(1) Lecture 4 Spring 2013

Note Title 10/7/2006

Learning parametric probability models. This lecture deals with distributions like $P(\underline{x})$ or $P(\underline{x},y)$. Later lectures will learn $P(y|\underline{x})$, the regression problem.

Parameterized model $P(x|\theta)$ ← model parameter

E.G. Gaussian distribution.

$$P(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi}\,\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \qquad \theta=(\mu,\sigma).$$

Assume that data is idependent identically distributed (iid).

(product for independence).

$$P(x_1,..,x_N|\theta) = \prod_{i=1}^{N} P(x_i|\theta).$$

Choose: $\hat{\theta} = \text{ARG MAX } P(x_1..x_N|\theta) = \text{ARG } \underset{\theta}{\text{MAX}} \log P(x_1..x_N|\theta)$

Hence $p(x_1,..x_N|\hat{\theta}) \geq p(x_1\,.\,x_N|\theta)$, for all θ

$$\log P(x_1...x_N|\mu,\sigma) = \sum_{i=1}^{N} \log P(x_i|\mu,\sigma)$$

$$= -\sum_{i=1}^{N} \frac{(x_i-\mu)^2}{2\sigma^2} - \sum_{i=1}^{N} \log \sqrt{2\pi}\,\sigma.$$

Differentiate w.r.t μ,σ gives

$$\frac{\partial}{\partial\mu} \log P(x_1..x_N|\mu,\sigma) = \frac{1}{\sigma^2}\sum_{i=1}^{N}(x_i-\mu).$$

$$\frac{\partial}{\partial\sigma} \log P(x_1..x_N|\mu,\sigma) = \frac{1}{\sigma^3}\sum_{i=1}^{N}(x_i-\mu)^2 - \frac{N}{\sigma}.$$

(2) Spring 2013

Maxima occurs at

$$\hat{\mu} = \frac{1}{N}\sum_{i=1}^{N} x_i$$

$$\hat{\sigma}_2 = \frac{1}{N}\sum_{i=1}^{N} (x_i - \hat{\mu})^2.$$

Note: The Gaussian is a special case. It is often impossible to solve $\frac{\partial}{\partial\theta}\log P(x_1..x_N|\theta)=0$ analytically. An algorithm is required. (see later)

Now $\hat{\theta} = \text{ARG MAX } P(x_1..x_N|\theta)$ is the ML estimator.

We could use the MAP estimator

$$\hat{\theta} = \text{ARG MAX } P(\theta|x_1,..x_N)$$

$$P(\theta|x_1..x_N) = \frac{P(x_1 \quad x_N|\theta)\,P(\theta)}{P(x_1..x_N)}$$

$$\log P(\theta|x_1..x_N) = \underbrace{\sum_{i=1}^{N} \log P(x_i|\theta)}_{N \text{ terms}} + \underbrace{\log P(\theta)}_{1 \text{ term}} - \log P(x_1 \,.x_N)$$

Note if N is large then the prior $P(\theta)$ usually has little effect.

We could use other estimators. We could formulate this in terms of Bayes risk and use a loss function which penalizes errors in θ.

But this lecture will stick with ML estimation.

Exponential Distributions

$$P(\underline{x}|\underline{\lambda}) = \frac{1}{Z[\underline{\lambda}]}\, e^{\underline{\lambda}\cdot\underline{\phi}(\underline{x})}$$

normalization factor.

$\underline{\lambda}$ – parameters $\quad \underline{\lambda} = (\lambda_1, \lambda_2, \ldots, \lambda_M)$

$\underline{\phi}(\underline{x})$ – statistics. $\quad \underline{\phi}(\underline{x}) = (\phi_1(\underline{x}), \phi_2(\underline{x}), \ldots, \phi_M(\underline{x}))$.

<u>Almost</u> every named distribution can be expressed as an exponential distribution (particularly if we allow missing/hidden variables).

For Gaussian in 1-dimension.

<u>write</u> $\quad \underline{\phi}(x) = (x, x^2) \quad \underline{\lambda} = \lambda_1, \lambda_2$

$$P(x|\lambda) = \frac{1}{Z[\underline{\lambda}]}\, e^{\lambda_1 x + \lambda_2 x^2} \quad \text{compare to} \quad \frac{1}{\sqrt{2\pi}\,\sigma}\, e^{-\frac{(x-\mu)^2}{2\sigma^2}}.$$

<u>Translation</u>

$$\left\{\begin{aligned} \lambda_2 &= -1/2\sigma^2 \\ \lambda_1 &= \mu/\sigma^2. \\ Z[\underline{\lambda}] &= \sqrt{2\pi}\,\sigma\, e^{\mu^2/2\sigma^2} \end{aligned}\right.$$

Similar translation into exponential distributions can be made for Poisson, Beta, Dirichlet ~ most (all) distributions you have been taught.

