Notes to Clarify Material on Tuesday Afternoon

Exponential Model: \( P(d|\lambda) = \frac{e^{-\lambda d}}{Z(\lambda)} \)

Data: \( D = \{ d^\mu; \mu = 1:N \} \)

\[ P(D|\lambda) = \prod_{\mu} P(d^\mu|\lambda) \]

Maximum Likelihood (ML): \( \hat{\lambda} = \arg \max P(D|\lambda) \)

\[ \hat{\lambda} = \arg \max \left( \frac{1}{N} \sum P(d^\mu|\lambda) \right) = \frac{1}{\hat{\lambda}} \sum d \]

Probability of the data with best \( \hat{\lambda} \):

\[ P(D|\hat{\lambda}) = \prod_{\mu} P(d^\mu|\hat{\lambda}) \]

Intuitively, if \( P(D|\hat{\lambda}) \) is big — i.e., the data is very probable — then we think that the model fits the data well.

Model Selection:

Suppose we have two models:

\[ P_1(d|\lambda_1) = \frac{1}{Z_1(\lambda_1)} e^{\lambda_1 d} \quad P_2(d|\lambda_2) = \frac{1}{Z_2(\lambda_2)} e^{\lambda_2 d} \]

with different statistics.

Which model is best?

For each model, find: \( \hat{\lambda}_i = \arg \max P_i(D|\hat{\lambda}_i) \)

the best parameter for each model.

Evaluate:

\[ P_1(D|\hat{\lambda}_1) = \prod_{\mu=1}^{N} P_1(d^\mu|\hat{\lambda}_1) \]

and \( P_2(D|\hat{\lambda}_2) = \prod_{\mu=1}^{N} P_2(d^\mu|\hat{\lambda}_2) \)

Select model 1, if \( P_1(D|\hat{\lambda}_1) > P_2(D|\hat{\lambda}_2) \)

model 2, if \( P_2(D|\hat{\lambda}_2) > P_1(D|\hat{\lambda}_1) \)

Model Selection (type I)
Note there is an interpretation of this based on entropy.

\[
H[\mathcal{P}] = - \sum_{d} P(d) \log P(d)
\]

Entropy is a measure of how much information we gain from making an observation \(d\). Suppose \(P(d) = \delta(d-d_s)\),

\[
= \begin{cases} 
0, & \text{if } d \neq d_s \\
\infty, & \text{if } \Delta d = c
\end{cases}
\]

We get no information from observing \(d_s\), because we know that it is the on observation we can get.

\[
H(P_o) = 0 \quad (0 \log 0 = 0, \quad 1 \log 1 = 0)
\]

Suppose \(P_i(d) \sim U(d)\) (uniform distribution)

then we get \(H(P_i) = \log n\)

\[
\text{Result: } \log P(D|\mathcal{I}^L) = \sum_{d} \log P(d|\mathcal{I}^L)
\]

\[
= \sum_{d} P(d|\mathcal{I}^L) \log P(d|\mathcal{I}^L)
\]

\[
\text{Entropy of } P(d|\mathcal{I}^L) = - \sum_{d} P(d|\mathcal{I}^L) \log P(d|\mathcal{I}^L)
\]

\[
= - \sum_{d} P(d|\mathcal{I}^L) \left( \frac{\sum_{\mathcal{I}^L} P(\mathcal{I}^L)}{n} \log \frac{\sum_{\mathcal{I}^L} P(\mathcal{I}^L)}{n} \right)
\]

But \(\sum_{\mathcal{I}^L} P(\mathcal{I}^L) = \frac{n}{m}\), definition of ML

Here \(\log P(D|\mathcal{I}^L) = - n \text{ Entropy}(P(d|\mathcal{I}^L))\)

\(P(D|\mathcal{I}^L) = e^{-n \text{ Entropy}(P(d|\mathcal{I}^L))}\)

So \(P(D|\mathcal{I}^L)\) is big if \(\text{Entropy}(P(d|\mathcal{I}^L))\) is small.

Maximum Likelihood corresponds to minimizing Entropy.

The best model to fit data has lowest energy, hence best ability to predict.
Feature Pursuit:

Make a dictionary \( A = \{ \phi_1, \ldots, \phi_n \} \) of possible features.

**Task**: want to construct a probability model

\[
P(A | \{z_i\}) = \frac{1}{Z(\{z_i\})} e^{\sum_{i=1}^{n} z_i \phi_i(A)}
\]

Wanted to keep model simple — use only a few of the features (also data limitations — later in course).

Want \( \Lambda_i = 0 \) for most \( i \).

Two tasks:

1. **Selection** — which features to use (i.e., to have \( \Lambda_i \neq 0 \))

2. **Weighting** — how to weight features and assign \( \Lambda_i \)?

This is a hard search problem (easier for discriminative learning — later in the course).

**Strategy**: Feature Pursuit. \( \Rightarrow \) Della Pietra, Lafferty.

1. Find best model with one feature only

Calculate: \( \hat{i} \) s.t.

\[
P_i(D | \{z_i\}) \geq P_i(D | \{z_i\})
\]

Here \( P_i(D | \{z_i\}) = \frac{1}{Z(\{z_i\})} e^{\sum_{i=1}^{n} z_i \phi_i(A)} \)

This selects feature \( \phi_i \) and assigns it weight \( \hat{\Lambda}_i \).

