1}^n$, $x_i \in \mathbb{R}^2$
- We know hidden variables for the objects from the training set that are binary labels $T = \{t_i\}_{i=1}^n$, $t_i \in \{-1, 1\}$
- After training we also know the weights W that define separating hyperplane: $W^T x + w_0$
- Now we are able to estimate binary hidden variable for the arbitrary observed x $\hat{t}(x) = \mathrm{sign}(W^T x + w_0)$



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

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

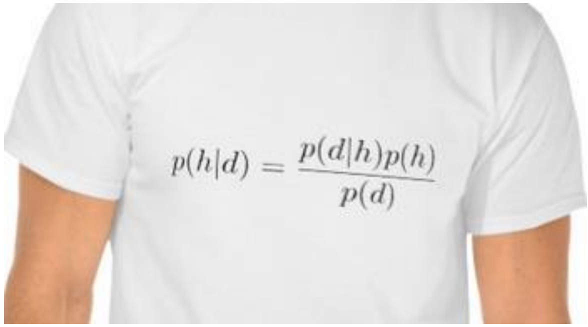
Bayesian Framework

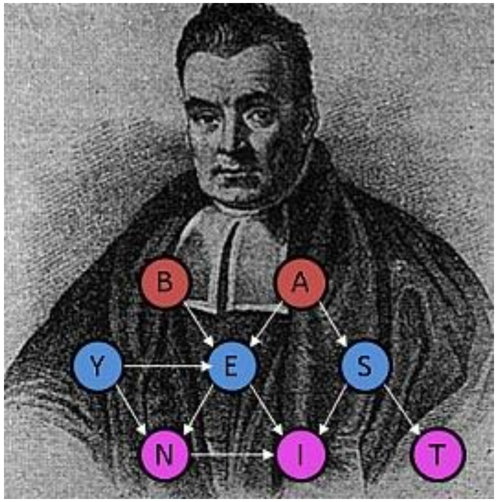
- Treats everything as random variables
- Encodes ignorance in terms of distributions
- Makes use of **Bayes Theorem**

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

- Possible to compute the estimate for arbitrary **unknown** variable (U) given **observed** data (O) and not having any knowledge about **latent** variables (L) from the joint distribution $p(U, O, L)$:

$$p(U|O) = \frac{\int p(U, O, L)dL}{\int p(U, O, L)dLdU}$$





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:

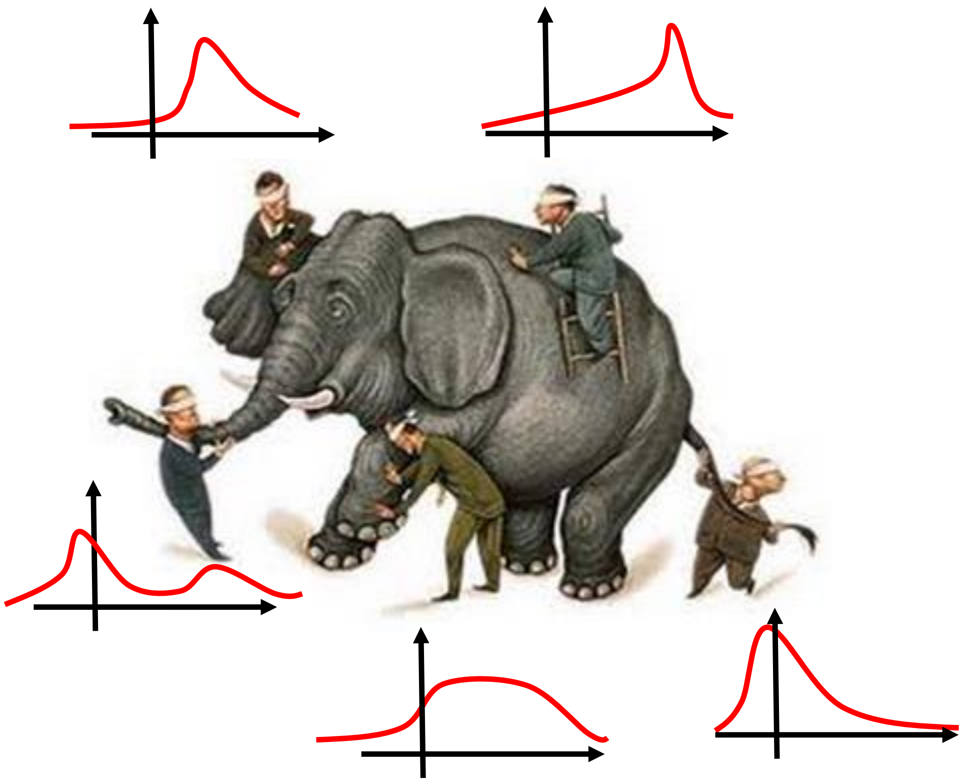
$$p(W|X_{tr}, T_{tr}) = \frac{p(T_{tr}, X_{tr}|W)p(W)}{\int p(T_{tr}, X_{tr}|W)p(W)dW}$$

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

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

Combining models

- Bayesian framework allows to combine different models
- We may build complex models from simpler ones using the latter as building blocks
- Posterior from one model may serve as a prior for the next model and so on



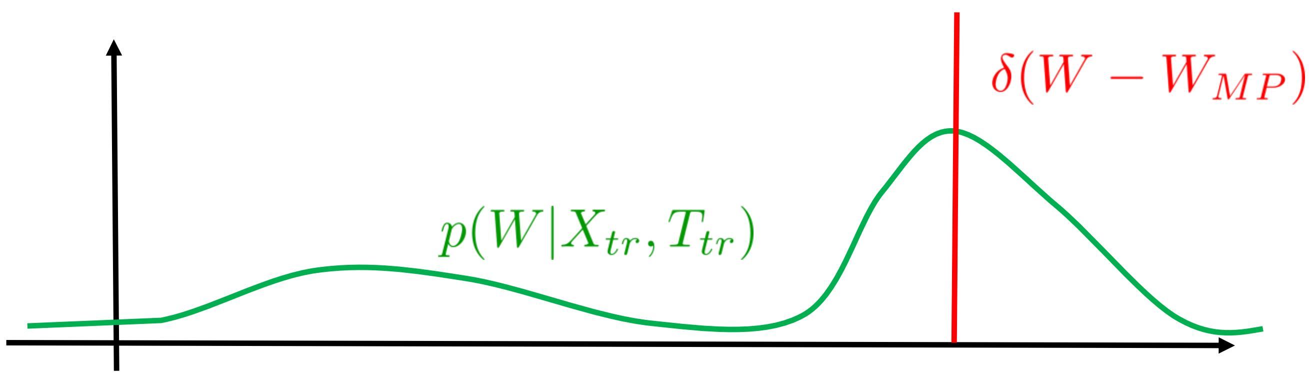
Maximal a posteriori (MAP) learning

- 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

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



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

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

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

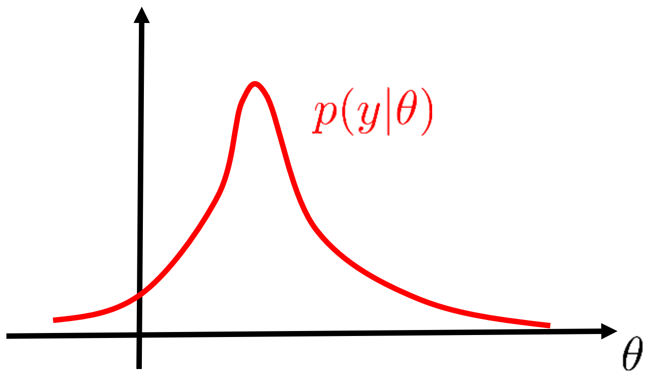
where $\theta_1 = -\frac{1}{2\sigma^2}$ and $\theta_2 = \frac{\mu}{\sigma^2}$

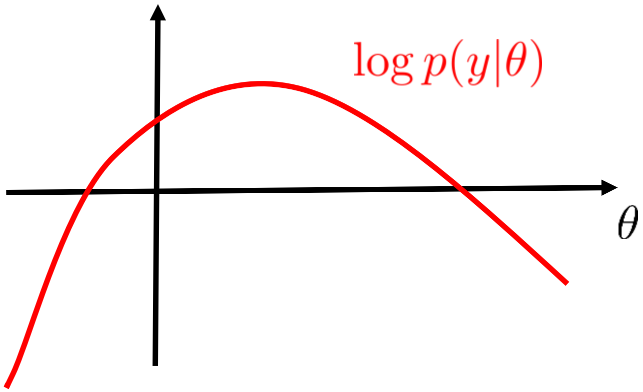
- Hence x and x^2 are sufficient statistics and

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

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

Log-concavity of exponential class





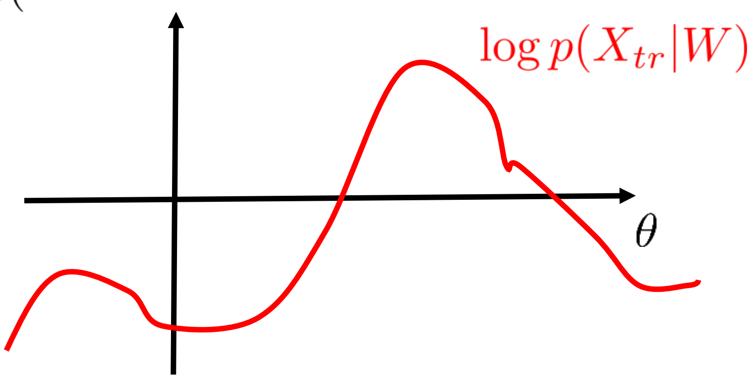
- For log-concave distributions maximum likelihood estimation can be done in an efficient manner
- All discrete distributions and many continuous (Gaussian, Laplace, Gamma, Dirichlet, Wishart, Beta, Chi-squared, etc.) belong to exponential class

