FIAS Frankfurt Institute for Advanced Studies

GOETHE UNIVERSITÄT FRANKFURT AM MAIN

www.goethe-universitaet.de

Machine Learning II: (Graphical Models, EM, Variational Inference)

Prepared by:
Prof. Dr. Visvanathan Ramesh

References and Sources: Nils Bertschinger (ML II lecture slides) Simon Prince (Learning and Vision), HMM Topology Selection (Stenger et al), Variational Inference (D. Blei)

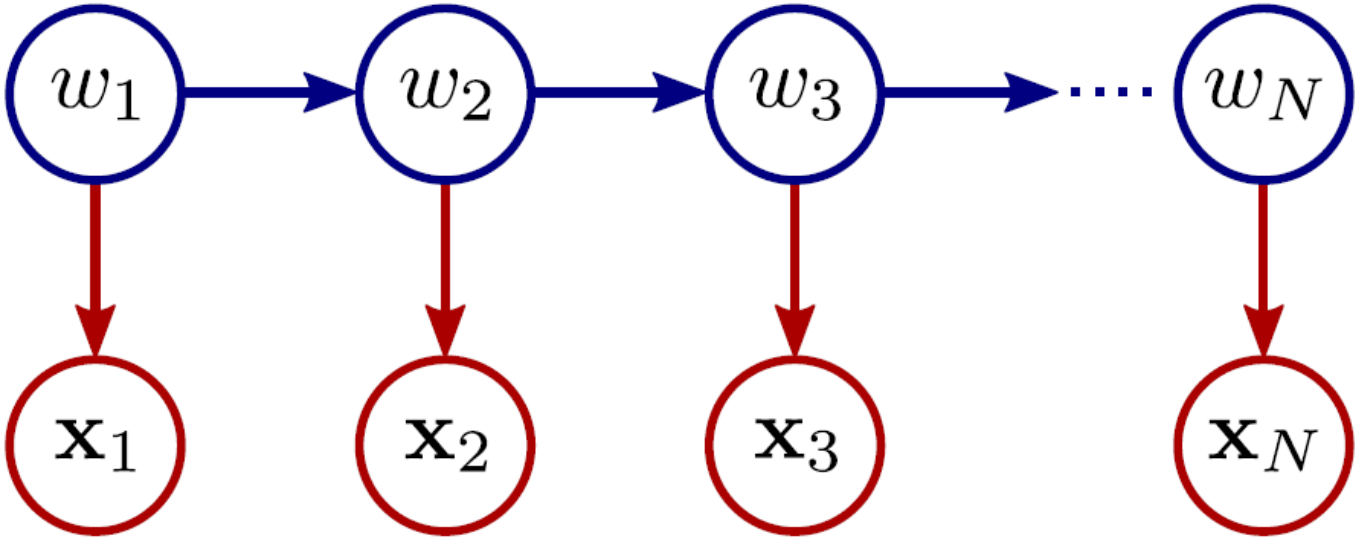
Outline

- **Recap – Past Lectures (Based on Simon Prince, Chapters 6, 7, 9, 10, 11)**
  - Regression, Classification, Application examples in Vision
  - Graphical Models and Inference
    - Graphical Models (directed, undirected)
    - Models for Chains and Trees

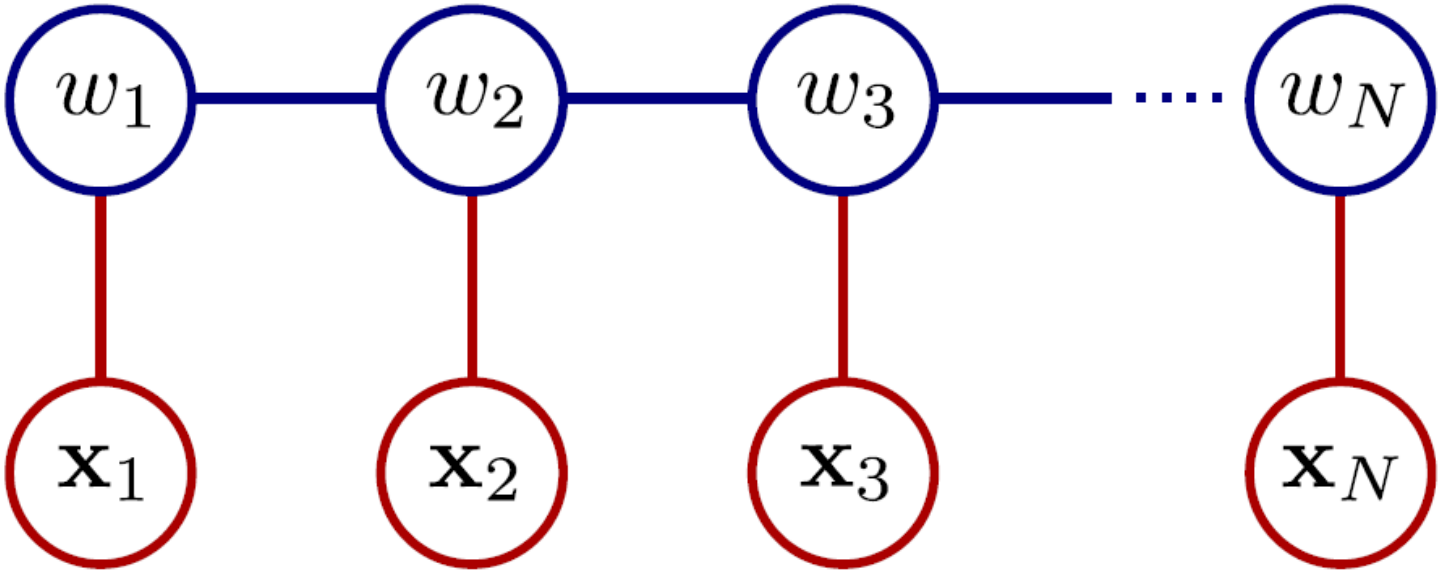
  - Today's Lecture
    - Expectation Maximization Algorithm
    - HMM Model selection with application in Vision
    - Variational Bayes

Brief Recap

Directed model for chains (Hidden Markov model)



Undirected model for chains



Learning in chains and trees

Supervised learning (where we know world states w_n) - relatively easy.

Unsupervised learning (where we do not know world states w_n) is more challenging. Use the EM algorithm:

- E-step – compute posterior marginals over states
- M-step – update model parameters

For the chain model (hidden Markov model) this is known as the Baum-Welch algorithm.

FIAS Frankfurt Institute for Advanced Studies

GOETHE UNIVERSITÄT FRANKFURT AM MAIN

EM Algorithm

Outline

- Expectation Maximation Algorithm (Background)
- Convexity
- Jensen’s Inequality
- EM Algorithm Formulation
- Short outline of Proofs
- Summary

Convex Functions

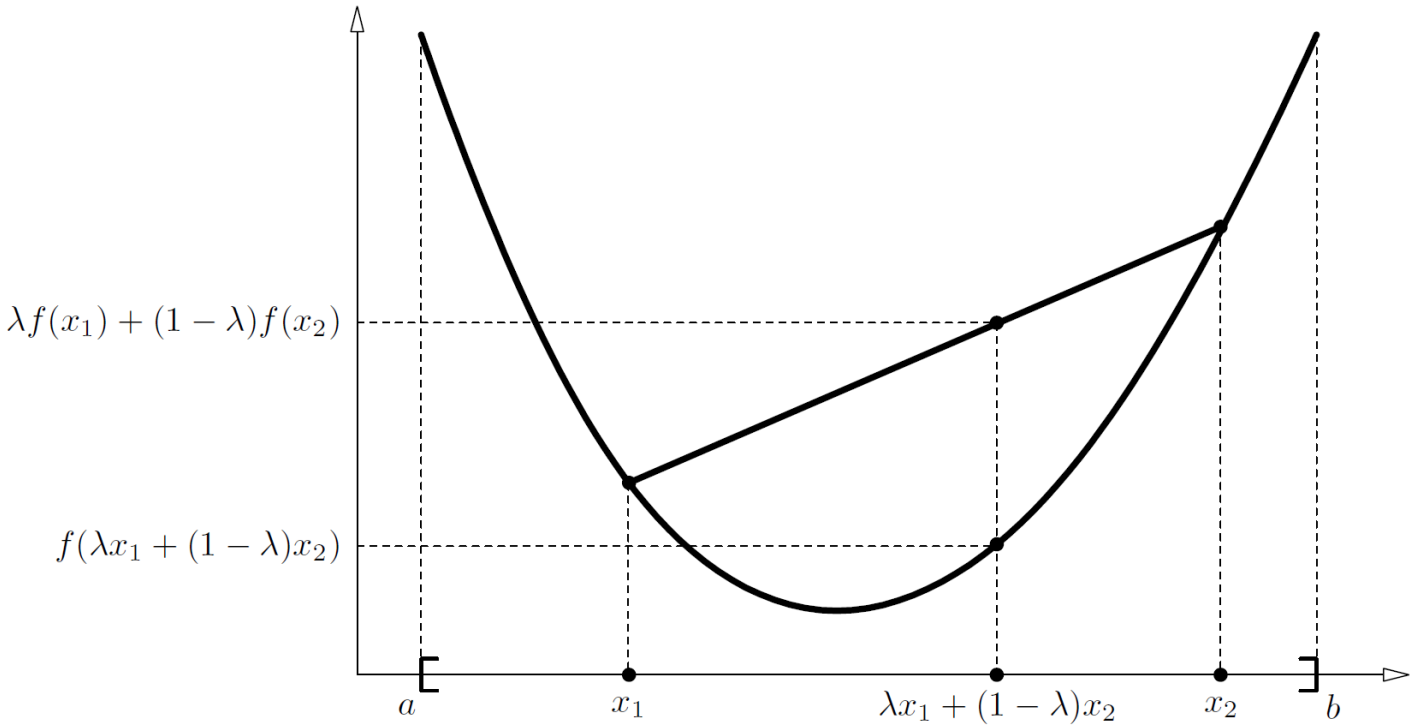


Figure 1: f is *convex* on $[a, b]$ if $f(\lambda x_1 + (1-\lambda)x_2) \leq \lambda f(x_1) + (1-\lambda) f(x_2)$ $\forall x_1, x_2 \in [a, b], \quad \lambda \in [0, 1]$.

Definitions

Definition 1 *Let f be a real valued function defined on an interval $I = [a, b]$. f is said to be* convex *on I if $\forall x_1, x_2 \in I, \lambda \in [0, 1]$,*

$$f(\lambda x_1 + (1 - \lambda)x_2) \leq \lambda f(x_1) + (1 - \lambda)f(x_2).$$

f is said to be strictly convex *if the inequality is strict. Intuitively, this definition states that the function falls below (strictly convex) or is never above (convex) the straight line (the secant) from points $(x_1, f(x_1))$ to $(x_2, f(x_2))$. See Figure (1).*

Definition 2 *f is concave (strictly concave) if $-f$ is convex (strictly convex).*

Theorem 1 *If $f(x)$ is twice differentiable on $[a, b]$ and $f''(x) \geq 0$ on $[a, b]$ then $f(x)$ is convex on $[a, b]$.*

)

Jensen's Inequality

Theorem 2 (Jensen's inequality) *Let f be a convex function defined on an interval I. If $x_1, x_2, \ldots, x_n \in I$ and $\lambda_1, \lambda_2, \ldots, \lambda_n \geq 0$ with $\sum_{i=1}^{n} \lambda_i = 1$,*

$$f\left(\sum_{i=1}^{n} \lambda_i x_i\right) \leq \sum_{i=1}^{n} \lambda_i f(x_i)$$

- Proof follows by Induction (Trivial for n = 1, n=2 → follows from convexity, demonstrate for n+1 assuming theorem true for n).