2. Next add another feature/statistic:

Consider all models of form:

\[
P_{i,j}(D | \{z_i, z_j\}) = \frac{1}{Z(2 \{z_i, z_j\})} e^{\sum_{i=1}^{n} z_i \phi_i(A) + \sum_{j=1}^{n} z_j \phi_j(A)}
\]
Select the second feature \( \hat{\gamma} \) by finding

\[
P_{\hat{\gamma}}(D|\hat{\alpha}_i, \hat{\beta}_3) > P_{\hat{\gamma}}(D|\hat{\alpha}_i, \hat{\beta}_3)
\]

for all \( \hat{\gamma} \).

Proceed to select and weight the third, fourth, fifth, \ldots features and weight them.

When to stop?

Adding a new feature allows the model to fit the data better.

- i.e. \( P_{\hat{\gamma}}(D|\hat{\alpha}_i, \hat{\beta}_3) > P_{\hat{\gamma}}(D|\hat{\alpha}_i) \)

because the model with two features has more flexibility and can fit the data better.

Example: \( y = a + bx \) or to curve \( y = a + bx + cx^2 \) easier, more flexibility.

Stop if increase by adding a new feature falls below a threshold:

\[ \text{if } P_{\hat{\gamma}}(D|\hat{\alpha}_i, \hat{\beta}_3) \leq P_{\hat{\gamma}}(D|\hat{\alpha}_i) + T \]

Note: A more advanced form of model selection will avoid this. -> Occam's Razor (often impractical).

Required: Combining \( P_i(D) = \sum_{\lambda_i} P_i(D|\lambda_i) \)

\[
\frac{1}{D} P_i(D) = \sum_{\lambda_i} \frac{1}{D} P_i(D|\lambda_i)
\]

\[
\frac{1}{D} P_i(D) = 1
\]
Expectation-Maximization:

\[ P(d, h | \lambda) = \frac{1}{Z(\lambda)} e^{\lambda^T \phi(d, h)} \]

Do ML on \( P(d | \lambda) = \frac{1}{Z} \sum_h P(d, h | \lambda) \)
to estimate \( \lambda \)

Maximize : \( -\log P(d | \lambda) \) with respect to \( \lambda \)

Add a new variable \( q_h(h) \), a distribution over the hidden variables \( \frac{1}{Z} q(h) \).

New \( P(h | d, \lambda) \) is the probability of the hidden variable \( h \), if we know the parameter \( \lambda \).

Formally \( P(h | d, \lambda) = \frac{P(h, d | \lambda)}{\sum_h P(h, d | \lambda)} \) \( \frac{1}{Z} \) \( P(h | d, \lambda) \)

Calculating \( \frac{1}{Z} P(h | d, \lambda) \)

May be difficult - see later.

Want \( q(h) \) to be close to \( P(h | d, \lambda) \)

Kullback-Leibler

\[ \sum_h q(h) \log \frac{P(h | d, \lambda)}{\frac{1}{Z} \frac{P(h | d, \lambda)}{P(d, h | \lambda)}} > 0 \]

\[ \Rightarrow \frac{1}{Z} \frac{P(h | d, \lambda)}{P(d, h | \lambda)} \frac{1}{Z} \frac{P(h | d, \lambda)}{P(d, h | \lambda)} \]

Define:

\[ F[\lambda, q(.)] = -\log P(d | \lambda) + \sum_h q(h) \log q(h) \]

This is a function of \( \lambda \) and \( q(.) \).

Its global minimum occurs at:

\[ \hat{\lambda} = \arg \min_{\lambda} (-\log P(d | \lambda)) \]

\[ \Rightarrow \frac{d}{d\lambda} \sum_h q(h) \log q(h) = 0 \]

and at \( q(h) = P(h | d, \hat{\lambda}) \) - makes second term 0.
This is non-convex. Will have several minima in general.

Minimize by coordinate descent.

1. At state \( z^t \),
   Solve for \( q^t(h) = \arg\min_{q(h)} F[z^t, q(h)] \)
   Solution: \( q^t(h) = \frac{p(h, d, z^t)}{\sum_k p(h, d, z^t_k)} \)
   (require computing \( \frac{\sum_k p(h, d, z^t_k)}{\sum_k p(h, d, z^t_k)} \))

2. At state \( q^t(h) \)
   Compute \( z^{t+1} = \arg\min_{z} F[z, q^t] \)
   Solution: \( \sum_h q^t(h) \phi(d, h) = \sum_h \phi(d, h) p(h, d | z^t) \)
   (data statistic) \( h, d \) and expected \( h \)
   w.r.t. \( q^t(h) \) (model statistic)

Repeat steps.
Performance depends on the initial condition \( z^0 \).

Previous notes (Tuesday) gave results for the extended case when we have data \( D = \{d, \mu: \mu \in \mathcal{W}\} \).
Then replace \( F[z, q(h, \lambda)] \) by
\[
- \sum_n \log p(\lambda_n | x) + \sum_{n, w} \sum_{\mu} q(\mu_n | y_n) \log \frac{q(\mu_n | y_n)}{p(\mu_n | d_n, \mu, \lambda)}
\]