CPSC 540: Machine Learning

Kernel Density Estimation

Mark Schmidt

University of British Columbia

Winter 2018

Last Time: Expectation Maximization

- EM considers learning with observed data O and hidden data H.
- In this case the "observed" log-likelihood has a nasty form,

$$\log p(O \mid \Theta) = \log \left(\sum_H p(O, H \mid \Theta) \right).$$

- EM applies when "complete" likelihood, $p(O, H \mid \Theta)$, has a nice form.
- EM iterations take the form of a weighted "complete" NLL,

$$\Theta^{t+1} = \underset{\Theta}{\text{argmax}} \left\{ \sum_H \alpha_H \log p(O, H \mid \Theta) \right\},$$

where $\alpha_H = p(H \mid O, \Theta^t)$.

- For mixture models, has a closed-form solution for common distributions.

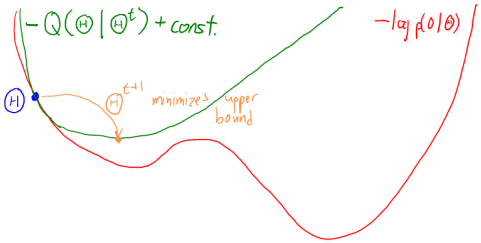
Monotonicity of EM

- Classic result is that EM iterations are monotonic:

$$\log p(O \mid \Theta^{t+1}) \geq \log p(O \mid \Theta^{t}),$$

- We don't need a step-size and this is useful for debugging.

- We can show this by proving that the below picture is "correct":



- The Q function leads to a global bound on the original function.
- At Θ^t the bound matches original function.
  - So if you improve on the Q function, you improve on the original function.

Monotonicity of EM

- Let's show that the Q function gives a global upper bound on NLL:

$$
\begin{aligned}
-\log p(O \mid \Theta) &= -\log \left(\sum_H p(O, H \mid \Theta) \right) && \text{(marginalization rule)} \\
&= -\log \left(\sum_H \alpha_H \frac{p(O, H \mid \Theta)}{\alpha_H} \right) && \text{(for } \alpha_H \neq 0\text{)} \\
&\leq -\sum_H \alpha_H \log \left(\frac{p(O, H \mid \Theta)}{\alpha_H} \right),
\end{aligned}
$$

because $-\log(z)$ is convex and the α_H are a convex combination.

Monotonicity of EM

- Using that log turns multiplication into addition we get

$$\begin{aligned} -\log p(O \mid \Theta) &\leq -\sum_H \alpha_H \log\left(\frac{p(O, H \mid \Theta)}{\alpha_H}\right) \\ &= \underbrace{-\sum_H \alpha_H \log p(O, H \mid \Theta)}_{Q(\Theta \mid \Theta^t)} + \underbrace{\sum_H \alpha_H \log \alpha_H}_{\text{negative entropy}} \\ &= -Q(\Theta \mid \Theta^t) - \text{entropy}(\alpha), \end{aligned}$$

  so we have the first part of the picture, $-\log p(O \mid \Theta^{t+1}) \leq -Q(\Theta|\Theta^t) + \text{const}$.
  - Entropy is a measure of how "random" the α_H values are.
  - Bound gets tighter for hidden data H that is more "predictable".

- Now we need to show that this holds with equality at Θ^t.

Bound on Progress of Expectation Maximization

- To show equality at Θ^t we use definition of conditional probability,

$$p(H \mid O, \Theta^t) = \frac{p(O, H \mid \Theta^t)}{p(O \mid \Theta^t)} \quad \text{or} \quad \log p(O \mid \Theta^t) = \log p(O, H \mid \Theta^t) - \log p(H \mid O, \Theta^t)$$

- Multiply by α_H and summing over H values,

$$\sum_H \alpha_H \log p(O \mid \Theta^t) = \underbrace{\sum_H \alpha_H \log p(O, H \mid \Theta^t}_{Q(\Theta^t \mid \Theta^t)} - \sum_H \alpha_H \log \underbrace{p(H \mid O, \Theta^t)}_{\alpha_H}.$$

- Which gives the result we want:

$$\log p(O \mid \Theta^t) \underbrace{\sum_H \alpha_H}_{=1} = Q(\Theta^t \mid \Theta^t) + \text{entropy}(\alpha),$$

Bound on Progress of Expectation Maximization

- Thus we have the two bounds

$$\log p(O \mid \Theta) \geq Q(\Theta \mid \Theta^t) + \text{entropy}(\alpha)$$
$$\log p(O \mid \Theta^t) = Q(\Theta^t \mid \Theta^t) + \text{entropy}(\alpha).$$

- Subtracting these and using $\Theta = \Theta^{t+1}$ gives a stronger result,

$$\log p(O \mid \Theta^{t+1}) - \log p(O \mid \Theta^t) \geq Q(\Theta^{t+1} \mid \Theta^t) - Q(\Theta^t \mid \Theta^t),$$

  that we improve objective by at least the decrease in Q.