Since $\ln(x)$ is concave, we may apply Jensen's inequality to obtain the useful result,

$$\ln \sum_{i=1}^{n} \lambda_i x_i \geq \sum_{i=1}^{n} \lambda_i \ln(x_i). \qquad (6)$$

This allows us to lower-bound a logarithm of a sum, a result that is used in the derivation of the EM algorithm.

EM Algorithm Overview

Let $\mathbf{X}$ be random vector which results from a parameterized family. We wish to find θ such that $\mathcal{P}(\mathbf{X}|\theta)$ is a maximum. This is known as the Maximum Likelihood (ML) estimate for θ. In order to estimate θ, it is typical to introduce the *log likelihood function* defined as,

$$L(\theta) = \ln \mathcal{P}(\mathbf{X}|\theta). \tag{7}$$

The likelihood function is considered to be a function of the parameter θ given the data $\mathbf{X}$. Since $\ln(x)$ is a strictly increasing function, the value of θ which maximizes $\mathcal{P}(\mathbf{X}|\theta)$ also maximizes $L(\theta)$.

The EM algorithm is an iterative procedure for maximizing $L(\theta)$. Assume that after the n^{th} iteration the current estimate for θ is given by θ_n. Since the objective is to maximize $L(\theta)$, we wish to compute an updated estimate θ such that,

$$L(\theta) > L(\theta_n) \tag{8}$$

Equivalently we want to maximize the difference,

$$L(\theta) - L(\theta_n) = \ln \mathcal{P}(\mathbf{X}|\theta) - \ln \mathcal{P}(\mathbf{X}|\theta_n). \tag{9}$$

(12)

EM Algorithm (Derivation)

$$L(\theta) - L(\theta_n) = \ln\left(\sum_{\mathbf{z}} \mathcal{P}(\mathbf{X}|\mathbf{z},\theta)\mathcal{P}(\mathbf{z}|\theta)\right) - \ln \mathcal{P}(\mathbf{X}|\theta_n). \tag{11}$$

$$\begin{aligned}
L(\theta) - L(\theta_n) &= \ln\left(\sum_{\mathbf{z}} \mathcal{P}(\mathbf{X}|\mathbf{z},\theta)\mathcal{P}(\mathbf{z}|\theta)\right) - \ln \mathcal{P}(\mathbf{X}|\theta_n) \\
&= \ln\left(\sum_{\mathbf{z}} \mathcal{P}(\mathbf{X}|\mathbf{z},\theta)\mathcal{P}(\mathbf{z}|\theta) \cdot \frac{\mathcal{P}(\mathbf{z}|\mathbf{X},\theta_n)}{\mathcal{P}(\mathbf{z}|\mathbf{X},\theta_n)}\right) - \ln \mathcal{P}(\mathbf{X}|\theta_n) \\
&= \ln\left(\sum_{\mathbf{z}} \mathcal{P}(\mathbf{z}|\mathbf{X},\theta_n)\frac{\mathcal{P}(\mathbf{X}|\mathbf{z},\theta)\mathcal{P}(\mathbf{z}|\theta)}{\mathcal{P}(\mathbf{z}|\mathbf{X},\theta_n)}\right) - \ln \mathcal{P}(\mathbf{X}|\theta_n) \\
&\geq \sum_{\mathbf{z}} \mathcal{P}(\mathbf{z}|\mathbf{X},\theta_n) \ln\left(\frac{\mathcal{P}(\mathbf{X}|\mathbf{z},\theta)\mathcal{P}(\mathbf{z}|\theta)}{\mathcal{P}(\mathbf{z}|\mathbf{X},\theta_n)}\right) - \ln \mathcal{P}(\mathbf{X}|\theta_n) \qquad (12) \\
&= \sum_{\mathbf{z}} \mathcal{P}(\mathbf{z}|\mathbf{X},\theta_n) \ln\left(\frac{\mathcal{P}(\mathbf{X}|\mathbf{z},\theta)\mathcal{P}(\mathbf{z}|\theta)}{\mathcal{P}(\mathbf{z}|\mathbf{X},\theta_n)\mathcal{P}(\mathbf{X}|\theta_n)}\right) \qquad (13) \\
&\triangleq \Delta(\theta|\theta_n). \qquad (14)
\end{aligned}$$

$$l(\theta|\theta_n) \triangleq L(\theta_n) + \Delta(\theta|\theta_n)$$

EM Algorithm (Continued)

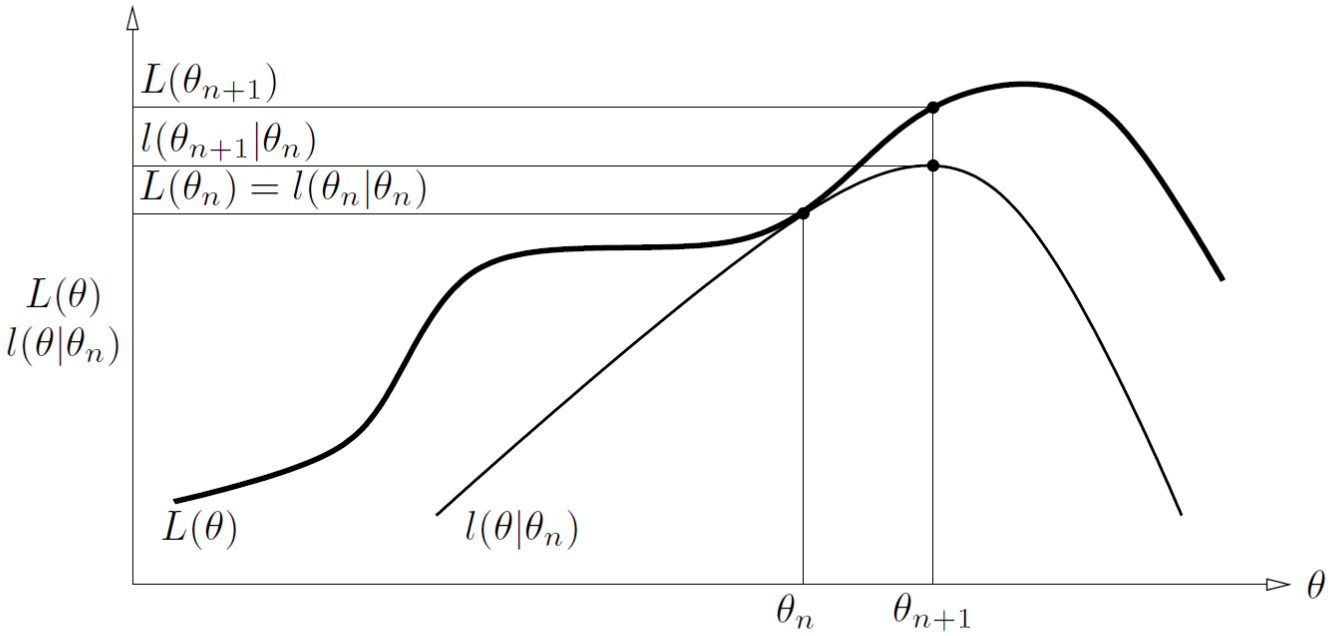


Figure 2: Graphical interpretation of a single iteration of the EM algorithm: The function $l(\theta|\theta_n)$ is bounded above by the likelihood function $L(\theta)$. The functions are equal at $\theta = \theta_n$. The EM algorithm chooses θ_{n+1} as the value of θ for which $l(\theta|\theta_n)$ is a maximum. Since $L(\theta) \geq l(\theta|\theta_n)$ increasing $l(\theta|\theta_n)$ ensures that the value of the likelihood function $L(\theta)$ is increased at each step.

EM Algorithm (Derivation)

$$
\begin{aligned}
\theta_{n+1} &= \arg\max_{\theta} \{l(\theta|\theta_n)\} \\
&= \arg\max_{\theta} \left\{ L(\theta_n) + \sum_{\mathbf{z}} \mathcal{P}(\mathbf{z}|\mathbf{X},\theta_n) \ln \frac{\mathcal{P}(\mathbf{X}|\mathbf{z},\theta)\mathcal{P}(\mathbf{z}|\theta)}{\mathcal{P}(\mathbf{X}|\theta_n)\mathcal{P}(\mathbf{z}|\mathbf{X},\theta_n)} \right\} \\
&\quad \text{Now drop terms which are constant w.r.t. } \theta \\
&= \arg\max_{\theta} \left\{ \sum_{\mathbf{z}} \mathcal{P}(\mathbf{z}|\mathbf{X},\theta_n) \ln \mathcal{P}(\mathbf{X}|\mathbf{z},\theta)\mathcal{P}(\mathbf{z}|\theta) \right\} \\
&= \arg\max_{\theta} \left\{ \sum_{\mathbf{z}} \mathcal{P}(\mathbf{z}|\mathbf{X},\theta_n) \ln \frac{\mathcal{P}(\mathbf{X},\mathbf{z},\theta)}{\mathcal{P}(\mathbf{z},\theta)} \frac{\mathcal{P}(\mathbf{z},\theta)}{\mathcal{P}(\theta)} \right\} \\
&= \arg\max_{\theta} \left\{ \sum_{\mathbf{z}} \mathcal{P}(\mathbf{z}|\mathbf{X},\theta_n) \ln \mathcal{P}(\mathbf{X},\mathbf{z}|\theta) \right\} \\
&= \arg\max_{\theta} \left\{ \mathrm{E}_{\mathbf{Z}|\mathbf{X},\theta_n} \left\{ \ln \mathcal{P}(\mathbf{X},\mathbf{z}|\theta) \right\} \right\} \qquad (17)
\end{aligned}
$$

EM Algorithm - Summary

1. *E-step*: Determine the conditional expectation $\mathrm{E}_{\mathbf{Z}|\mathbf{X},\theta_n}\{\ln \mathcal{P}(\mathbf{X},\mathbf{z}|\theta)\}$
2. *M-step*: Maximize this expression with respect to θ.