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

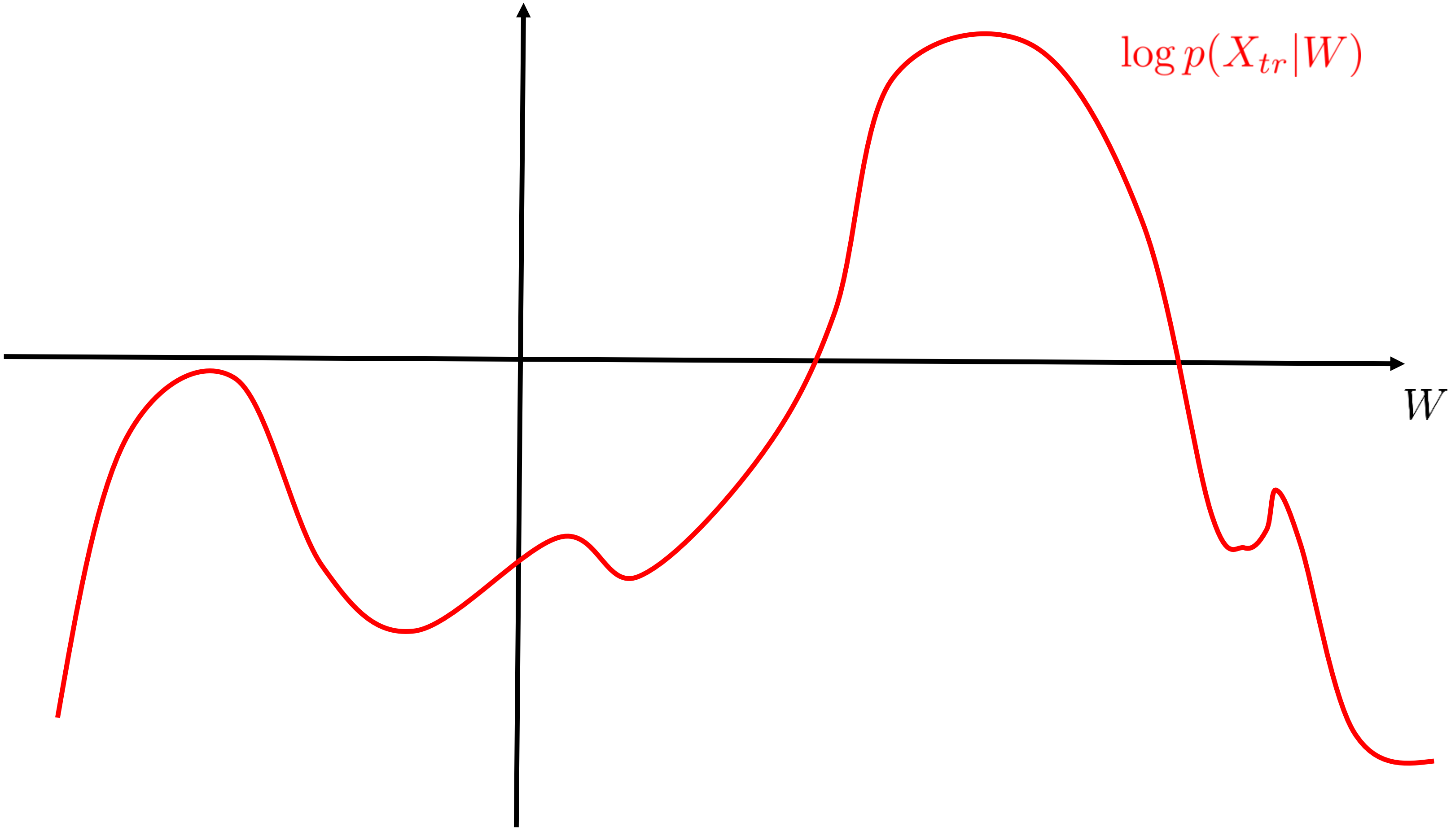
$$W_t = \arg\max_W \mathcal{L}(q_t, W) = \arg\max_W \int q_t(T) \log \frac{p(X_{tr}, T|W)}{q_t(T)} dT =$$
$$\arg\max_W \int q_t(T) \log p(X_{tr}, T|W) dT$$