(4)

Spring 2013.

Learning an Exponential Distribution

You can learn them by Maximum Likelihood,

Examples:

$$P(\{\underline{x}_1, \underline{x}_2, .., \underline{x}_N\} | \underline{\lambda}) = \prod_{\mu=1}^{N} \frac{e^{\underline{\lambda} \cdot \underline{\phi}(\underline{x}_\mu)}}{Z[\underline{\lambda}]}.$$

Maximize w.r.t $\underline{\lambda}$ ||

This has a very nice form, which occurs because the exponential distribution depends on the data $\underline{x}$ only in terms of the function $\underline{\phi}(\underline{x})$ — the sufficient statistics.

Note: Important factor. The normalization term $Z[\underline{\lambda}]$ is a function of $\underline{\lambda}$, $Z[\underline{\lambda}] = \sum_{\underline{x}} e^{\underline{\lambda} \cdot \underline{\phi}(\underline{x})}$

Claim: $\frac{\partial \log Z[\underline{\lambda}]}{\partial \underline{\lambda}} = \sum_{\underline{x}} \underline{\phi}(\underline{x}) P(\underline{x} | \underline{\lambda})$ expected value of the statistics $\underline{\phi}(\underline{x})$ w.r.t. $P(\underline{x} | \underline{\lambda})$

Proof. $\frac{\partial \log Z[\underline{\lambda}]}{\partial \underline{\lambda}} = \frac{1}{Z[\underline{\lambda}]} \frac{\partial Z[\underline{\lambda}]}{\partial \underline{\lambda}} = \frac{1}{Z[\underline{\lambda}]} \sum_{\underline{x}} \underline{\phi}(\underline{x}) e^{\underline{\lambda} \cdot \underline{\phi}(\underline{x})} = \sum_{\underline{x}} \underline{\phi}(\underline{x}) P(\underline{x} | \underline{\lambda})$.

(5) Spring 2013.

Claim: For exponential distribution, ML corresponds to finding the value of $\underline{\lambda}$ s.t. the model statistics are equal to the data statistics

solve $\sum_{\underline{x}} \underline{\phi}(\underline{x}) P(\underline{x}|\underline{\lambda}) = \frac{1}{N}\sum_{i=1}^{N} \underline{\phi}(\underline{x}_i)$.

E.G. For Gaussian.

model statistics $\int d\underline{x}\ \underline{x}\ \frac{1}{(2\pi)^{n/2}|\det \underline{\underline{\Sigma}}|} e^{-\frac{1}{2}(\underline{x}-\underline{\mu})^T \underline{\underline{\Sigma}}^{-1}(\underline{x}-\underline{\mu})} = \underline{\mu}$

data statistic $\frac{1}{N}\sum_{i=1}^{N} \underline{x}_i$.

Proof. ML minimizes $-\log \prod_{i=1}^{N} P(\underline{x}_i|\underline{\lambda}) = -\sum_{i=1}^{N} \log P(\underline{x}_i|\underline{\lambda})$

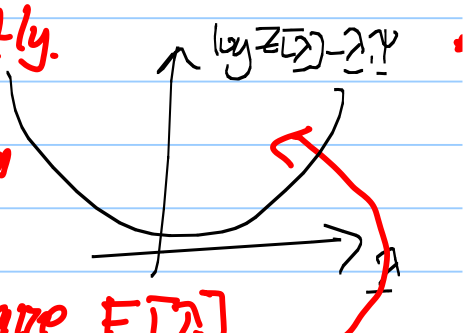
For exponential distributions this is

$$F[\underline{\lambda}] = N \log Z[\underline{\lambda}] - \sum_{i=1}^{N} \underline{\lambda}\cdot\underline{\phi}(\underline{x}_i)$$

Differentiating w.r.t. $\underline{\lambda}$

$$\frac{\partial F}{\partial \underline{\lambda}} = N\sum_{\underline{x}} \underline{\phi}(\underline{x}) P(\underline{x}|\underline{\lambda}) - \sum_{i=1}^{N} \underline{\lambda}.\underline{\phi}(\underline{x}_i)$$ result follows.