Key Points:

- Iteratively converges to a local maximum
- Detailed Proof done later demonstrates convergence may not be only to Maxima (e.g. saddle points)
- Method is a unified principle for a number of estimation problems with Hidden variables and/or missing data.
- Several methods followed addressing computational speedups of algorithm

FIAS Frankfurt Institute for Advanced Studies

GOETHE UNIVERSITÄT FRANKFURT AM MAIN

Adaptive Background Modeling Using a Hidden Markov Model

Slide Sources: Thesis by Bjoern Stenger (2000)

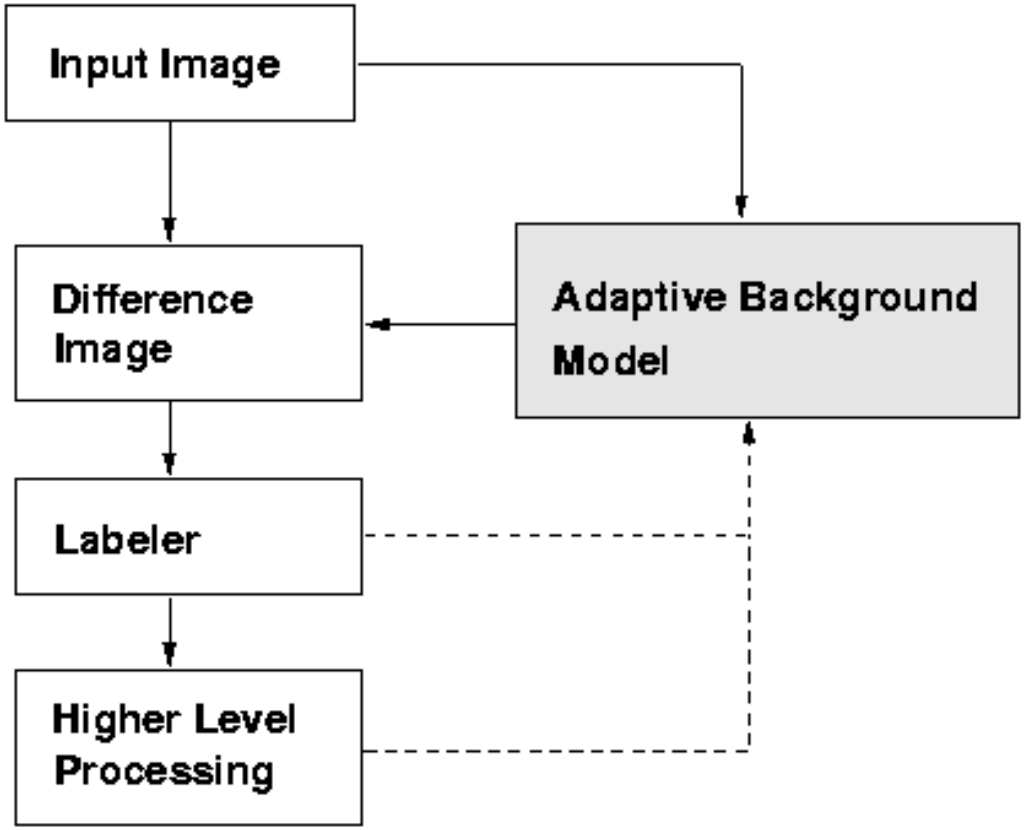
SCR Supervisor: Dr. Ramesh Visvanathan

Academic: Prof. J. Buhmann (Uni-Bonn)

Overview

- Problem of Background adaptation in Video Sequences
- Literature Review
- An “offline” HMM State-Splitting Algorithm
- An “online” version for adapting HMM parameters
- Example Background adaptation using HMMs
- Demonstration
- Summary

Module in Video Monitoring – Background Adaption



Background Adaption – Related Literature

Statistics Estimation

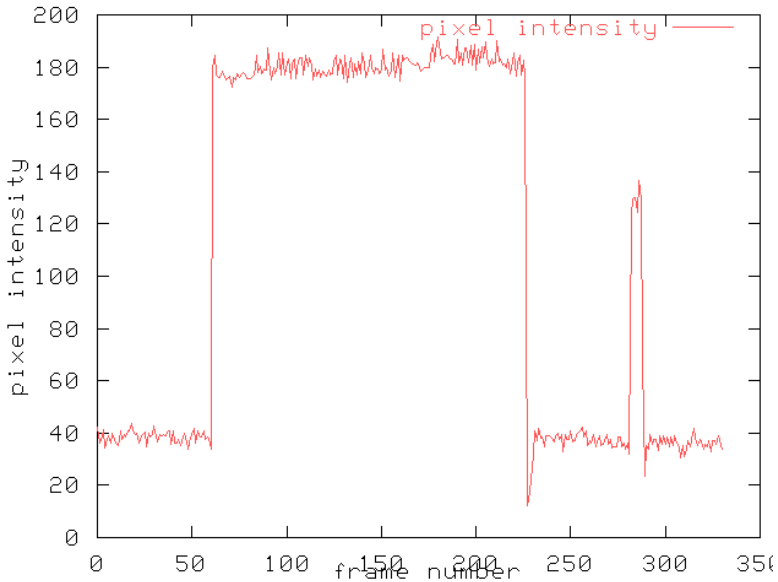
- Mean and variances per pixel

Dynamic adaptation

- Linear Prediction
- Kalman-Filter
- “Exponential Forgetting”

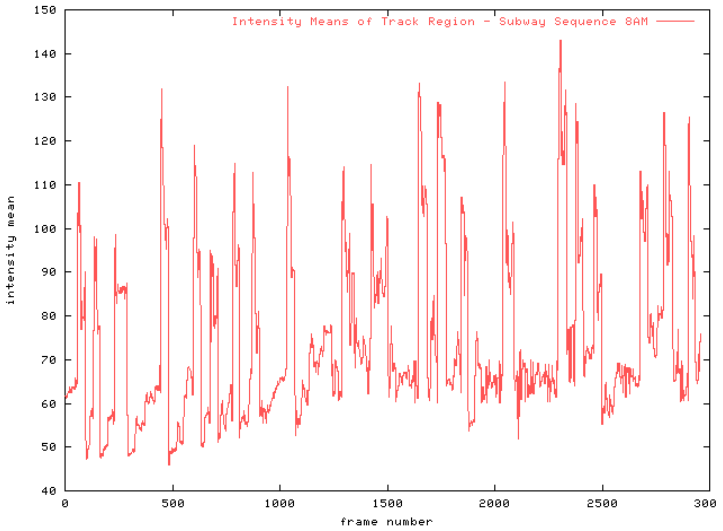
Problem: A single global model is not necessarily representative of the statistics. Need a mechanism to handle multiple states and to adapt the model online

Idea: Multi-State model for Global Changes

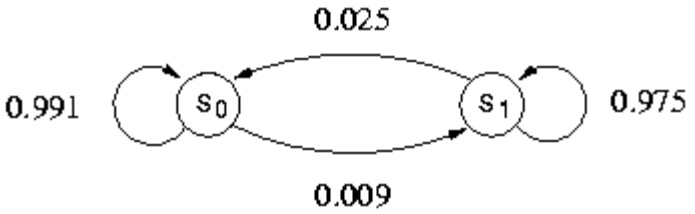


State Learning from Video Sequences

Example Subway Sequence Observations & HMM Learned



No Train in Station Train in Station



Relevance to Segmentation

Hidden Markov Model: Overview

Set of States *S* with Cardinality *N*, State at time index t denoted by s_t

Probability of being in a given state "i" at time t=1:

$$\pi(i) = P(s_t = i) \qquad i \in S.$$

Transition Probability Matrix

$$a_{ij} = P(s_{t+1} = j \mid s_t = i) \quad i,j \in S.$$

Given a hidden state s_t, the observations are assumed to be sampled

from the Alphabet *O*

$$b_i(k) = P(o_t = k \mid s_t = i) \qquad k \in O,\ i \in S$$

or from a continuous density:

$$b_i(x) = f_{\theta(i)}(x) \qquad i \in S.$$

Algorithms for HMMs

Generation of Sequence

Forward/Backward Algorithms

- **Useful to evaluate the probability $P(O \mid \theta)$**

Viterbi Algorithm

- **Used to find the optimal state sequence given the observations and parameters (I.e.) argmax $_S$ $P(S \mid O, \theta)$**

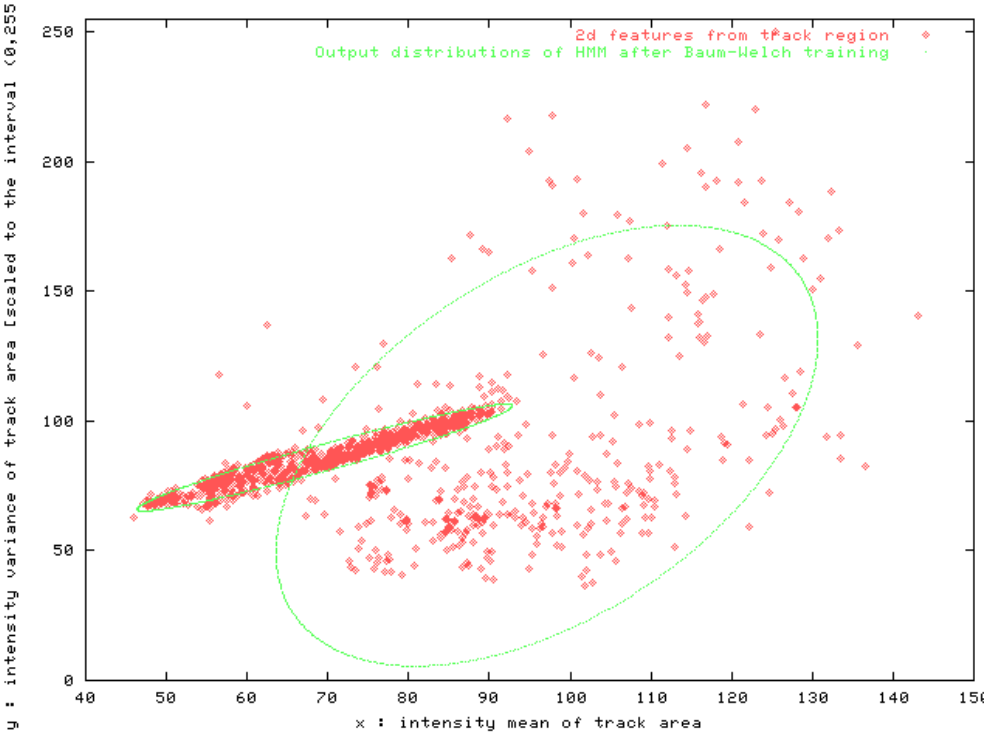
Baum-Welch Algorithm

- **To determine the maximum likelihood estimate (I.e) the theta that maximizes: $P (O \mid \theta)$**

State Learning from Video Sequences (offline)

- **Data: Subway monitoring sequence**

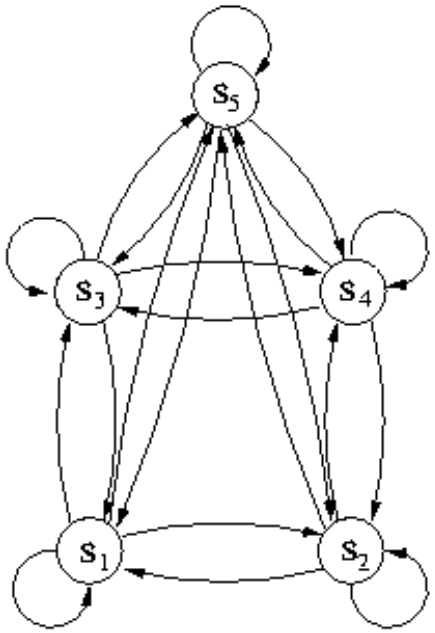
Number of hidden states assumed to be given apriori: Train in station/ not in station