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

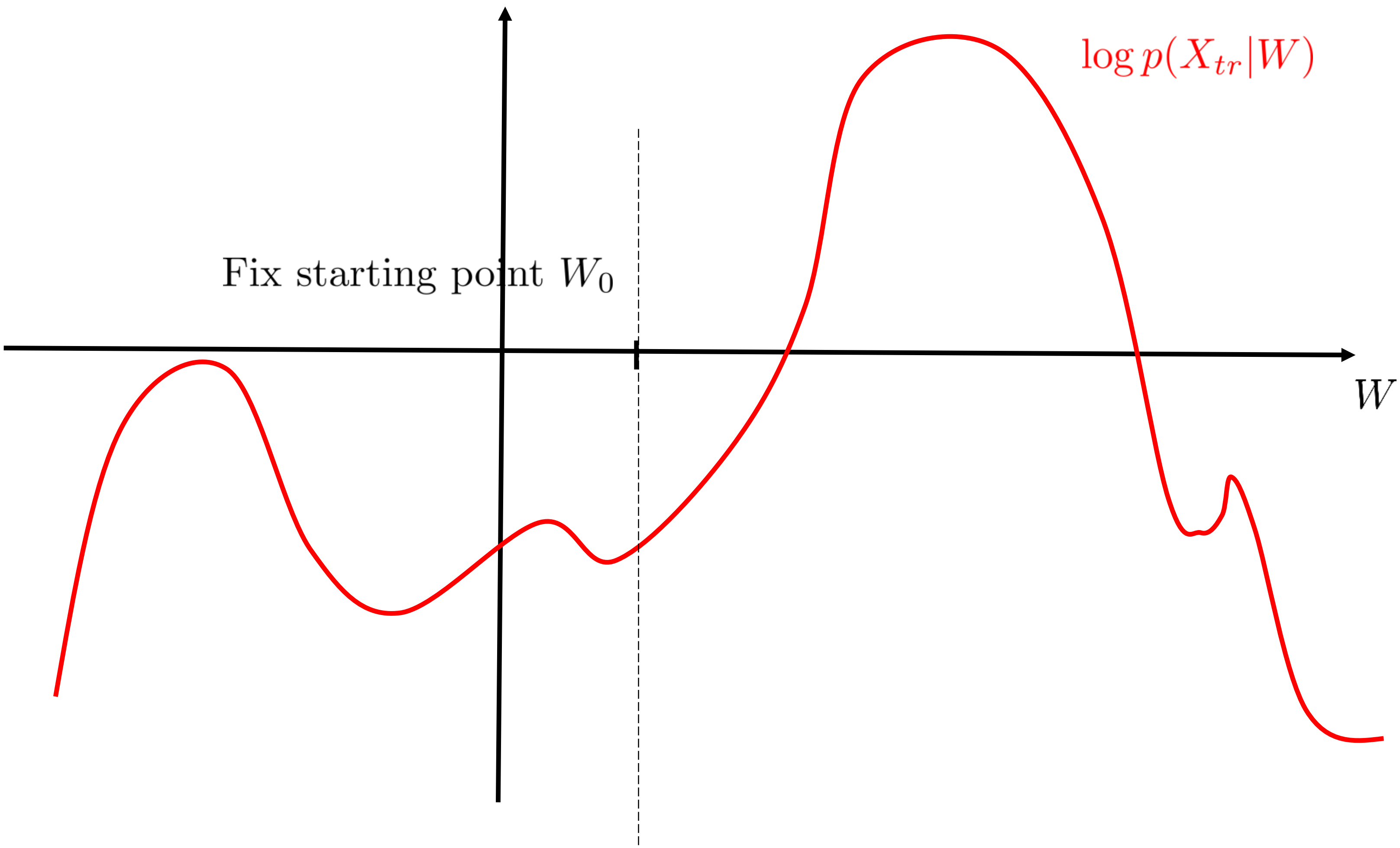
- Iterate until convergence

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

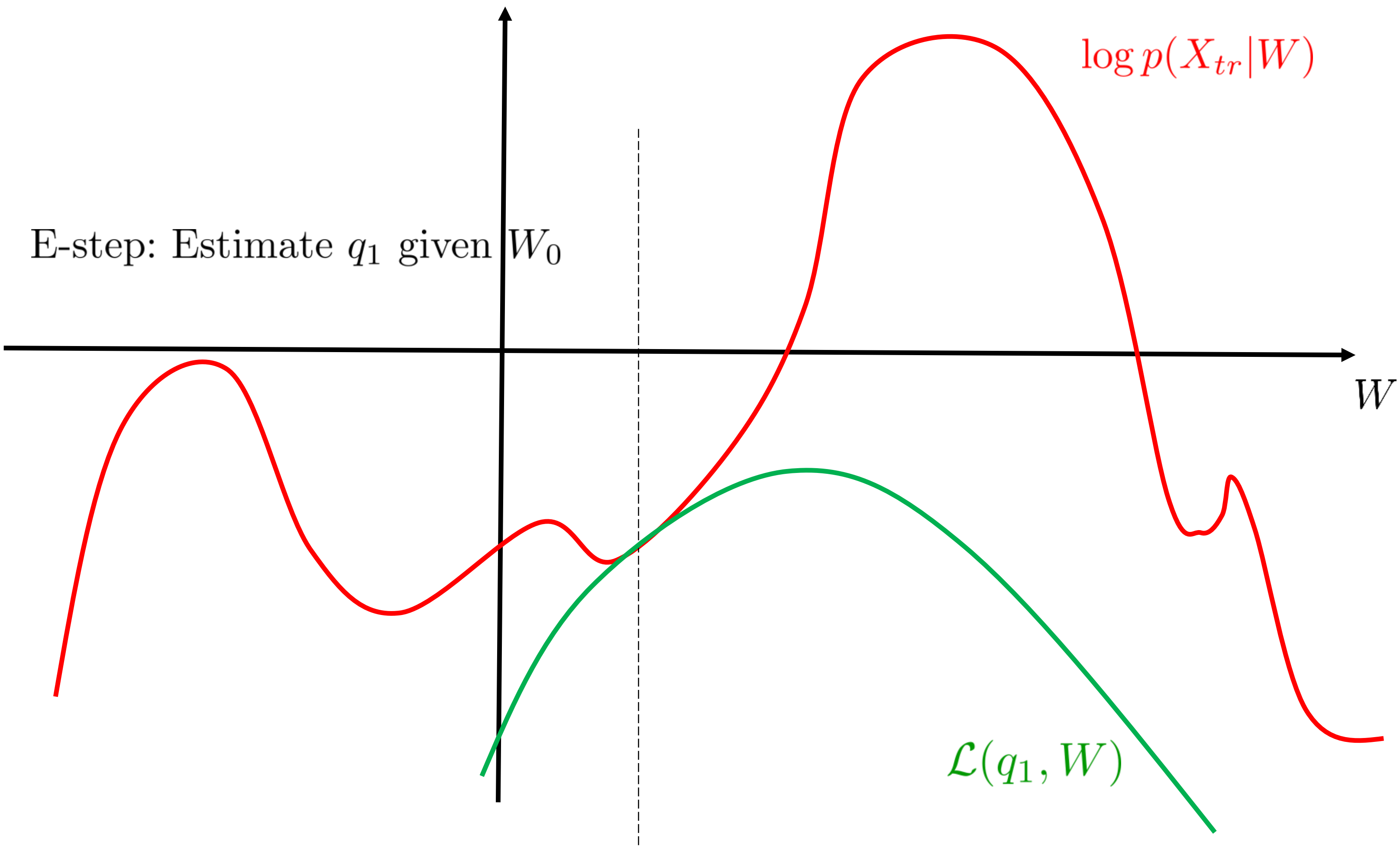
EM-algorithm



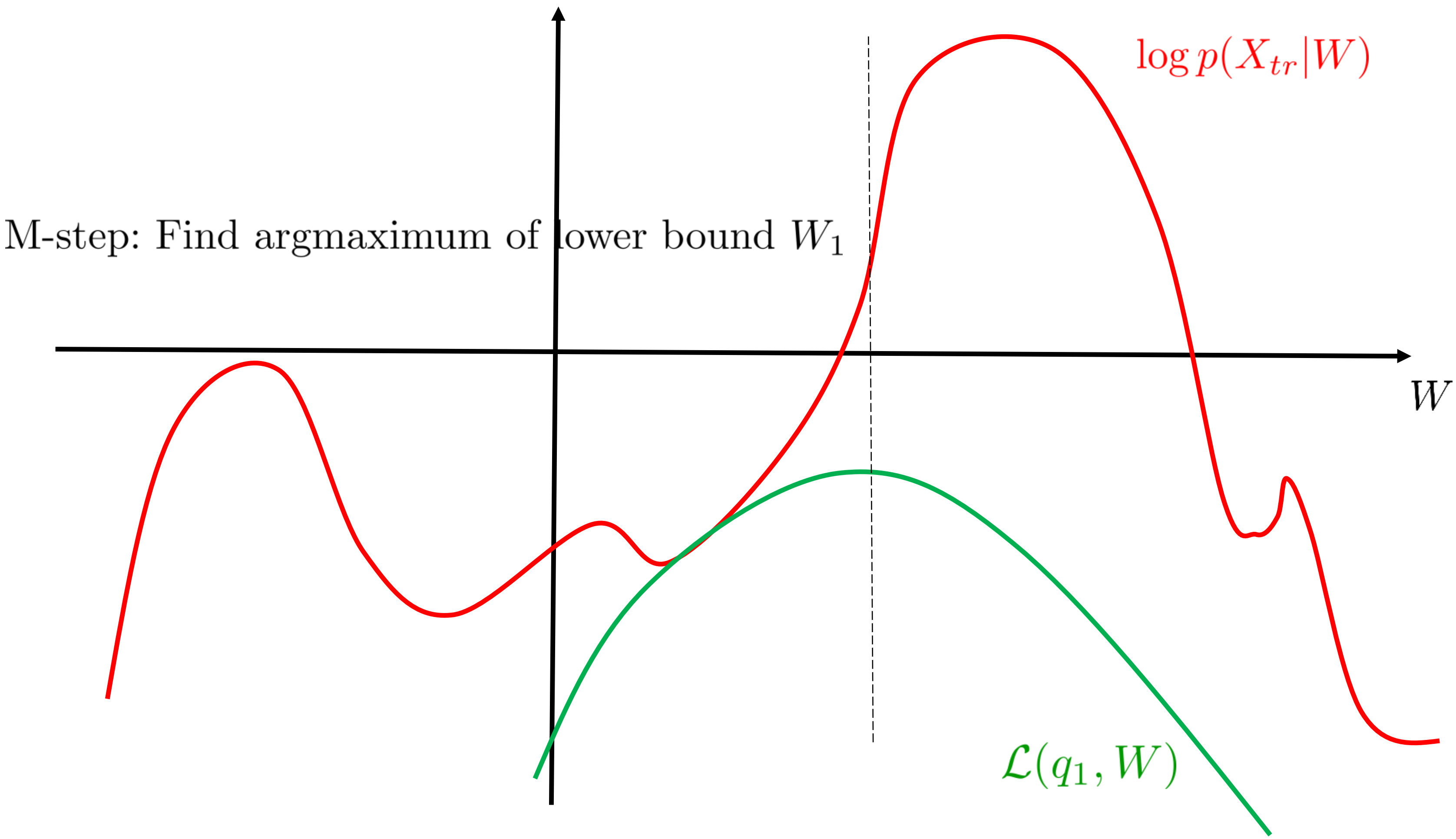
EM-algorithm