- Inequality holds for any choice of Θ^{t+1}.
  - Approximate M-steps are ok: we just need to decrease Q to improve likelihood.

- For imputation, we instead improve "complete" log-likelihood, $\log p(O, H \mid \Theta^t)$.
  - Which isn't quite what we want, treats hidden data as a "parameter".

Convergence of Expectation Maximization

- We've shown that

$$\log p(O \mid \Theta^{t+1}) - \log p(O \mid \Theta^t) \geq Q(\Theta^{t+1} \mid \Theta^t) - Q(\Theta^t \mid \Theta^t),$$

  that guaranteed progress is at least as large as difference in Q.

- Does this imply convergence?
  - Yes, the algorithm can't keep improving if the likelihood is bounded above.

- Does this imply convergence to a local optimum or a stationary point?
  - No, although many papers wrongly say that it does.

  - Could have maximum of 3 and objective values of $1, 1.5, 1.75, 1.875, \ldots$

Convergence Rate of Expectation Maximization

- Can we say EM converges to stationary point or analyze convergence rate?
- If $\log p(O \mid \Theta)$ is differentiable, then we can show that

$$\nabla \log p(O \mid \Theta^t) = \nabla Q(\Theta^t \mid \Theta^t),$$

that gradient of bound agrees with gradient of function at Θ^t.

- If the bound Q is L-Lipschitz continuous, then we have

$$-Q(\Theta^{t+1} \mid \Theta^t) \leq -Q(\Theta^t \mid \Theta^t) - \frac{1}{2L}\|\nabla Q(\Theta^t)\|^2,$$

since optimizing Q does at least as well as one iteration of gradient descent.

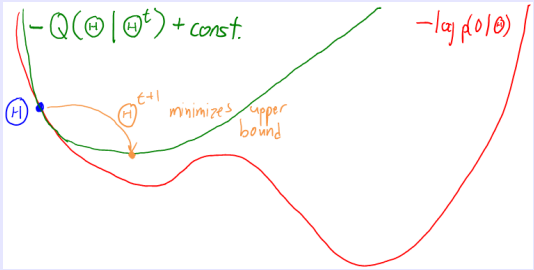
- Using our relationships between Q and objective f gives our usual progress bound

$$f(\Theta^{t+1}) \leq f(\Theta^t) - \frac{1}{2L}\|\nabla f(\Theta^t)\|^2,$$

so EM has convergence rate at least as fast as gradient descent.

Convergence Rate of Expectation Maximization

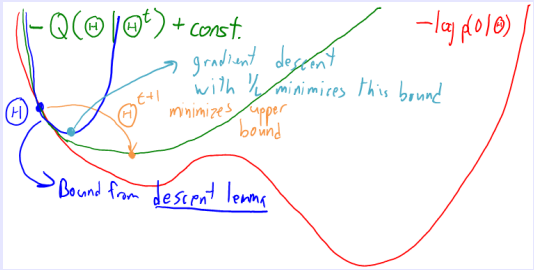
- Expectation maximization decreases f at least much as gradient descent:



- Slight subtle point: we measure L across Q values, rather than for f.

Convergence Rate of Expectation Maximization

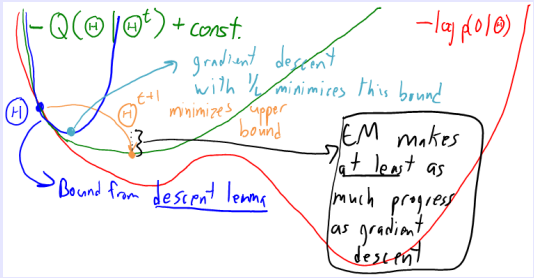
- Expectation maximization decreases f at least much as gradient descent:



- Slight subtle point: we measure L across Q values, rather than for f.

Convergence Rate of Expectation Maximization

- Expectation maximization decreases f at least much as gradient descent:



- Slight subtle point: we measure L across Q values, rather than for f.