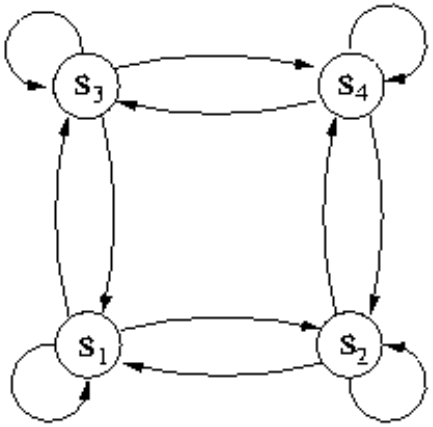


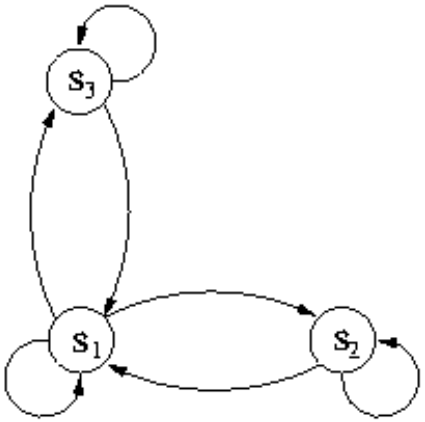
Correct Classification:

EM: 97.2%
HMM: 99.8%

State Splitting and Learning of Topology



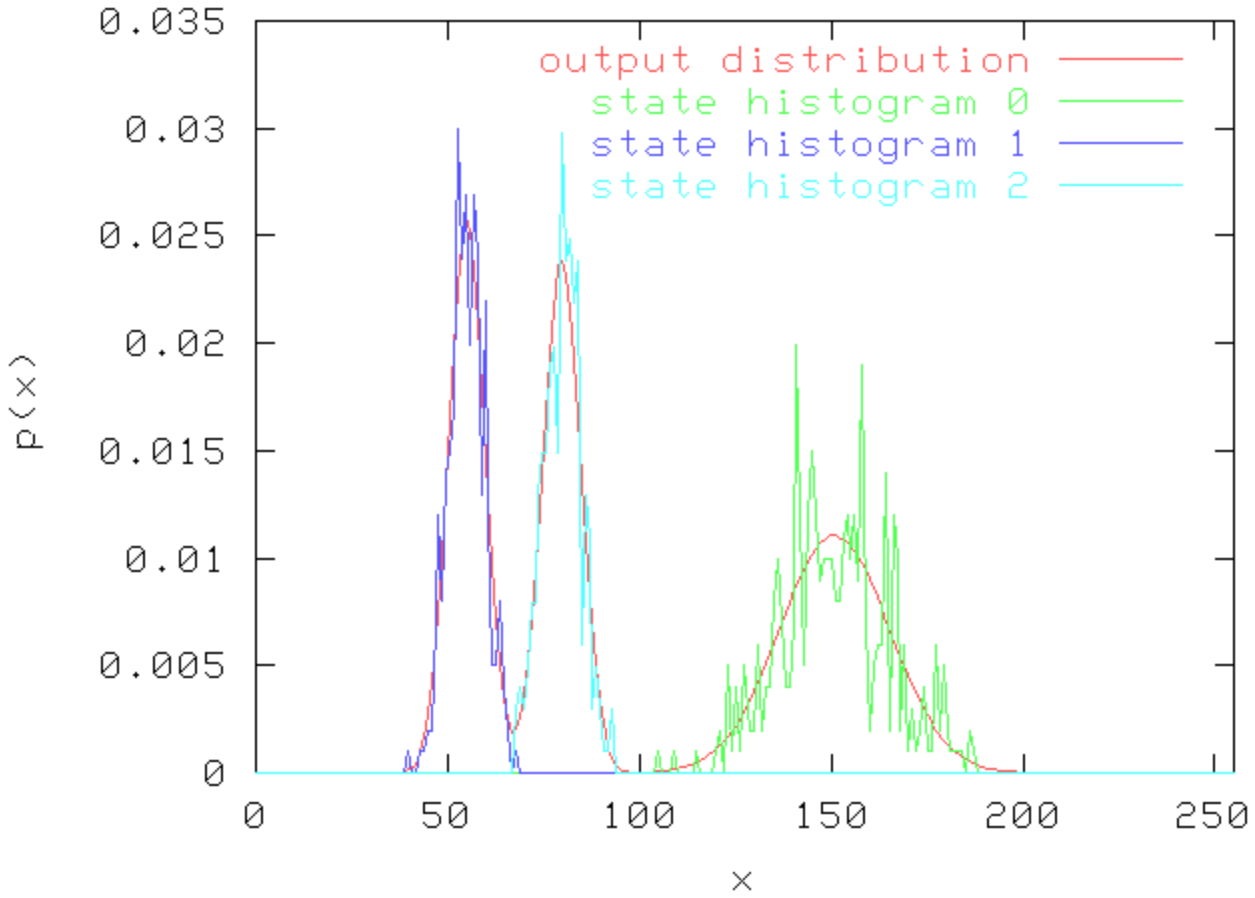




State-Splitting with the χ^2-goodness of fit test

1. Initialize an HMM with N=1 State
2. Train the HMM using the Baum-Welch Algorithm
3. Generate State histograms using the Viterbi Algorithm.
4. Compute the χ^2-statistic between the state distributions and the state histogram.
5. If at least one difference is statistically significant, split the state with the highest significance.
6. Go to step 2.

Simulation


output distribution
state histogram 0
state histogram 1
state histogram 2
0.035
0.03
0.025
0.02
0.015
0.01
0.005
0
p(x)
0
50
100
150
200
250
x

Limitations of the χ^2-test state-splitting method

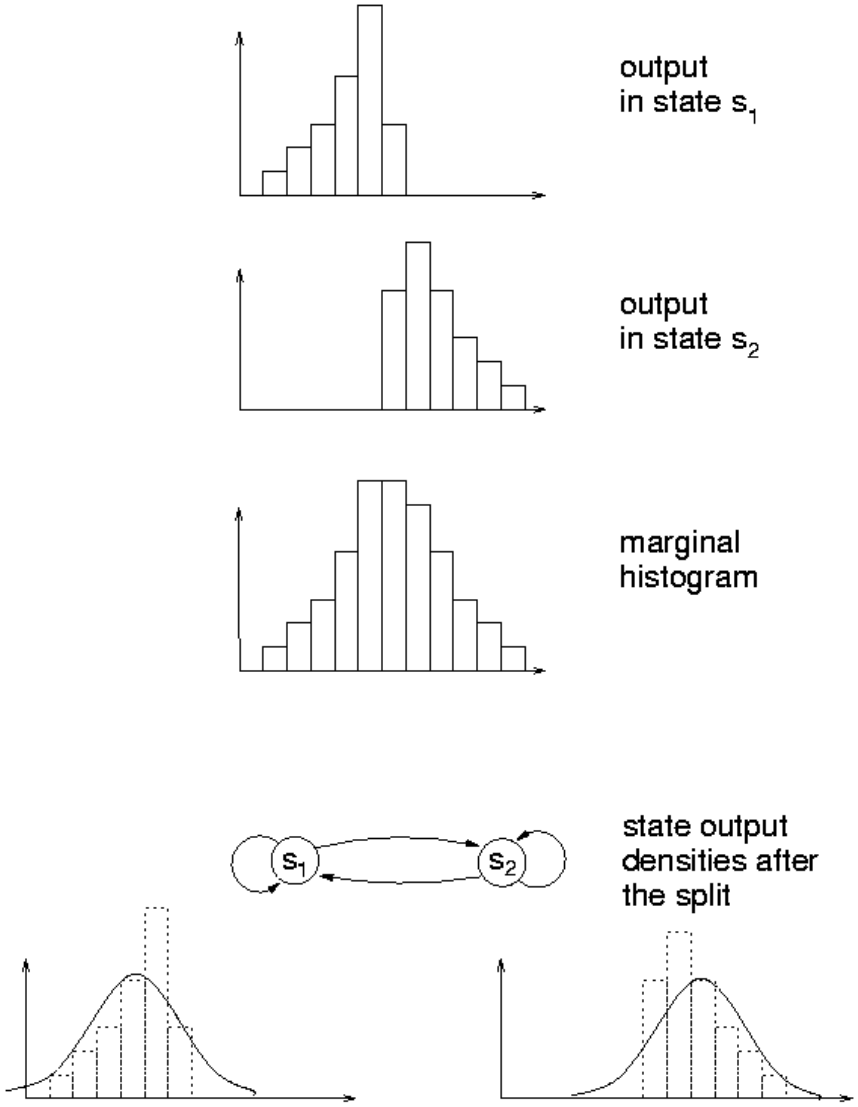
Unless the distributions are indeed Gaussian, the chi-squared test is not robust!

Merging may be needed to deal with false splits

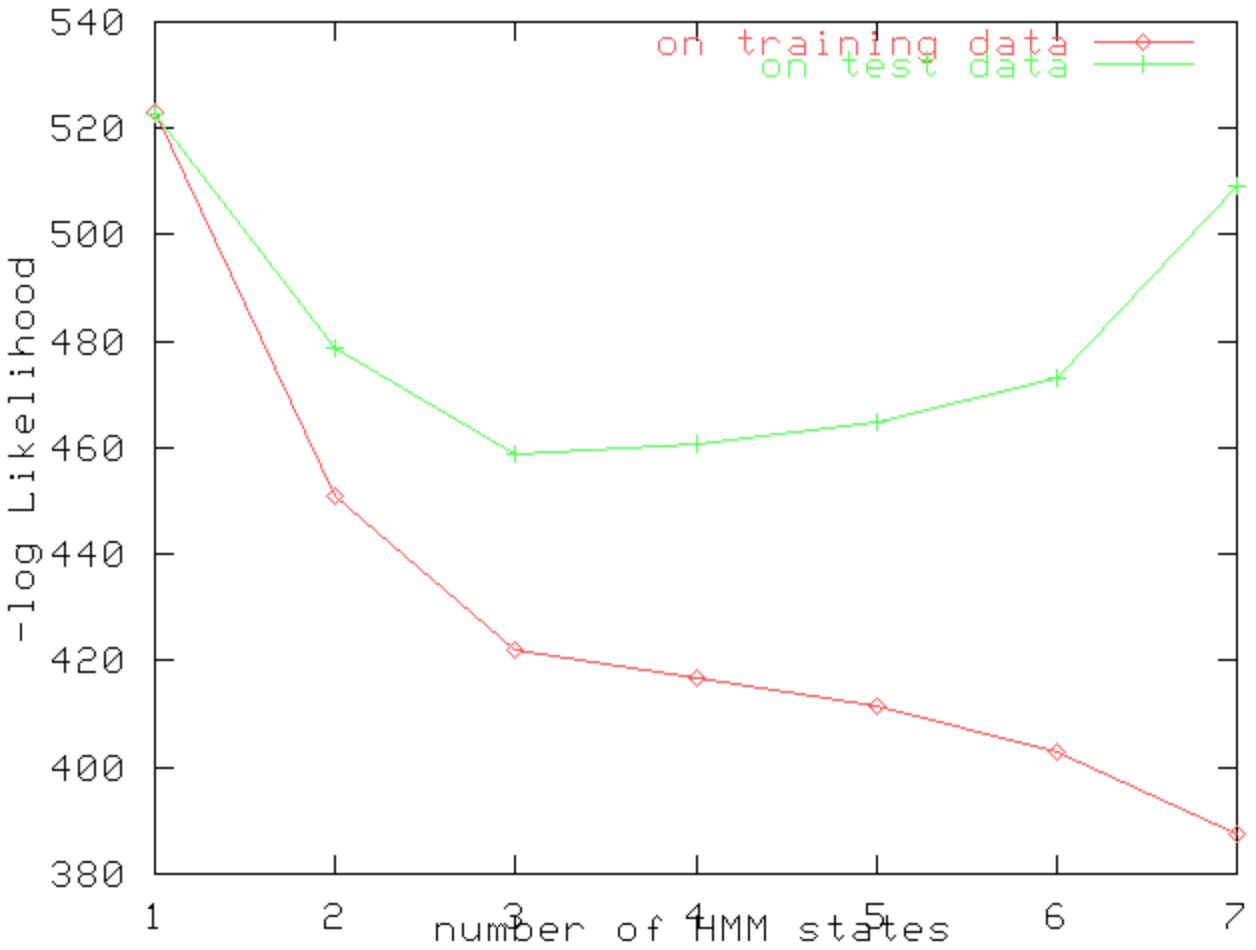
Generalization to higher dimensions is an issue.