EM-algorithm

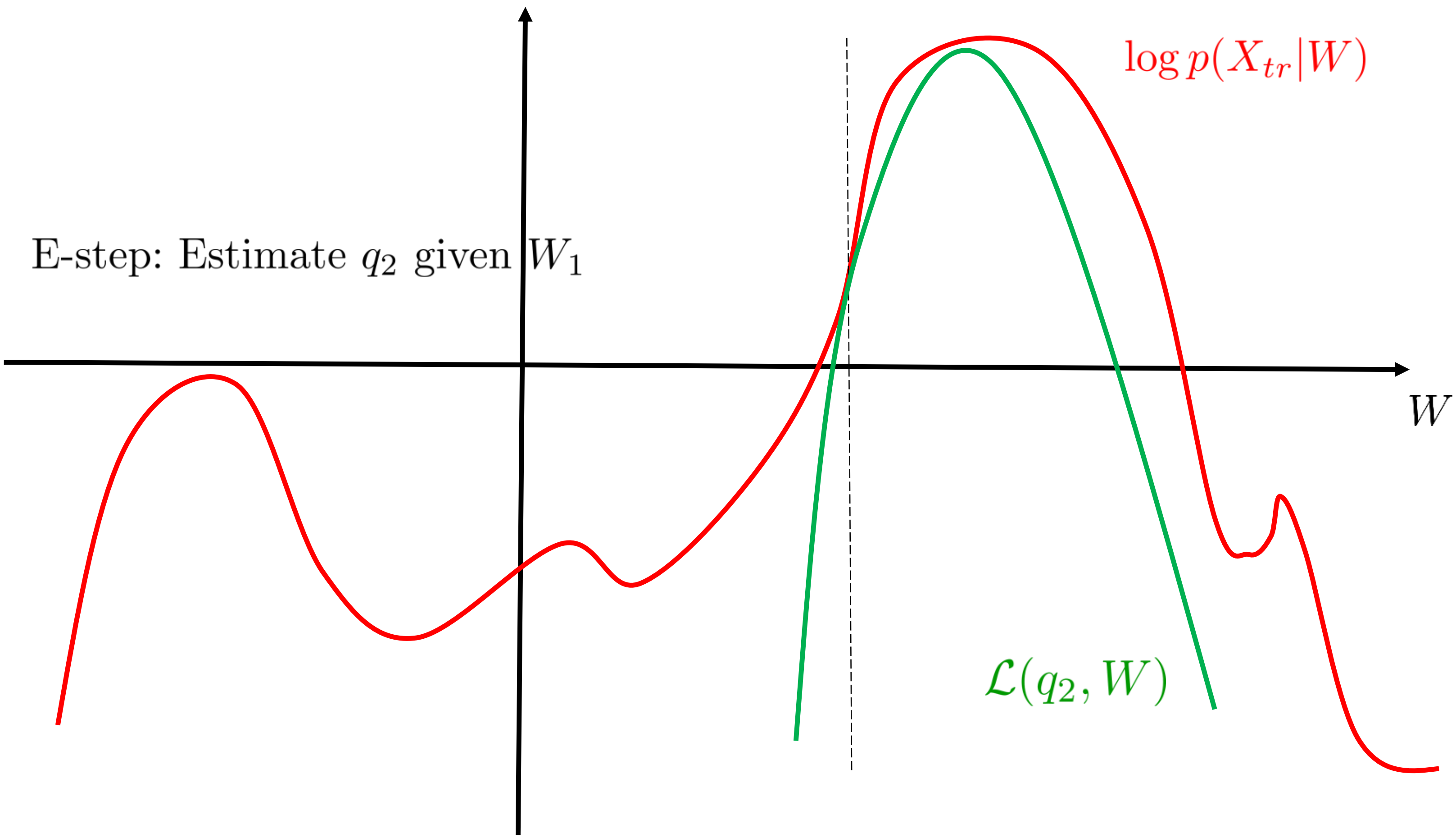


EM-algorithm



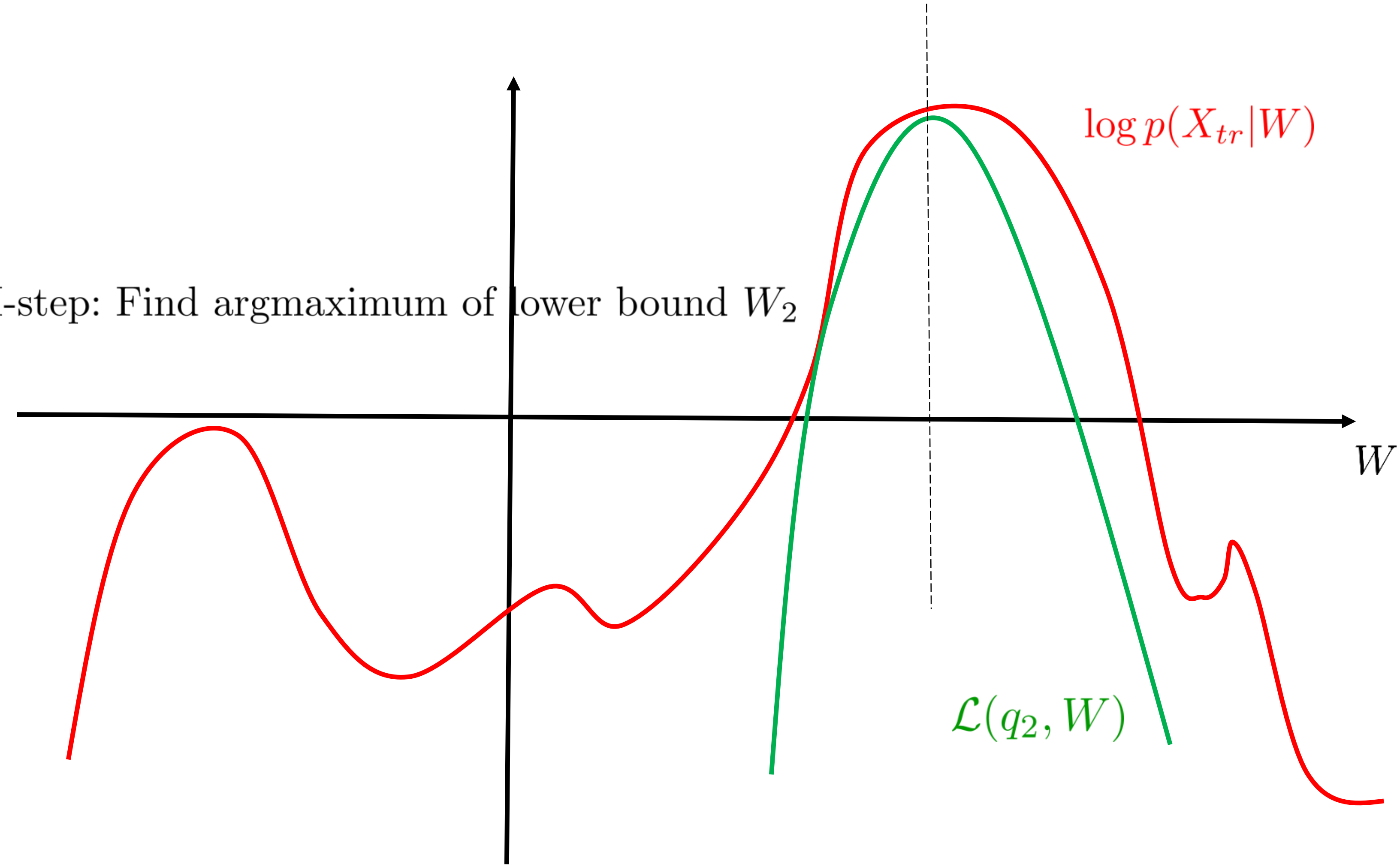
EM-algorithm

E-step: Estimate q_2 given W_1



EM-algorithm

M-step: Find argmaximum of lower bound W_2

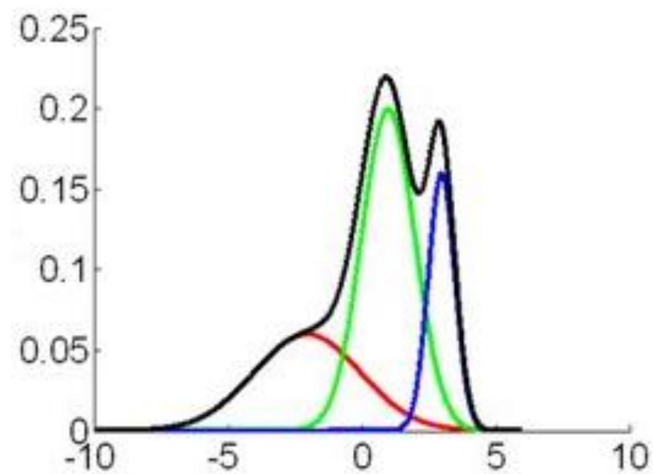


Discrete T