Outline

A Non-Parametric Mixture Model

- The classic parametric mixture model has the form

$$p(x^i) = \sum_{c=1}^{k} p(z^i = c)p(x^i \mid z^i = c).$$

- A natural way to define a non-parametric mixture model is

$$p(x^i) = \sum_{j=1}^{n} p(z^i = j)p(x^i \mid z^i = j),$$

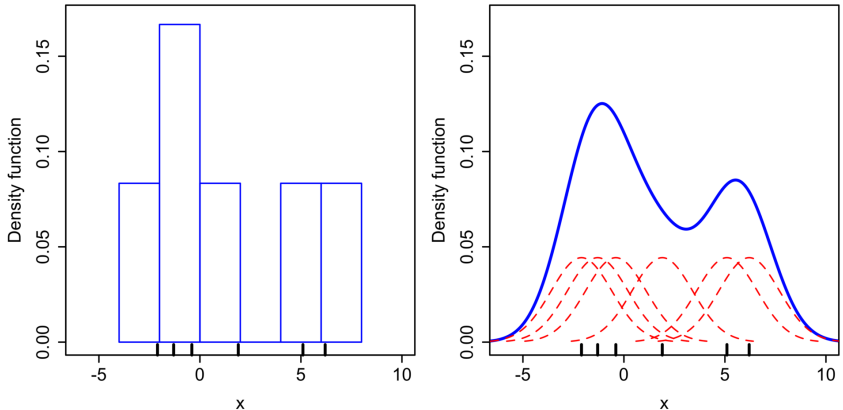
  where we have one mixture for every training example i.
- Common example: z^i is uniform and $x^i \mid z^i$ is Gaussian with mean x^j,

$$p(x^i) = \frac{1}{n} \sum_{j=1}^{n} \mathcal{N}(x^i \mid x^j, \sigma^2 I),$$

  and we use a shared covariance $\sigma^2 I$ (σ can be estimated by cross-validation).
- This is a special case of kernel density estimation (or Parzen window).

Histogram vs. Kernel Density Estimator

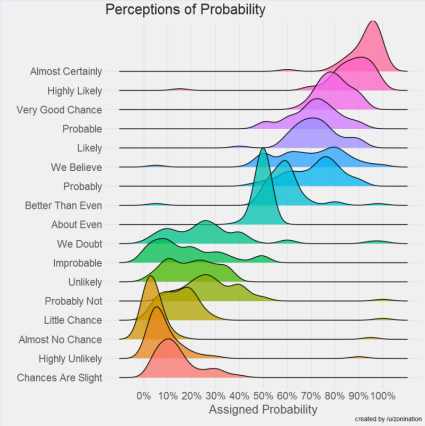
- Think of kernel density estimator as a generalization of histogram:



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

Kernel Density Estimator for Visualization

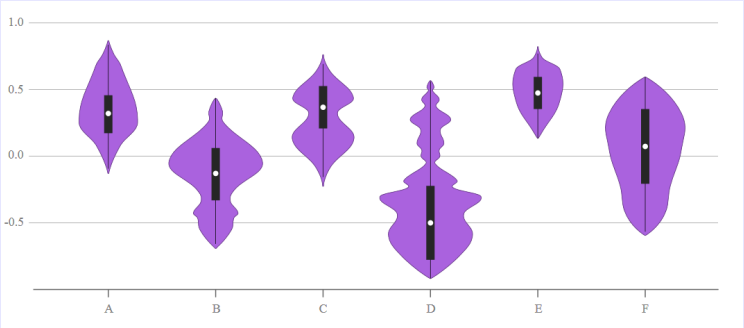
- Visualization of people's opinions about what "likely" and other words mean.



http://blog.revolutionanalytics.com/2017/08/probably-more-probably-than-probable.html

Violin Plot: Added KDE to a Boxplot

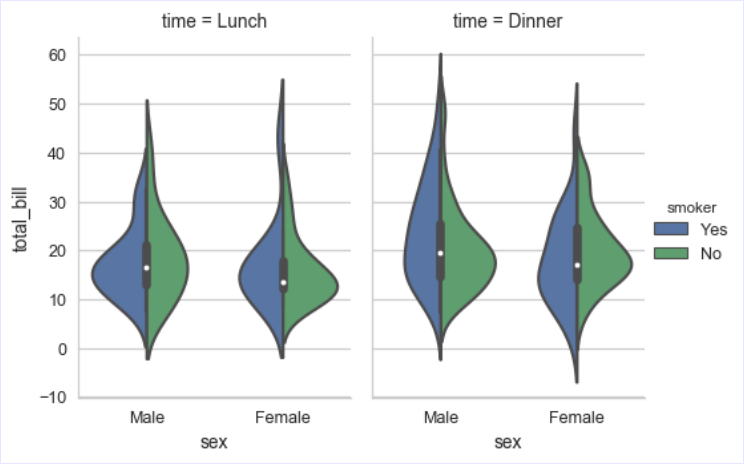
- Violin plot adds KDE to a boxplot:



https://datavizcatalogue.com/methods/violin_plot.html

Violin Plot: Added KDE to a Boxplot

- Violin plot adds KDE to a boxplot:



https://seaborn.pydata.org/generated/seaborn.violinplot.html

Kernel Density Estimation

- The 1D kernel density estimation (KDE) model uses

$$p(x^i) = \frac{1}{n}\sum_{j=1}^{n} k_\sigma \underbrace{(x^i - x^j)}_{r},$$

  where the PDF k is the "kernel" and the parameter σ is the "bandwidth".
- In the previous slide we used the (normalized) Gaussian kernel,

$$k_1(r) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{r^2}{2}\right), \quad k_\sigma(r) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{r^2}{2\sigma^2}\right).$$

- Note that we can add a "bandwith" (standard deviation) σ to any PDF k_1, using

$$k_\sigma(r) = \frac{1}{\sigma} k_1\left(\frac{r}{\sigma}\right),$$

  from the change of variables formula for probabilities ($|\frac{d}{dr}\left[\frac{r}{\sigma}\right]| = \frac{1}{\sigma}$).
- Under common choices of kernels, KDEs can model any continuous density.

Efficient Kernel Density Estimation

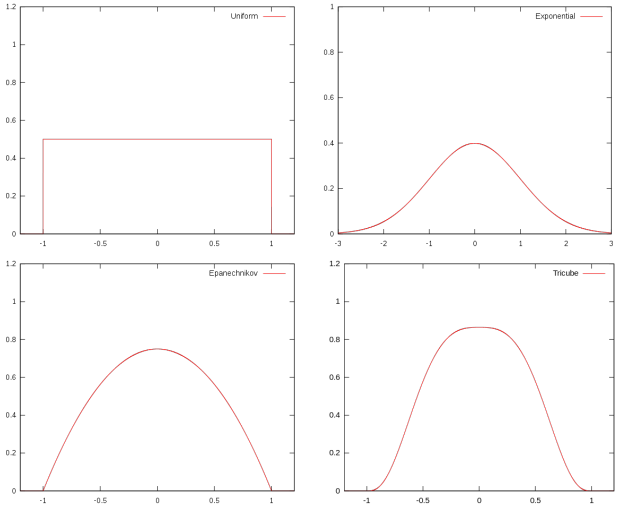
- KDE with the Gaussian kernel is slow at test time:
  - We need to compute distance of test point to every training point.
- A common alternative is the Epanechnikov kernel,

$$k_1(r) = \frac{3}{4}\left(1 - r^2\right)\mathcal{I}\left[|r| \leq 1\right].$$

- This kernel has two nice properties:
  - Epanechnikov showed that it is asymptotically optimal in terms of squared error.
  - It can be much faster to use since it only depends on nearby points (use hashing).
    - You can use hashing to quickly find neighbours in training data.
- It is non-smooth at the boundaries but many smooth approximations exist.
  - Quartic, triweight, tricube, cosine, etc.

Visualization of Common Kernel Functions

Histogram vs. Gaussian vs. Epanechnikov vs. tricube:



Multivariate Kernel Density Estimation

- The multivariate kernel density estimation (KDE) model uses

$$p(x^i) = \frac{1}{n}\sum_{j=1}^{n} k_A(\underbrace{x^i - x^j}_{r}),$$

- The most common kernel is a product of independent Gaussians,

$$k_I(r) = \frac{1}{(2\pi)^{\frac{d}{2}}} \exp\left(-\frac{\|r\|^2}{2}\right).$$

- We can add a bandwith matrix A to any kernel using

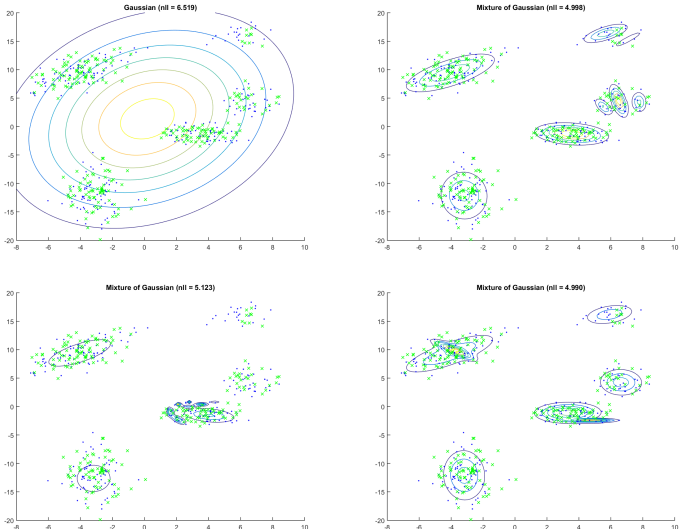
$$k_A(r) = \frac{1}{|A|} k_1(A^{-1}r) \qquad \text{(generalizes } k_\sigma(r) = \frac{1}{\sigma} k_1\left(\frac{r}{\sigma}\right)\text{)},$$

  and in Gaussian case we get a multivariate Gaussian with $\Sigma = AA^T$.

- To reduce number of parameters, we typically:
  - Use a product of independent distributions and use $A = \sigma I$ for some σ.

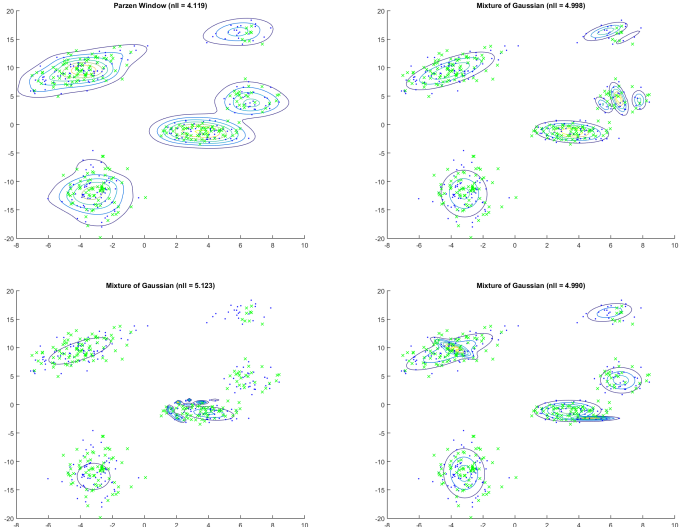
KDE vs. Mixture of Gaussian

- By fixing mean/covariance/k, we don't have to worry about local optima.



KDE vs. Mixture of Gaussian

- By fixing mean/covariance/k, we don't have to worry about local optima.



Mean-Shift Clustering

- Mean-shift clustering uses KDE for clustering:
  - Define a KDE on the training examples, and then for test example $\hat{x}$:
    - Run gradient descent to maximize $p(x)$ starting from $\hat{x}$.
  - Clusters are points that reach same local minimum.
- `https://spin.atomicobject.com/2015/05/26/mean-shift-clustering`
- Not sensitive to initialization, no need to choose k, can find non-convex clusters.
- Similar to density-based clustering from 340.
  - But doesn't require uniform density within cluster.
  - And can be used for vector quantization.
- "The 5 Clustering Algorithms Data Scientists Need to Know":
  - `https://towardsdatascience.com/the-5-clustering-algorithms-data-scientists-need-to-know-a36d136ef68`

Continuous Mixture Models

- We've been discussing mixture models where z^i is discrete,

$$p(x^i) = \sum_{z^i=1}^{k} p(z^i)p(x^i \mid z^i = c).$$

- We can also consider mixtures models where z^i is continuous,

$$p(x^i) = \int_{z^i} p(z^i)p(x^i \mid z^i = c)dz^i.$$

- Unfortunately, computing the integral might be hard.
  - But if both probabilities are Gaussian then it's straightforward.

Probabilistic PCA

- In 340 we discussed PCA, which approximates (centered) x^i by

$$x^i \approx W^T z^i.$$

- In probabilistic PCA we assume that

$$x^i \sim \mathcal{N}(W^T z^i, \sigma^2 I), \quad z^i \sim \mathcal{N}(0, I).$$

- Continuous mixture integral will be marginal of a joint Gaussian, and gives

$$x^i \mid W \sim \mathcal{N}(0, W^T W + \sigma^2 I).$$

- Regular PCA is obtained as the limit of σ^2 going to 0.
  - Shows that PCA is just fitting a multivariate Gaussian with a restricted form for Σ.
  - Allows you to do things like mixture of PCAs.