Main problem: robustness to false splits.

State Splitting *(temporal split)*



Overfitting Effect



State-Splitting with Cross-Validation

1. Initialize an HMM with N=1 states.
2. Train the model on the training set using the Baum-Welch Algorithm.
3. Compute the data likelihood of the test set, given this model.
4. If the likelihood on the test set decreases with the split, stop.
5. Select a split candidate with a goodness-of-fit test and split this state.
6. Go to step 2.

Model Selection

Given several models $M_1, ..., M_k$

And data $x=x_1,...,x_n$

Cost function for Model Selection:

$$Q(\theta_k \mid x) = -\log L(\hat{\theta}_k \mid x) + C(n, \eta(k)),$$

$\eta(k)$: Number of free model parameters

$\hat{\theta}(k)$: Model parameters (estimated during MLE).

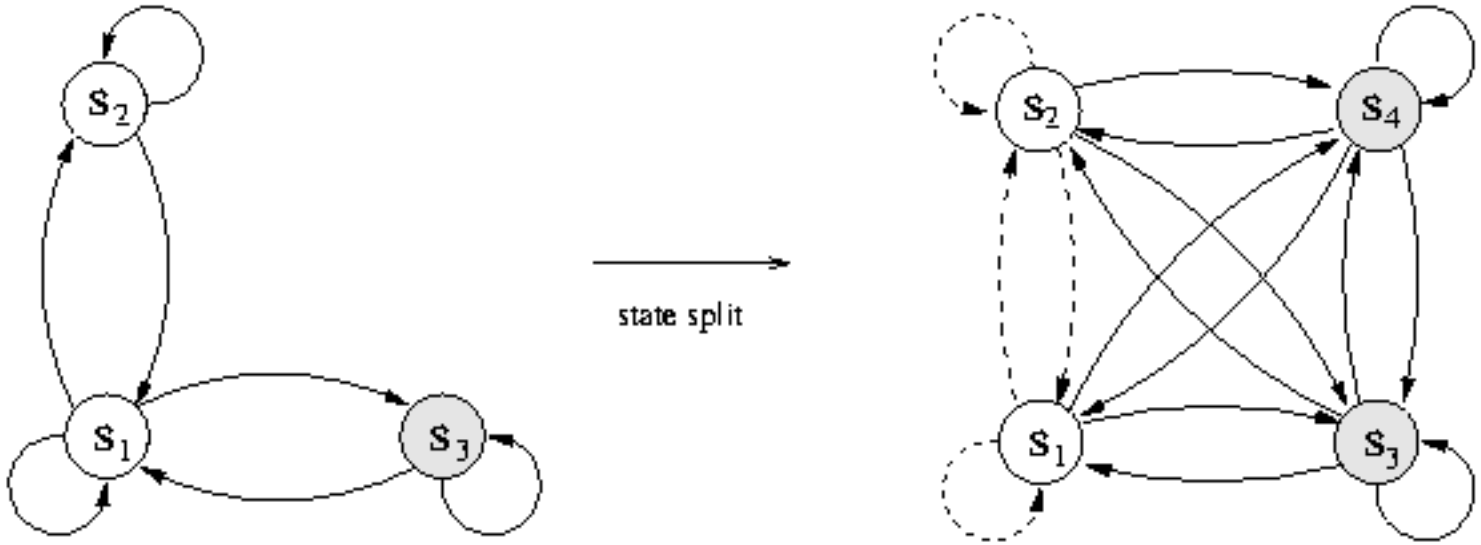
Penalty for model complexity: $C(n, \eta(k))$:

An Information Criterion (AIC): $\eta(k)$

Minimum Description Length (MDL): $\eta(k) \log n$

State Splitting

- Split the state which maximally increases the likelihood.
- Compute the likelihood increase only on a constrained subset of the model parameters



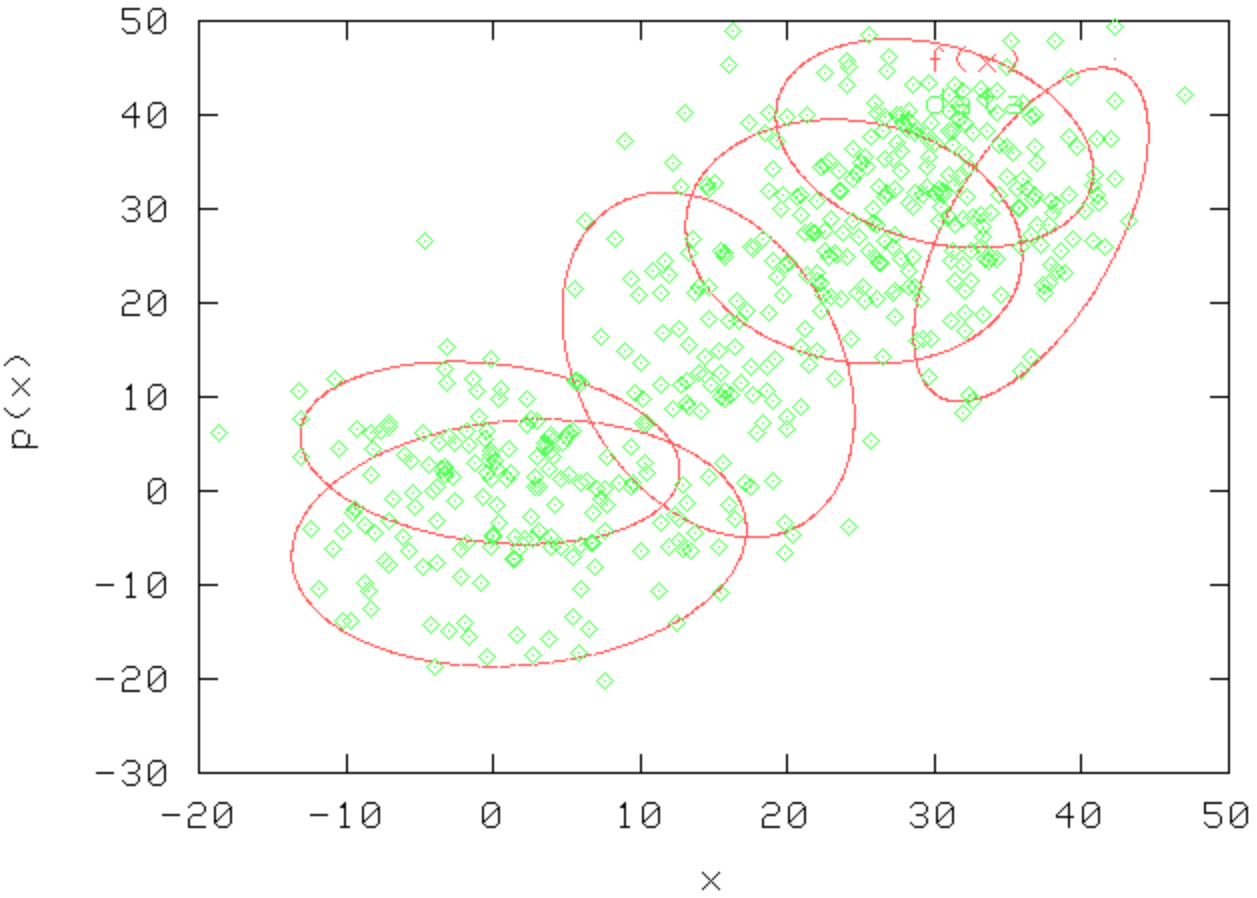
State-Splitting with MDL

1. Initialize a model with N=1 states and Train the HMM with the Baum Welch Algorithm.
2. Select the split which maximally increases the likelihood on a constrained subset of parameters.
3. Determine the likelihood increase for the complete model by training a model after state-splitting with Baum-Welch.
4. If the increased likelihood is greater than the MDL penalty difference, stop.
5. Go to step 2.