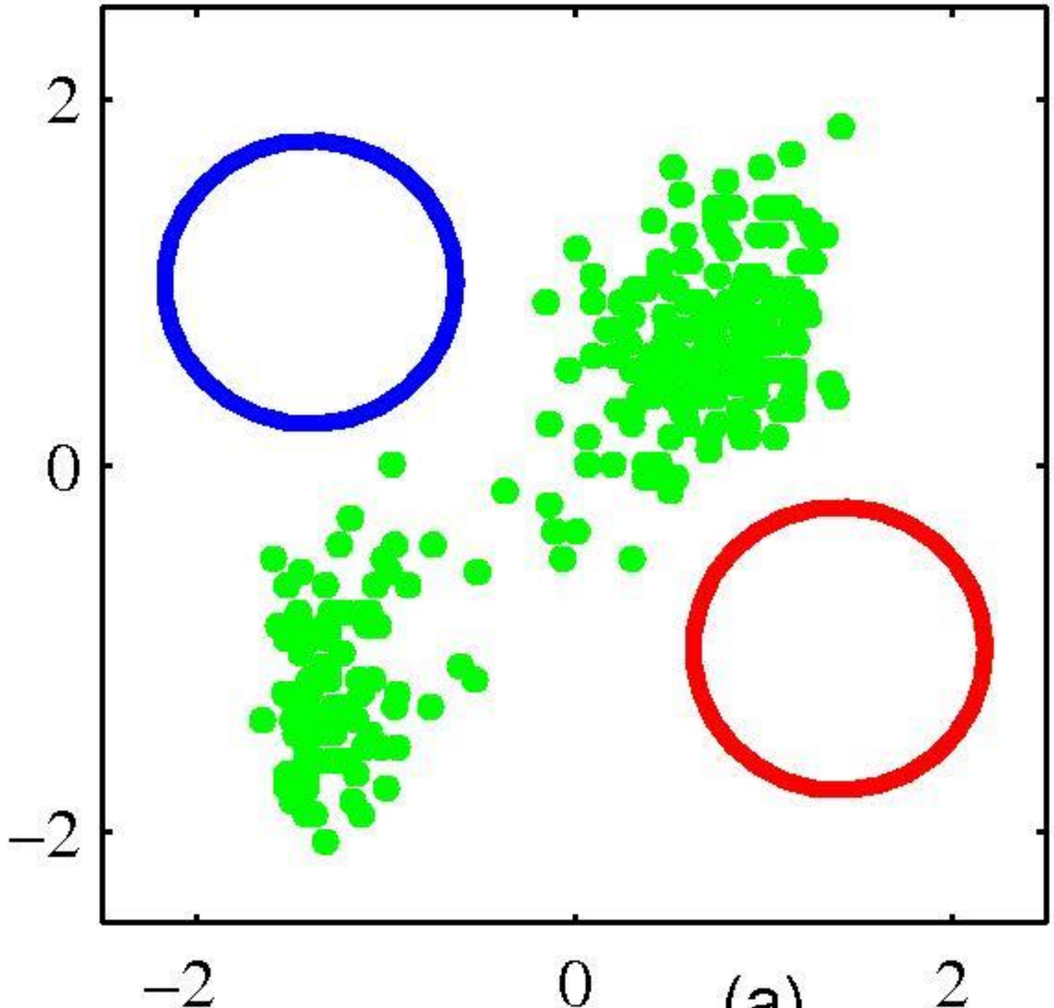
- Let $t \in \{1, \dots, K\}$, then

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

- 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

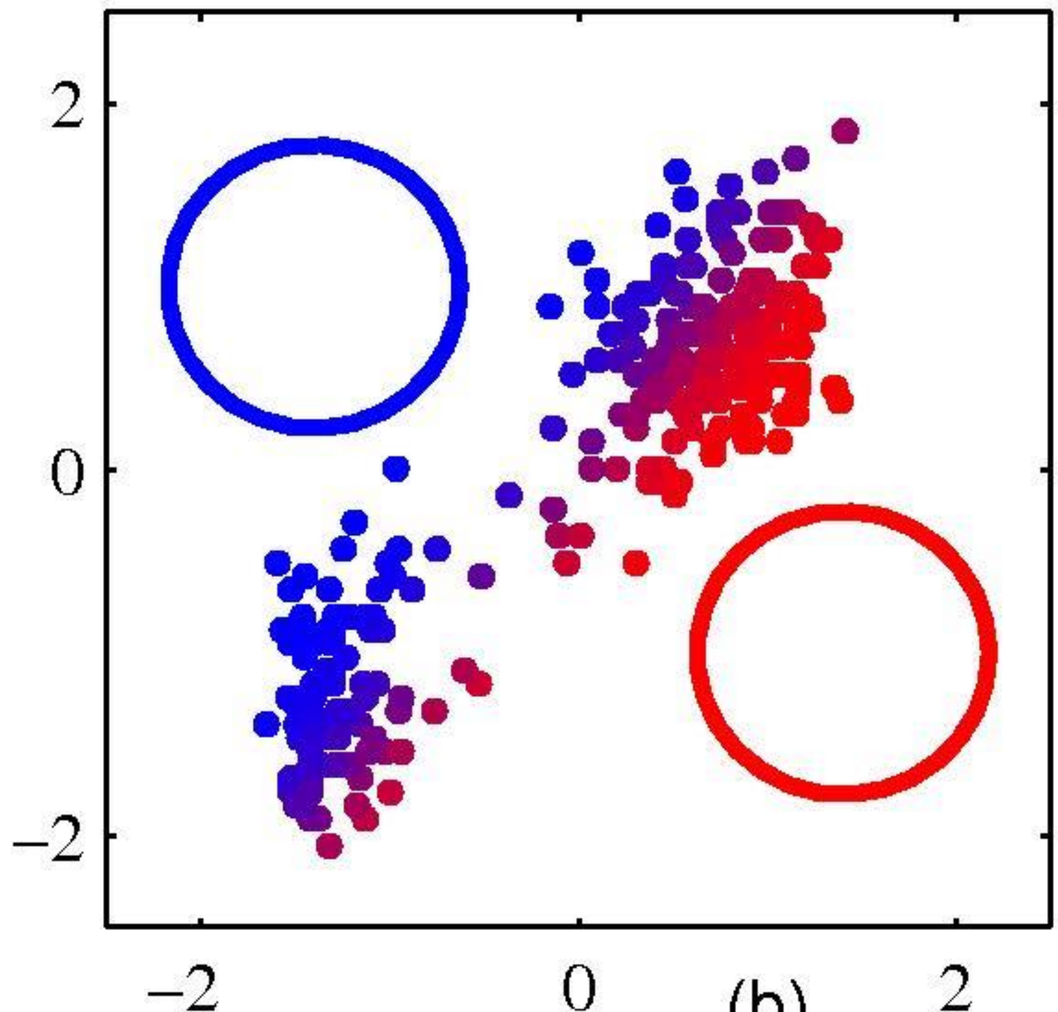


Mixture of gaussians



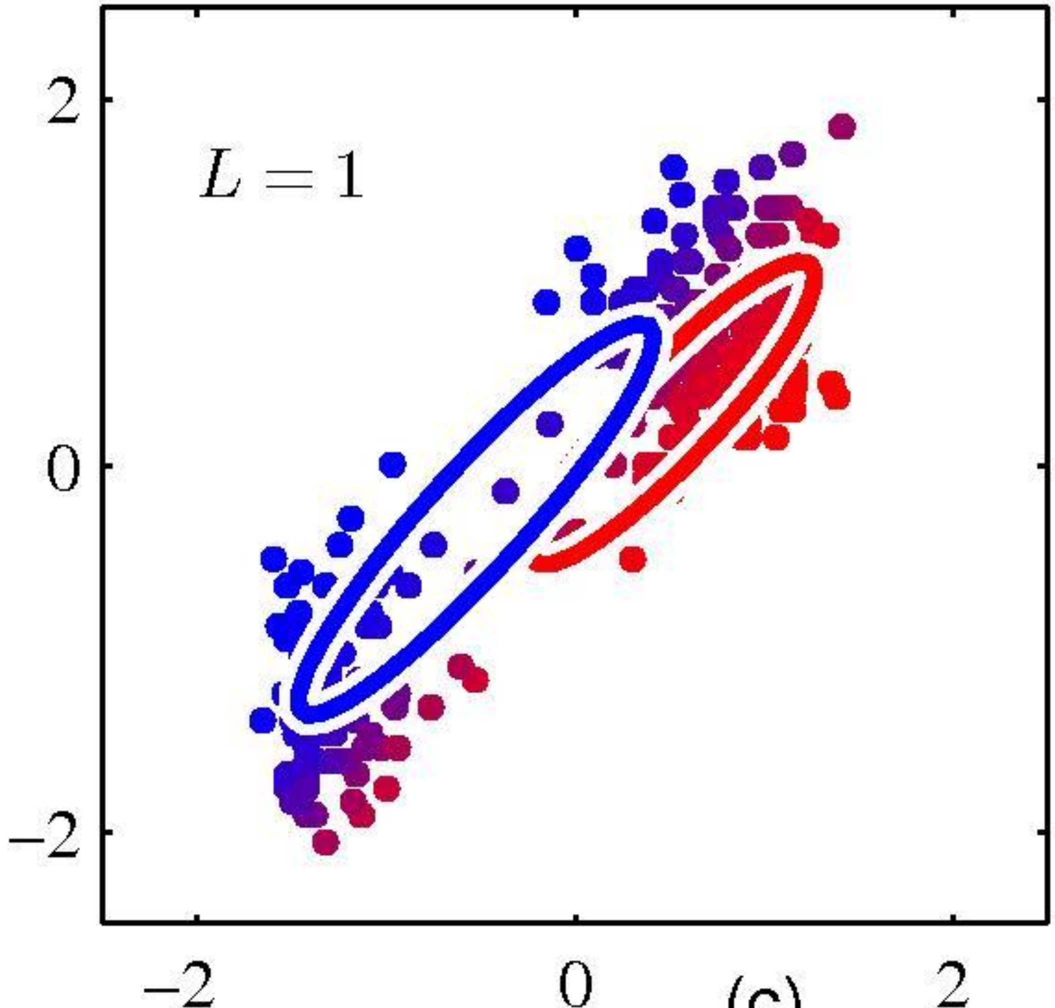
(a)