Factor Analysis

- A related method for discovering latent factors is factor analysis (FA).
  - A standard tool and widely-used across science and engineering.

Trait	Description
Openness	Being curious, original, intellectual, creative, and open to new ideas.
Conscientiousness	Being organized, systematic, punctual, achievement-oriented, and dependable.
Extraversion	Being outgoing, talkative, sociable, and enjoying social situations.
Agreeableness	Being affable, tolerant, sensitive, trusting, kind, and warm.
Neuroticism	Being anxious, irritable, temperamental, and moody.

https://new.edu/resources/big-5-personality-traits

- Historical applications are measures of intelligence and personality traits.
  - Some controversy, like trying to find factors of intelligence due to race.

(without normalizing for socioeconomic factors)

Factor Analysis

- FA approximates (centered) x^i by

$$x^i \approx W^T z^i,$$

  and assumes z^i and $x^i \mid z^i$ are Gaussian.

- Which should sound familiar...

- Are PCA and FA the same?
  - Both are more than 100 years old.
  - There are many online discussions about whether they are the same.
    - Some software packages run PCA when you call their FA method.
    - Some online discussions claiming they are completely different.

PCA vs. Factor Analysis

- In probabilistic PCA we assume

$$x^i \mid z^i \sim \mathcal{N}(W^T z^i, \sigma^2 I), \quad z^i \sim \mathcal{N}(0, I),$$

  and we obtain PCA as $\sigma \to 0$.

- In FA we assume

$$x^i \mid z^i \sim \mathcal{N}(W^T z^i, D), \quad z^i \sim \mathcal{N}(0, I),$$

  where D is a diagonal matrix.

- The difference is that you can have a noise variance for each dimension.
  - So FA has extra degrees of freedom in variance of original variables.
  - In practice there often isn't a huge difference.

Summary

- Monotonicity of EM: EM is guaranteed not to decrease likelihood.
  - Very-recent results giving convergence rates.
- Kernel density estimation: Non-parametric density estimation method.
  - Allows smooth variations on histograms.
- Probabilistic PCA:
  - Continuous mixture models based on Gaussian assumptions.
  - Factor analysis extends probabilistic PCA with different noise in each dimension.
    - Very similar but not identical to PCA.
- Next time: the sad truth about rain in Vancouver.

Alternate View of EM as BCD

- We showed that given α the M-step minimizes in Θ the function

$$F(\Theta, \alpha) = -\mathbb{E}_{\alpha}[\log p(O, H \mid \Theta)] - \text{entropy}(\alpha).$$

- The E-step minimizes this function in terms of α given Θ.
  - Setting $\alpha_H = p(H \mid O, \Theta)$ minimizes it.
- Note that F is not the NLL, but F and the NLL have same stationary points.
- From this perspective, we can view EM as a block coordinate descent method.
- This perspective is also useful if you want to do approximate E-steps.

Alternate View of EM as KL-Proximal

- Using definitions of expectation and entropy and α in the last slide gives

$$\begin{aligned}
F(\Theta, \alpha) &= -\sum_H p(H \mid O, \theta^t) \log p(O, H \mid \Theta) + \sum_H p(H \mid O, \theta^t) \log p(H \mid O, \theta^t) \\
&= -\sum_H p(H \mid O, \theta^t) \log \frac{p(O, H \mid \theta)}{p(H \mid O, \theta^t)} \\
&= -\sum_H p(H \mid O, \theta^t) \log \frac{p(H \mid O, \theta) p(O \mid \theta)}{p(H \mid O, \theta^t)} \\
&= -\sum_H \log p(O \mid \Theta) - \sum_H p(H \mid O, \theta^t) \log \frac{p(H \mid O, \theta)}{p(H \mid O, \theta^t)} \\
&= NLL(\Theta) + \mathsf{KL}(p(H \mid O, \theta^t) \mid\mid p(H \mid O, \theta)).
\end{aligned}$$

- From this perspective, we can view EM as a "proximal point" method.
  - Classical proximal point method uses $\frac{1}{2}\|\theta^t - \theta\|^2$, EM uses KL divergence.
- From this view we can see that EM doesn't depend on parameterization of Θ.
- If we linearize NLL and we multiply KL term by $1/\alpha_k$ (step-size), we get the natural gradient method.