Simulation

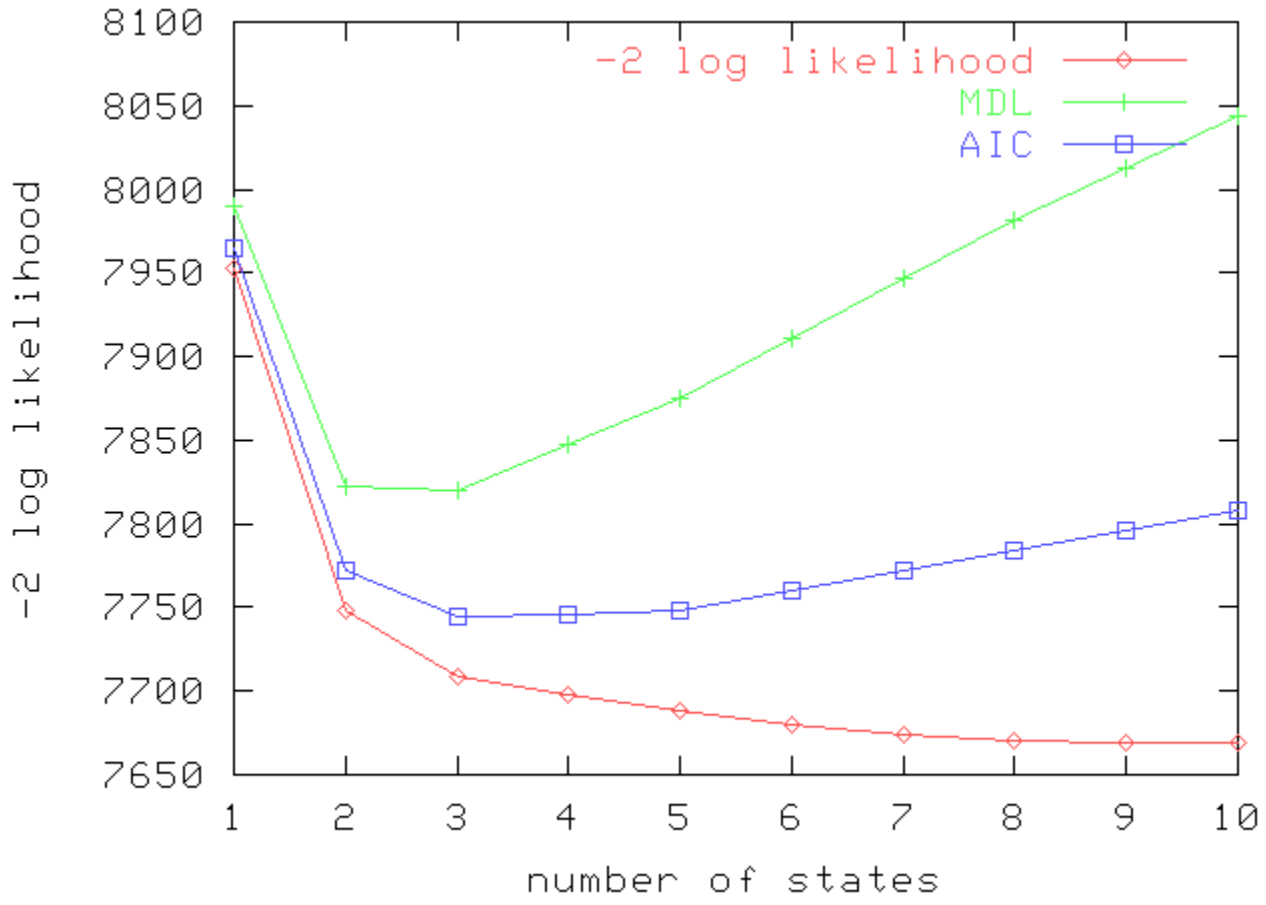
FIAS Frankfurt Institute for Advanced Studies

GOETHE UNIVERSITÄT FRANKFURT AM MAIN



www.goethe-universitaet.de

Cost function for Model Selection



FIAS Frankfurt Institute for Advanced Studies

GOETHE UNIVERSITÄT FRANKFURT AM MAIN

Online Version

Need an online version to handle drifts in data over time. Devised an algorithm based on the online version of the Baum-Welch algorithm.

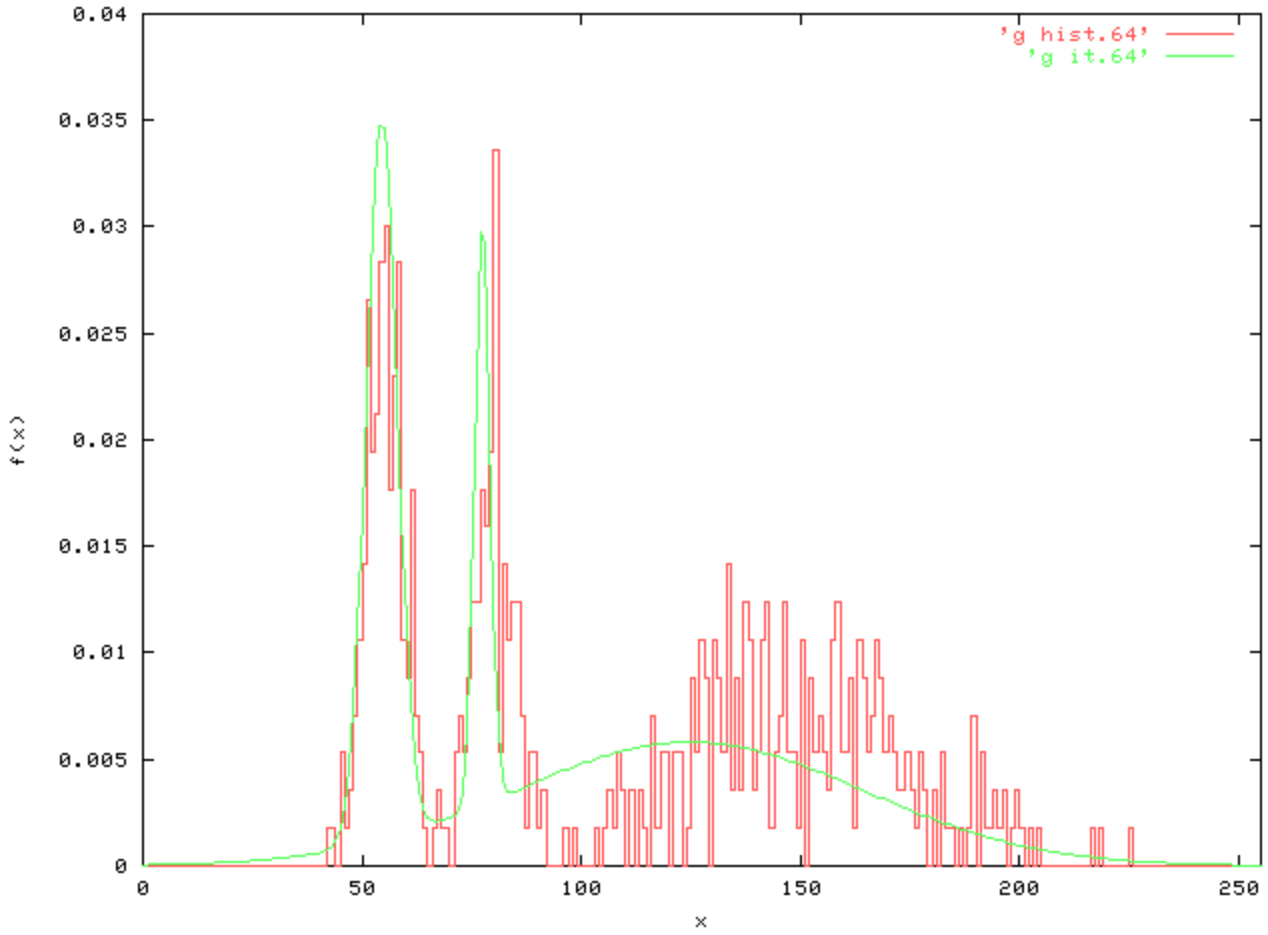
For each data sample x_i the model parameters are updated.

Algorithm Motivated by 'Incremental EM-Versions' of Baum-Welch (Nowlan, 1991)(Neal, Hinton).

Converges for stationary processes

Further Advantage: Little memory requirement

Example


'g hist.64'
'g it.64'
f(x)
x
0.04
0.035
0.03
0.025
0.02
0.015
0.01
0.005
0
0
50
100
150
200
250

FIAS Frankfurt Institute for Advanced Studies

GOETHE UNIVERSITÄT FRANKFURT AM MAIN

Demo

Static Background

Stauffer-Grimson (no state information)

HMM State Model

Conclusions

HMM State-Splitting Algorithms to select HMM topology and parameter estimation.

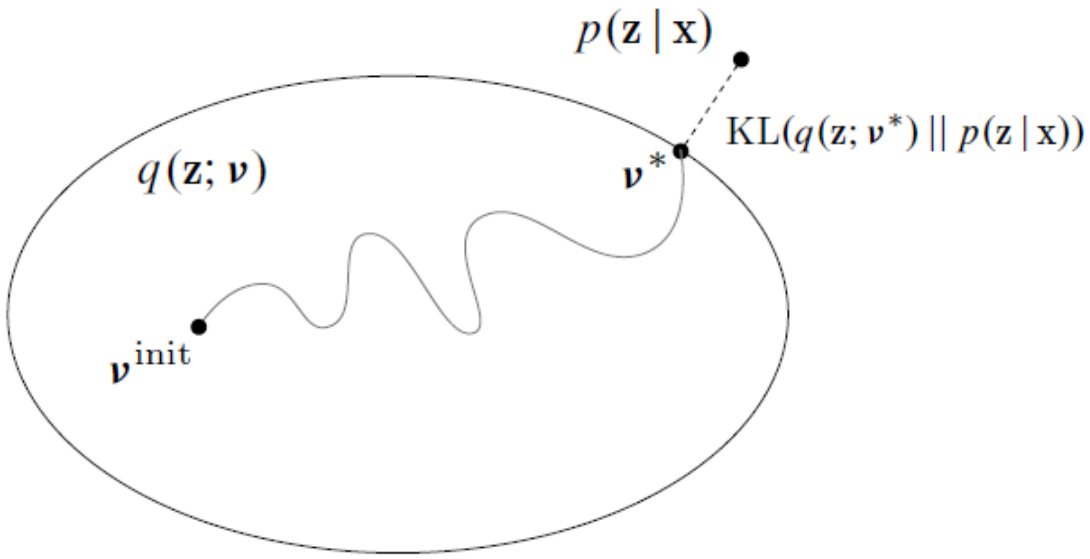
Online-Algorithm for Adapting model parameters.

Application for Global state changes in video analysis.

Variational Inference

Source: David Blei's Tutorial Article

Variational Inference



- VI turns **inference into optimization**.
- Posit a **variational family** of distributions over the latent variables,

$$q(\mathbf{z}; \boldsymbol{\nu})$$

- Fit the **variational parameters** $\boldsymbol{\nu}$ to be close (in KL) to the exact posterior.
  (There are alternative divergences, which connect to algorithms like EP, BP, and others.)

Setup

- As usual, we will assume that $x = x_{1:n}$ are observations and $z = z_{1:m}$ are hidden variables. We assume additional parameters α that are fixed.

- Note we are general—the hidden variables might include the "parameters," e.g., in a traditional inference setting. (In that case, α are the hyperparameters.)

- We are interested in the **posterior distribution**,

$$p(z \mid x, \alpha) = \frac{p(z, x \mid \alpha)}{\int_z p(z, x \mid \alpha)}. \tag{1}$$

- As we saw earlier, the posterior links the data and a model. It is used in all downstream analyses, such as for the predictive distribution.

- (Note: The problem of computing the posterior is an instance of a more general problem that variational inference solves.)

Motivation

- **Cannot compute exact posterior distribution for many models**

  - Consider the Bayesian mixture of Gaussians,

    1. Draw $\mu_k \sim \mathcal{N}(0, \tau^2)$ for $k = 1 \dots K$.
    2. For $i = 1 \dots n$:
       (a) Draw $z_i \sim \text{Mult}(\pi)$;

       (b) Draw $x_i \sim \mathcal{N}(\mu_{z_i}, \sigma^2)$.