Mixture of gaussians



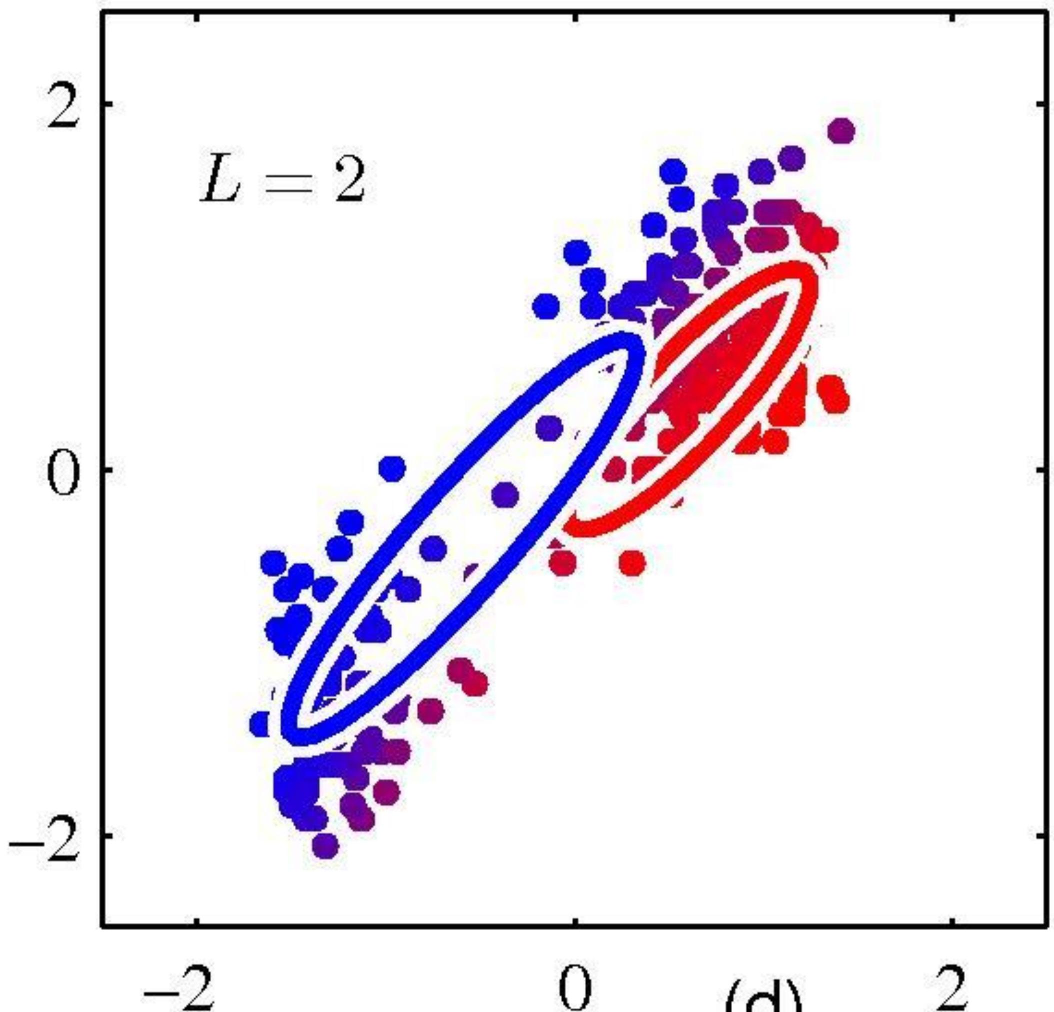
(b)

Mixture of gaussians



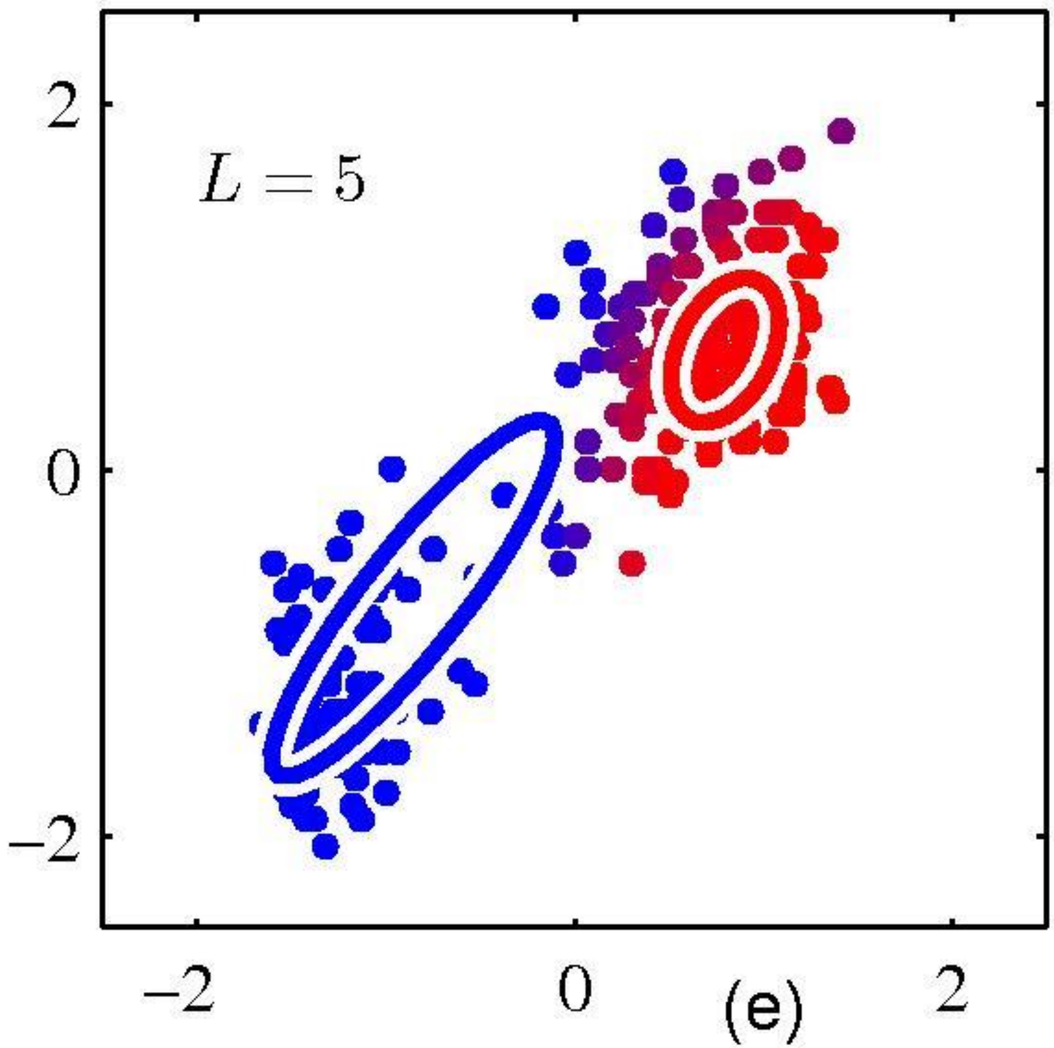
(c)

Mixture of gaussians

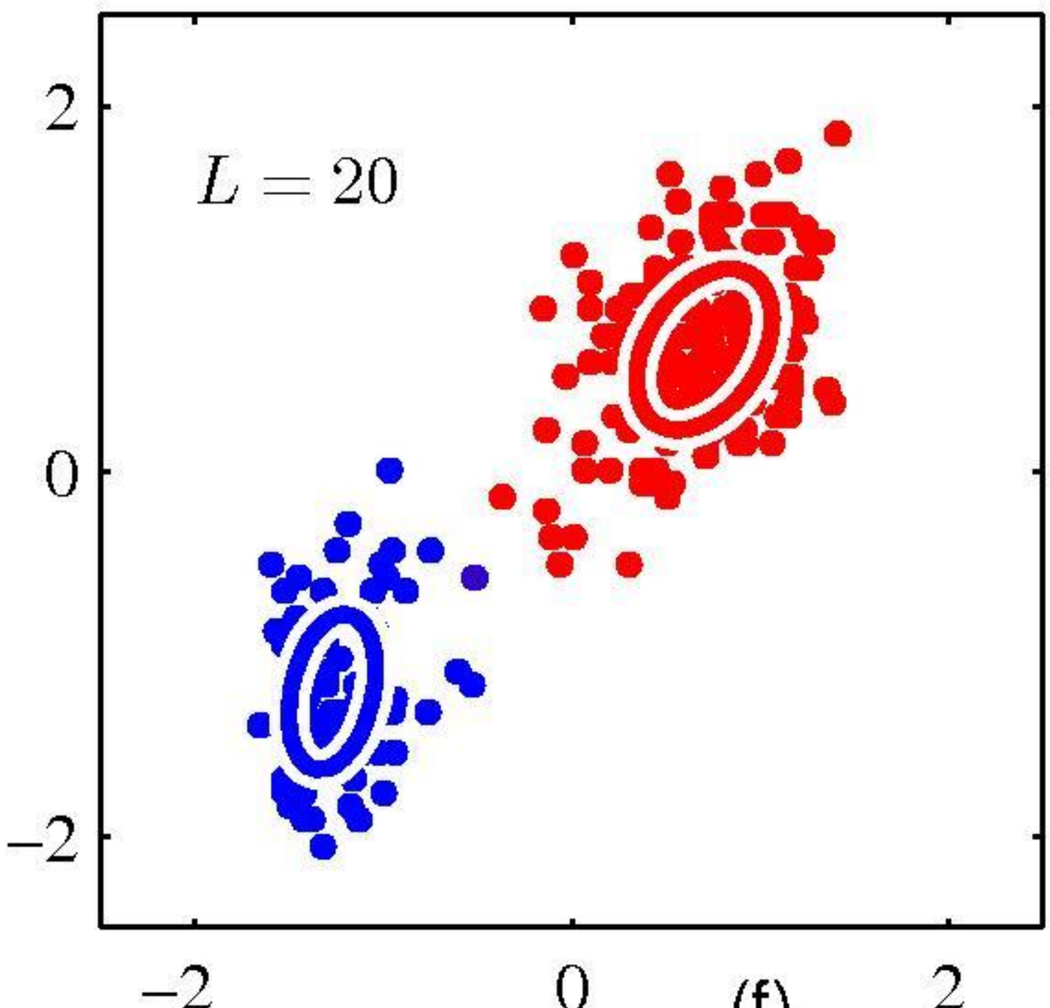


(d)