  - Suppressing the fixed parameters, the posterior distribution is

$$p(\mu_{1:K}, z_{1:n} \mid x_{1:n}) = \frac{\prod_{k=1}^{K} p(\mu_k) \prod_{i=1}^{n} p(z_i) p(x_i \mid z_i, \mu_{1:K})}{\int_{\mu_{1:K}} \sum_{z_{1:n}} \prod_{k=1}^{K} p(\mu_k) \prod_{i=1}^{n} p(z_i) p(x_i \mid z_i, \mu_{1:K})}. \quad (2)$$

  - The numerator is easy to compute for any configuration of the hidden variables. The problem is the denominator.

Main Idea

- We return to the general $\{x, z\}$ notation.

- The main idea behind variational methods is to pick a family of distributions over the latent variables with its own **variational parameters**,

$$q(z_{1:m} \mid \nu). \tag{5}$$

- Then, find the setting of the parameters that makes q close to the posterior of interest.

- Use q with the fitted parameters as a proxy for the posterior, e.g., to form predictions about future data or to investigate the posterior distribution of the hidden variables.

- Typically, the true posterior is not in the variational family. (Draw the picture from Wainwright and Jordan, 2008.)

KL Divergence

- We measure the closeness of the two distributions with Kullback-Leibler (KL) divergence.
- This comes from **information theory**, a field that has deep links to statistics and machine learning. (See the books "Information Theory and Statistics" by Kullback and "Information Theory, Inference, and Learning Algorithms" by MacKay.)
- The KL divergence for variational inference is

$$\mathrm{KL}(q||p) = \mathrm{E}_q \left[\log \frac{q(Z)}{p(Z \mid x)} \right]. \tag{6}$$

- Intuitively, there are three cases
  - If q is high and p is high then we are happy.
  - If q is high and p is low then we pay a price.
  - If q is low then we don't care (because of the expectation).

Evidence Lower Bound (ELBO)

- We actually can't minimize the KL divergence exactly, but we can minimize a function that is equal to it up to a constant. This is the **evidence lower bound** (ELBO).

- We use Jensen's inequality on the log probability of the observations,

$$\begin{aligned}
\log p(x) &= \log \int_z p(x, z) && (8)\\
&= \log \int_z p(x, z) \frac{q(z)}{q(z)} && (9)\\
&= \log \left(\mathrm{E}_q \left[\frac{p(x, Z)}{q(z)} \right] \right) && (10)\\
&\geq \mathrm{E}_q[\log p(x, Z)] - \mathrm{E}_q[\log q(Z)]. && (11)
\end{aligned}$$

This is the ELBO. (Note: This is the same bound used in deriving the expectation-maximization algorithm.)

ELBO (continued)

- We choose a family of variational distributions (i.e., a parameterization of a distribution of the latent variables) such that the expectations are computable.

- Then, we maximize the ELBO to find the parameters that gives as tight a bound as possible on the marginal probability of x.

- Note that the second term is the entropy, another quantity from information theory.

ELBO – KL Divergence to Posterior (Relationship)

- What does this have to do with the KL divergence to the posterior?
  - First, note that

$$p(z \mid x) = \frac{p(z, x)}{p(x)}. \tag{12}$$

  - Now use this in the KL divergence,

$$\begin{aligned} \mathrm{KL}(q(z)||p(z \mid x)) &= \mathrm{E}_q\left[\log \frac{q(Z)}{p(Z \mid x)}\right] && (13)\\ &= \mathrm{E}_q[\log q(Z)] - \mathrm{E}_q[\log p(Z \mid x)] && (14)\\ &= \mathrm{E}_q[\log q(Z)] - \mathrm{E}_q[\log p(Z, x)] + \log p(x) && (15)\\ &= -(\mathrm{E}_q[\log p(Z, x)] - \mathrm{E}_q[\log q(Z)]) + \log p(x) && (16) \end{aligned}$$

    This is the negative ELBO plus the log marginal probability of x.

ELBO (continued)

- Notice that $\log p(x)$ does not depend on q. So, as a function of the variational distribution, minimizing the KL divergence is the same as maximizing the ELBO.

- And, the difference between the ELBO and the KL divergence is the log normalizer—which is what the ELBO bounds.

Mean Field Variational Inference

- In mean field variational inference, we assume that the variational family **factorizes**,

$$q(z_1, \ldots, z_m) = \prod_{j=1}^{m} q(z_j). \tag{17}$$

  Each variable is independent. (We are suppressing the parameters ν_j.)

- This is more general that it initially appears—the hidden variables can be grouped and the distribution of each group factorizes.

- Typically, this family does not contain the true posterior because the hidden variables are dependent.
  - E.g., in the Gaussian mixture model all of the cluster assignments z_i are dependent on each other and the cluster locations $\mu_{1:K}$ given the data $x_{1:n}$.
  - These dependencies are often what makes the posterior difficult to work with.

Coordinate Ascent Inference for Optimization of ELBO

- We now turn to optimizing the ELBO for this factorized distribution.

- We will use **coordinate ascent inference**, interatively optimizing each variational distribution holding the others fixed.

Coordinate Ascent Inference – Derivation

- First, recall the chain rule and use it to decompose the joint,

$$p(z_{1:m}, x_{1:n}) = p(x_{1:n}) \prod_{j=1}^{m} p(z_j \mid z_{1:(j-1)}, x_{1:n}) \tag{18}$$

Notice that the z variables can occur in any order in this chain. The indexing from 1 to m is arbitrary. (This will be important later.)

- Second, decompose the entropy of the variational distribution,

$$\mathrm{E}[\log q(z_{1:m})] = \sum_{j=1}^{m} \mathrm{E}_j[\log q(z_j)], \tag{19}$$

where E_j denotes an expectation with respect to $q(z_j)$.

- Third, with these two facts, decompose the the ELBO,

$$\mathcal{L} = \log p(x_{1:n}) + \sum_{j=1}^{m} \mathrm{E}[\log p(z_j \mid z_{1:(j-1)}, x_{1:n})] - \mathrm{E}_j[\log q(z_j)]. \tag{20}$$

Coordinate Ascent Inference – Derivation

- Consider the ELBO as a function of $q(z_k)$.
  - Employ the chain rule with the variable z_k as the last variable in the list.
  - This leads to the objective function

$$\mathcal{L} = \mathrm{E}[\log p(z_k \mid z_{-k}, x)] - \mathrm{E}_j[\log q(z_k)] + \text{const.} \tag{21}$$

  - Write this objective as a function of $q(z_k)$,

$$\mathcal{L}_k = \int q(z_k)\mathrm{E}_{-k}[\log p(z_k \mid z_{-k}, x)]dz_k - \int q(z_k) \log q(z_k)dz_k. \tag{22}$$

Coordinate Ascent Inference – Derivation

– Take the derivative with respect to $q(z_k)$

$$\frac{d\mathcal{L}_j}{dq(z_k)} = \mathrm{E}_{-k}[\log p(z_k \mid z_{-k}, x)] - \log q(z_k) - 1 = 0 \tag{23}$$

– This (and Lagrange multipliers) leads to the coordinate ascent update for $q(z_k)$

$$q^*(z_k) \propto \exp\{\mathrm{E}_{-k}[\log p(z_k \mid Z_{-k}, x)]\} \tag{24}$$

– But the denominator of the posterior does not depend on z_j, so

$$q^*(z_k) \propto \exp\{\mathrm{E}_{-k}[\log p(z_k, Z_{-k}, x)]\} \tag{25}$$

– Either of these perspectives might be helpful in deriving variational inference algorithms.

Coordinate Ascent Inference – Derivation

- There is a strong relationship between this algorithm and Gibbs sampling.
  - In Gibbs sampling we sample from the conditional.
  - In coordinate ascent variational inference, we iteratively set each factor to

$$\text{distribution of } z_k \propto \exp\{\mathrm{E}[\log(\text{conditional})]\}. \tag{26}$$

- Easy example: Multinomial conditionals
  - Suppose the conditional is multinomial

$$p(z_j \mid z_{-j}, x_{1:n}) := \pi(z_{-j}, x_{1:n}) \tag{27}$$

  - Then the optimal $q(z_j)$ is also a multinomial,

$$q^*(z_j) \propto \exp\{\mathrm{E}[\log \pi(z_{-j}, x)]\} \tag{28}$$

Exponential Family Conditionals

- Suppose each conditional is in the exponential family

$$p(z_j \mid z_{-j}, x) = h(z_j) \exp\{\eta(z_{-j}, x)^\top t(z_j) - a(\eta(z_{-j}, x))\} \tag{29}$$

- This describes *a lot* of complicated models
  - Bayesian mixtures of exponential families with conjugate priors
  - Switching Kalman filters
  - Hierarchical HMMs
  - Mixed-membership models of exponential families
  - Factorial mixtures/HMMs of exponential families
  - Bayesian linear regression

- Notice that any model containing conjugate pairs and multinomials has this property.

Examples of Exponential Family

Gaussian	$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}}\, e^{-\| x-\mu \|^2/(2\sigma^2)}$	$x \in \mathbb{R}$
Bernoulli	$p(x) = \alpha^x \left(1-\alpha\right)^{1-x}$	$x \in \{0,1\}$
Binomial	$p(x) = \binom{n}{x} \alpha^x \left(1-\alpha\right)^{n-x}$	$x \in \{0,1,2,\ldots,n\}$
Multinomial	$p(x) = \frac{n!}{x_1!x_2!\ldots x_n!} \prod_{i=1}^{n} \alpha_i^{x_i}$	$x_i \in \{0,1,2,\ldots,n\}\,,\ \sum_i x_i = n$
Exponential	$p(x) = \lambda\, e^{-\lambda x}$	$x \in \mathbb{R}^+$
Poisson	$p(x) = \frac{e^{-\lambda}}{x!}\, \lambda^x$	$x \in \{0,1,2,\ldots\}$
Dirichlet	$p(x) = \frac{\Gamma\left(\sum_i \alpha_i\right)}{\prod_i \Gamma(\alpha_i)} \prod_i x_i^{\alpha_i - 1}$	$x_i \in [0,1]\,,\ \sum_i x_i = 1$

Natural Parameter Form

$$
\begin{aligned}
p(x) &= \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x-\mu)^2/(2\sigma^2)} \\
&= \frac{1}{\sqrt{2\pi}} \exp\left(-\log\sigma - \frac{x^2}{2\sigma^2} + \frac{\mu x}{\sigma^2} - \frac{\mu^2}{2\sigma^2}\right) \\
&= \underbrace{\frac{1}{\sqrt{2\pi}}}_{h(x)} \exp\big(\theta^\top T(x) - \underbrace{\log\sigma - \mu^2/(2\sigma^2)}_{A(\theta)}\big)
\end{aligned}
$$

where

$$
T(x) = \begin{pmatrix} x \\ x^2 \end{pmatrix} \qquad \theta = \begin{pmatrix} \mu/\sigma^2 \\ -1/(2\sigma^2) \end{pmatrix}
$$

$$
\begin{aligned}
A(\theta) &= \tfrac{\mu^2}{2\sigma^2} + \log\sigma \\
&= -\tfrac{[\theta]_1^2}{4[\theta]_2} - \tfrac{1}{2}\log\left(-2[\theta]_2\right)
\end{aligned}
$$

Exponential Family Case

- Mean field variational inference is straightforward
  - Compute the log of the conditional

$$\log p(z_j \mid z_{-j}, x) = \log h(z_j) + \eta(z_{-j}, x)^\top t(z_j) - a(\eta(z_{-j}, x)) \tag{30}$$

  - Compute the expectation with respect to $q(z_{-j})$

$$\mathrm{E}[\log p(z_j \mid z_{-j}, x)] = \log h(z_j) + \mathrm{E}[\eta(z_{-j}, x)]^\top t(z_j) - \mathrm{E}[a(\eta(z_{-j}, x))] \tag{31}$$

  - Noting that the last term does not depend on q_j, this means that

$$q^*(z_j) \propto h(z_j) \exp\{\mathrm{E}[\eta(z_{-j}, x)]^\top t(z_j)\} \tag{32}$$

    and the normalizing constant is $a(\mathrm{E}[\eta(z_{-j}, x)])$.