Mixture of gaussians



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

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

- M-step

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

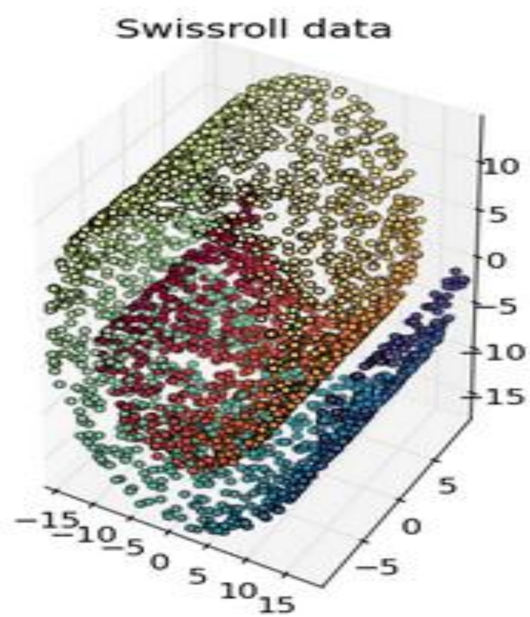
$$\Sigma_k = \frac{1}{n_k - 1}\sum_{i=1}^{n}(x_i - \mu_k)(x_i - \mu_k)^T$$

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

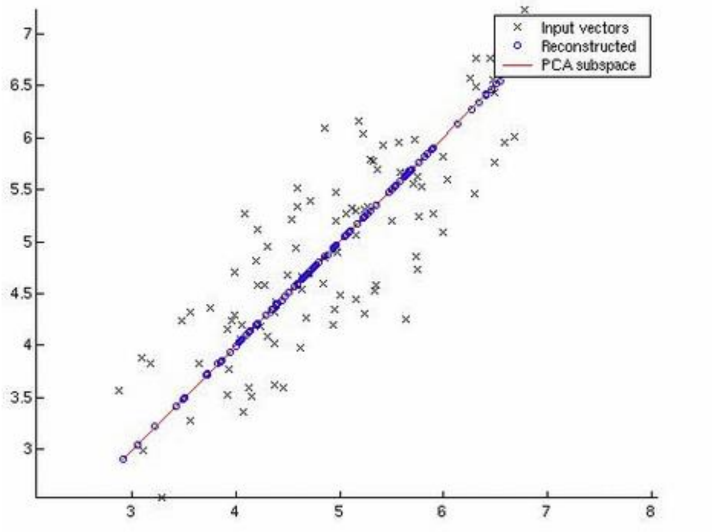


Example: PCA model

- 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?..

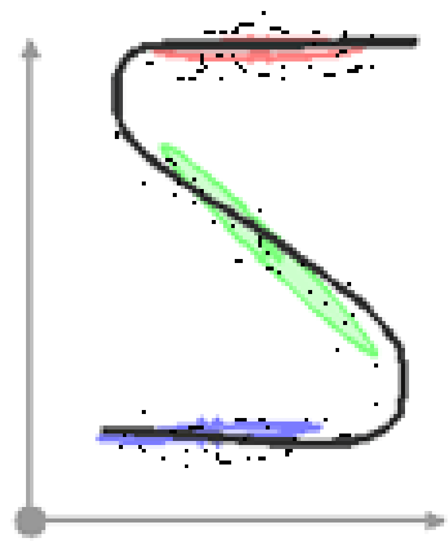
- EM updates have complexity $O(nDd)$ instead of $O(nD^2)$ in analytic solution
- Can process missing parts in X and present parts in T
- Can determinate d if $p(W)$ is established
- Can be extended to more general models such as mixture of PCA

Mixture of PCA

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

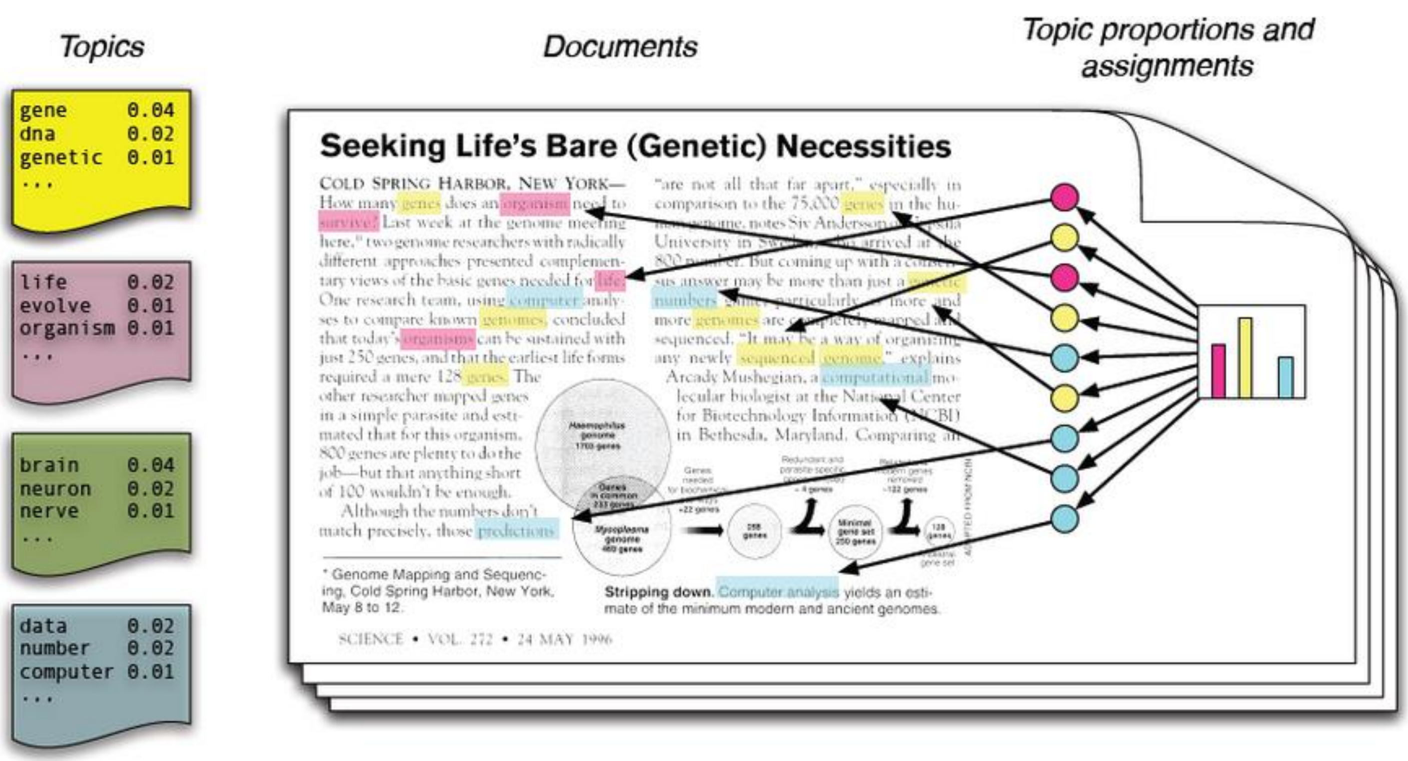
$$p(X, Z, T|W) = \prod_{i=1}^{n} p(x_i|t_i, z_i, W)p(t_i|W)p(z_i|W) = \prod_{i=1}^{n} \mathcal{N}(x_i|V_{z_i}t_i, \sigma_{z_i}^2 I)\mathcal{N}(t_i|0, I)\theta_{z_i}$$

- 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

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

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

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

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

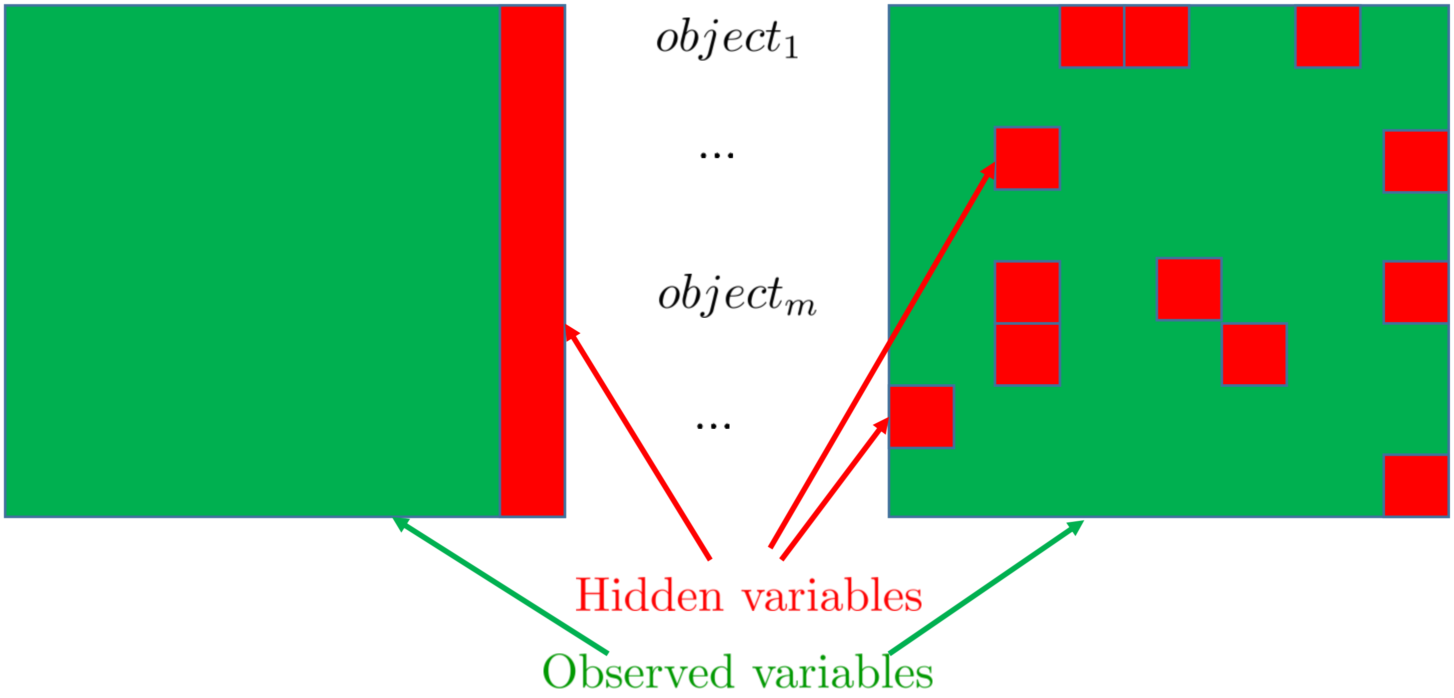
$$p(x_{di}|\psi_{z_d i}) = \psi_{z_{di},x_{di}} \quad \text{Probability of word } w_{di} \text{ belongs to topic } z_{di}$$