- So, the optimal $q(z_j)$ is in the same exponential family as the conditional.

Exponential Family Case - Continued

- Coordinate ascent algorithm
  - Give each hidden variable a variational parameter ν_j, and put each one in the same exponential family as its model conditional,

$$q(z_{1:m} \mid \nu) = \prod_{j=1}^{m} q(z_j \mid \nu_j) \tag{33}$$

  - The coordinate ascent algorithm iteratively sets each natural variational parameter ν_j equal to the expectation of the natural conditional parameter for variable z_j given all the other variables and the observations,

$$\nu_j^* = \mathrm{E}[\eta(z_{-j}, x)]. \tag{34}$$

Bayesian Mixtures of Gaussians

- Let's go back to the Bayesian mixture of Gaussians. For simplicity, assume that the data generating variance is one.

- The latent variables are cluster assignments z_i and cluster means μ_k.

- The mean field family is

$$q(\mu_{1:K}, z_{1:n}) = \prod_{k=1}^{K} q(\mu_k \mid \tilde{\mu}_k, \tilde{\sigma}_k^2) \prod_{i=1}^{n} q(z_i \mid \phi_i), \tag{35}$$

  where $(\tilde{\mu}_k, \tilde{\sigma}_k)$ are Gaussian parameters and ϕ_i are multinomial parameters (i.e., positive K-vectors that sum to one.)

Bayesian Mixtures of Gaussians - Variational Updates

the coordinate update for $q(z_i)$ is

$$q^*(z_i = k) \propto \exp\{\log \pi_k + x_i \mathrm{E}[\mu_k] - \mathrm{E}[\mu_k^2]/2\}. \tag{40}$$

– For the Gaussian conjugate prior, we map

$$\eta = \langle \mu/\sigma^2, 1/\sigma^2 \rangle. \tag{49}$$

– This gives the variational update in mean parameter form,

$$\mathrm{E}[\mu_k] = \frac{\mu_0/\sigma_0^2 + \sum_{i=1}^n \mathrm{E}[z_i^k] x_i}{1/\sigma_0^2 + \sum_{i=1}^n \mathrm{E}[z_i^k]} \tag{50}$$

$$\mathrm{Var}(\mu_k) = 1/(1/\sigma_0^2 + \textstyle\sum_{i=1}^n \mathrm{E}[z_i^k]). \tag{51}$$

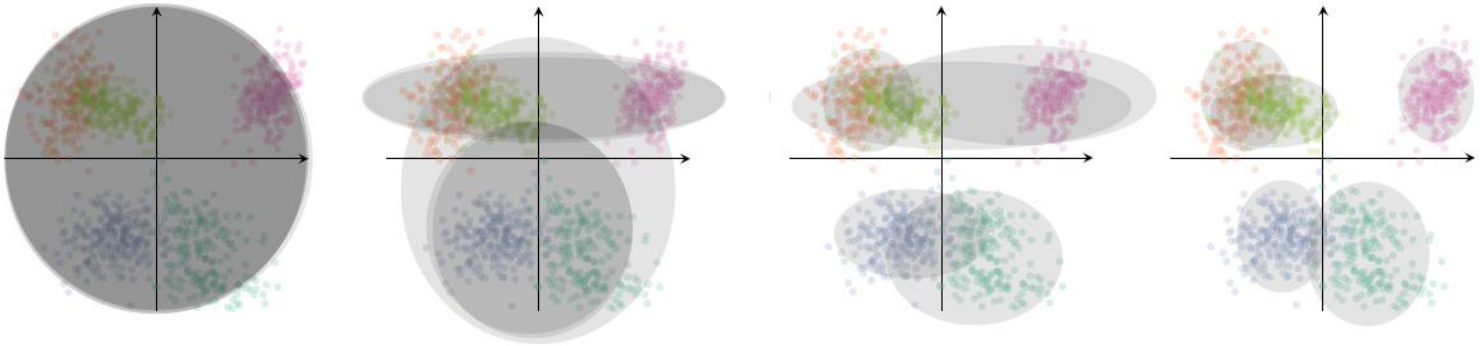
These are the usual Bayesian updates with the data weighted by its variational probability of being assigned to cluster k.

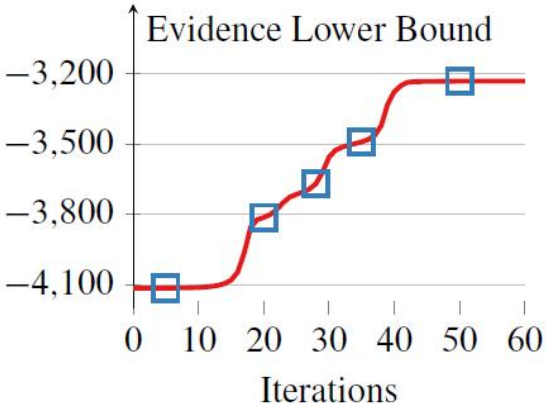
Bayesian Mixtures of Gaussians

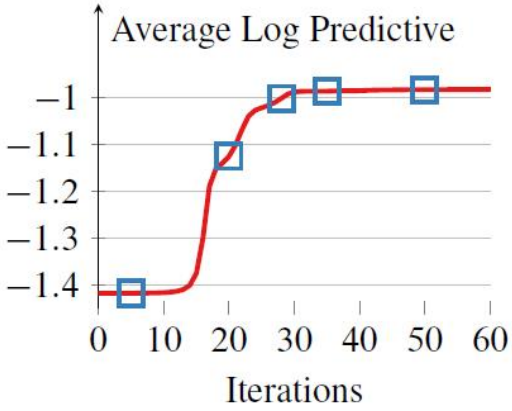
- We are given data $x_{1:n}$, hyperparameters μ_0 and σ_0^2, and a number of groups K.

- The variational distributions are
  - n variational multinomials $q(z_i)$
  - K variational Gaussians $q(\mu_k \mid \tilde{\mu}_k, \tilde{\sigma}_k^2)$.
- Repeat until the ELBO converges:
  1. For each data point x_i
     - Update the variational multinomial $q(z_i)$ from Equation 40.
  2. For each cluster $k = 1 \dots K$
     - Update the mean and variance from Equation 50 and Equation 51.

Mixtures of Gaussians - Example

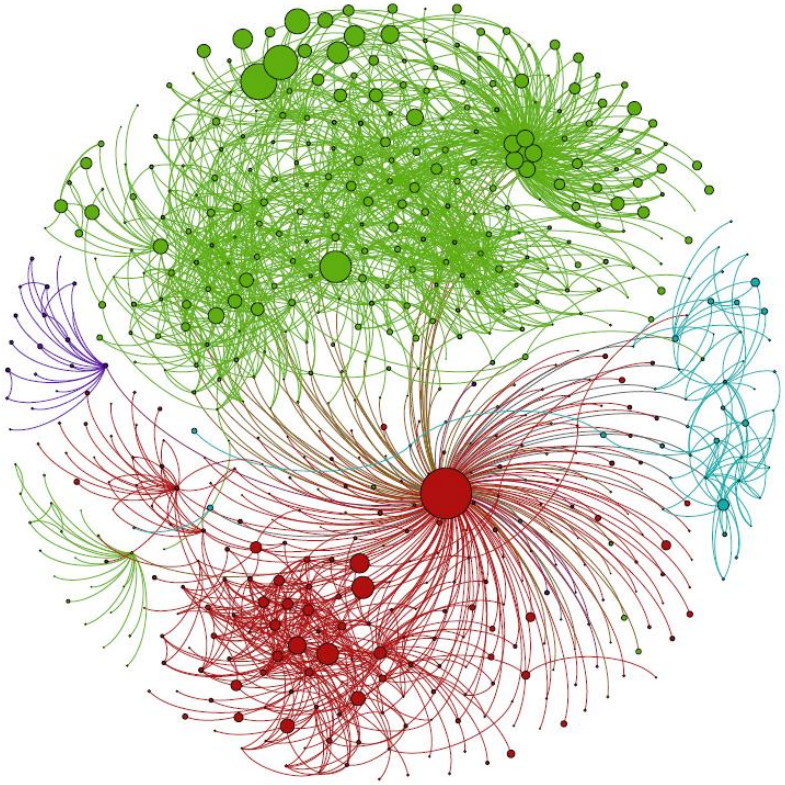






[images by Alp Kucukelbir]

Example Applications



Communities discovered in a 3.7M node network of U.S. Patents

[Gopalan and Blei, PNAS 2013]

Applications (Continued)

1	2	3	4	5
Game	Life	Film	Book	Wine
Season	Know	Movie	Life	Street
Team	School	Show	Books	Hotel
Coach	Street	Life	Novel	House
Play	Man	Television	Story	Room
Points	Family	Films	Man	Night
Games	Says	Director	Author	Place
Giants	House	Man	House	Restaurant
Second	Children	Story	War	Park
Players	Night	Says	Children	Garden

6	7	8	9	10
Bush	Building	Won	Yankees	Government
Campaign	Street	Team	Game	War
Clinton	Square	Second	Mets	Military
Republican	Housing	Race	Season	Officials
House	House	Round	Run	Iraq
Party	Buildings	Cup	League	Forces
Democratic	Development	Open	Baseball	Iraqi
Political	Space	Game	Team	Army
Democrats	Percent	Play	Games	Troops
Senator	Real	Win	Hit	Soldiers

11	12	13	14	15
Children	Stock	Church	Art	Police
School	Percent	War	Museum	Yesterday
Women	Companies	Women	Show	Man
Family	Fund	Life	Gallery	Officer
Parents	Market	Black	Works	Officers
Child	Bank	Political	Artists	Case
Life	Investors	Catholic	Street	Found
Says	Funds	Government	Artist	Charged
Help	Financial	Jewish	Paintings	Street
Mother	Business	Pope	Exhibition	Shot

Topics found in 1.8M articles from the New York Times

[Hoffman, Blei, Wang, Paisley, JMLR 2013]

Open Issues

- alternatives to the optimization of KL as the variational objective function (tighter lower bounds)
- Strong independence assumptions of the mean-field family
- Better understanding of statistical properties of variational inference

- Not Covered in this lecture:
  - Stochastic Gradient Descent combined with Variational Methods
  - Black-box Methods
  - Other methods – e.g. Variational Auto-encoders

Backup