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

$$\text{Required: } p(\Psi|X) \to \max_{\Psi}$$

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

Extending E-step

- E-step requires conjugate distributions to be performed analytically
- Otherwise normalization constant cannot be computed

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

- Recall that

$$p(T|X_{tr}, W) = \arg\max_q \mathcal{L}(q, W) = \arg\min_q KL(q(T)||p(T|X_{tr}, W)),$$

  where extremum is taken with respect to **all possible distributions** $q(T)$

- What if we limit ourselves with more restricted set of distributions?..

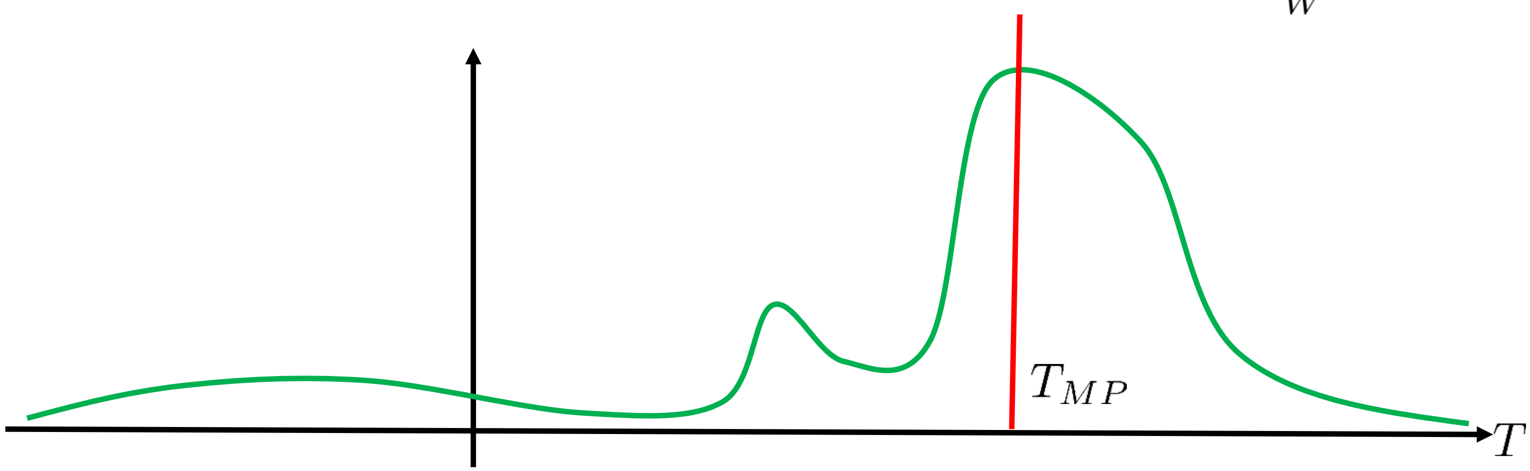
Crisp E-step

- Let's consider the family of δ-functions as a possible distributions $q(T)$
- It corresponds to point estimates for T
- It is easy to show that

$$\delta(W - W_{MP}) = \arg\min_{q() \in \Delta} KL(q(T) || p(T|X_{tr}, W))$$

- Note that M-step is then also simplified

$$\mathbb{E}_T \log p(X_{tr}, T|W) = \log p(X_{tr}, T_{MP}|W) \to \max_W$$

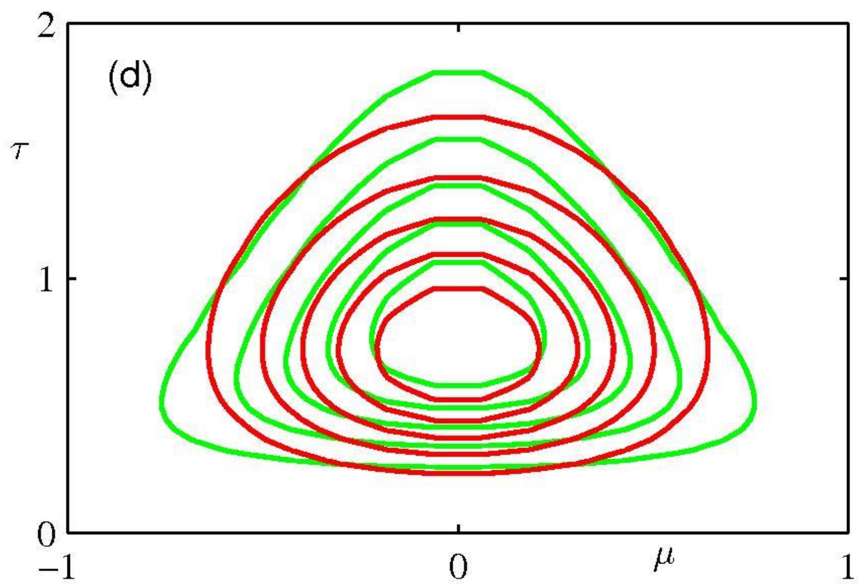


Variational E-step

- Let's consider the family of factorized distributions $q(T) = \prod_{j=1}^{k} q_j(t_j)$ as a possible distributions $q(T)$
- It is easy to get iterative re-estimation equations

$$\log q_j(t_j) = \mathbb{E}_{T \setminus t_j} \log p(X, T|W) + \text{Const}$$

- In the case of so-called block-conjugacy the expectation is computed analyticaly



Stochastic optimization

- New framework for working with big data
- Approximate super-fast optimization technique
- Allows to optimize function faster than the time needed to compute it in any given point
- Consider a function that is a sum of $N \gg 1$ items taken from the same distribution

$$F(\alpha) = \sum_{i=1}^{N} f(x_i, \alpha), \quad x_i \sim p(x)$$

- Then $N \nabla f(x_i, \alpha)$ is an unbiased estimate of $\nabla F(\alpha)$
- We may take **stochastic gradient** step

$$\alpha_{n+1} = \alpha_n + \varepsilon_n N \nabla f(x_i, \alpha)$$

- Under certain conditions such process converges to local maximum

Stochastic EM

- Consider huge sample of i.i.d. objects with observed and hidden variables $(X_{tr}, T) = (\{x_i\}_{i=1}^N, \{t_i\}_{i=1}^N)$
- Apply stochastic gradient step as M-step

$$W_{n+1} = W_n + \varepsilon_n N \mathbb{E}_T \nabla \log p(x_i, t_i | W)$$

- Then there is no need to computer anything except $q(t_i)$
- E-step becomes N times faster
- Orders of magnitude more efficient distributions of resourses!
- We may perform double stochastic scheme by removing $q(t_i)$ with a sample generated from $p(t_i | X_{tr}, W_n)$

Summary: extensions of basic EM

Extending E-step

- Crisp E-step: MAP estimate of T - no need to compute normalization constant
- Variational E-step: factorized approximation of $p(T|X_{tr}, W)$ - normalization constant may become tractable
- Monte Carlo E-step: provides with unbiased estimate of $p(T|X_{tr}, W)$

Extending M-step

- Early stop M-step: do not find $\arg\max \mathbb{E}_T \log p(X_{tr}, T|W)$ but improve W value
- Stochastic M-step: make stochastic subgradient step w.r.t. to only one object (or mini-batch)

Conclusion

- In the age of big data many data do not contain full labeling so there are lots of missing data
- The introduction of latent variables often allows to simplify the model
- We may enrich the model with prior knowledge (or preferences) about hidden variables by establishing $p(T)$ and/or $p(W)$
- The understanding of general idea of EM-algorithm allows one to invent numerious extensions without sacrificing the correctness of EM-framework

Challenge

For those who's interested

- Help Nick Carter to find the criminal who kidnapped lady Thun's dog http://cmp.felk.cvut.cz/cmp/courses/recognition/Labs/em/index_en.html