Note: for some exponential distributions it is possible compute $\sum_{\underline{x}} \underline{\phi}(\underline{x}) P(\underline{x}|\underline{\lambda})$ analytically as a function of $\underline{\lambda}$ (e.g. Gaussian). and hence solve ML directly.


$\log Z[\underline{\lambda}] - \underline{\lambda}.\underline{\Psi}$
$\underline{\lambda}$

But for other exponential distribution we cannot compute $\sum_{\underline{x}} \underline{\phi}(\underline{x}) P(\underline{x}|\underline{\lambda})$.

Instead we use an algorithm to minimize $F[\underline{\lambda}]$ w.r.t. $\underline{\lambda}$. Fortunately $F[\underline{\lambda}]$ is a convex function of $\underline{\lambda}$ and hence has only a single minimum (proof $\frac{\partial^2 F}{\partial\underline{\lambda}\,\partial\underline{\lambda}}$ is positive definite)

(6) Spring 2013

Algorithms for minimizing $F[\underline{\lambda}]$ include:

(i) Steepest Descent:

$$\underline{\lambda}^{t+1} = \underline{\lambda}^{t} - \Delta \frac{\partial F}{\partial \underline{\lambda}}$$

Δ is a small constant.

→ Newton Raphson.

$$\underline{\lambda}^{t+1} = \underline{\lambda}^{t} - \Delta \left\{ \sum_{\underline{x}} \underline{\phi}(\underline{x}) P(\underline{x}|\underline{\lambda}^{t}) - \frac{1}{N}\sum_{i=1}^{N} \underline{\phi}(\underline{x}_i) \right\}$$

(ii) Generalized Iterative Scaling (GIS)

$$\underline{\lambda}^{t+1} = \underline{\lambda}^{t} - \log \sum_{\underline{x}} \underline{\phi}(\underline{x}) P(\underline{x}|\underline{\lambda}^{t}) + \log \frac{1}{N}\sum_{i=1}^{N} \underline{\phi}(\underline{x}_i)$$

(Note $\log \underline{\psi} = (\log \psi_1, \log \psi_2 \ldots, \log \psi_m)$, where $\underline{\psi} = (\psi_1 \ldots \psi_m)$)
vector | all components are positive. |

Comment: Steepest descent requires specifying a step size Δ. If Δ is too big, algorithm fails to converge. If Δ is too small, algorithm converges slowly.
Both algorithms guaranteed to converge to the correct solution (provided Δ is well-chosen).

Note: both algorithms require computing
$\sum_{\underline{x}} \underline{\phi}(\underline{x}) P(\underline{x}|\underline{\lambda}^{t})$ at each stage t of the algorithm

Computing this can be difficult to perform numerically for some distributions. If so, stochastic sampling methods like Markov Chain Monte Carlo (MCMC) may be used.

(7) Spring 2013

The theory of ML estimation in the statistics literature assumes that $P(\underline{x}|\underline{\lambda})$ is the correct model for the data.

We now discuss how to justify ML as an approximation if the data is generated by a different distribution.

First define the Kullback-Leibler divergence between two distributions $P(\underline{x}|\underline{\lambda})$ and $f(\underline{x})$

$$D(f\|p) = \sum_{\underline{x}} f(\underline{x}) \log \left\{ \frac{f(\underline{x})}{P(\underline{x}|\underline{\lambda})} \right\}.$$

Properties: $D(f\|p) \geq 0$, $D(f\|p) = 0$ only if $f(\underline{x}) = P(\underline{x}|\underline{\lambda})$

i.e. If $D(f\|p)$ are small, then distributions $f(\underline{x})$ and $P(\underline{x}|\underline{\lambda})$ are similar. If it is large, then they are not.

We can express:

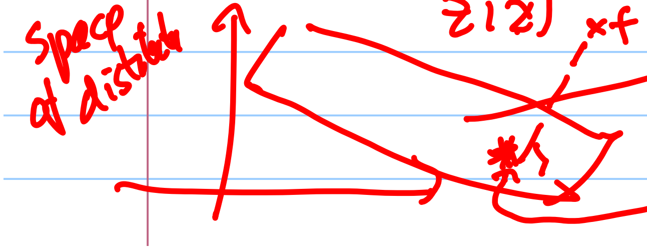
$$D(f\|p) = \underbrace{\sum_{\underline{x}} f(\underline{x}) \log f(\underline{x})}_{\text{indep of } \lambda} - \underbrace{\sum_{\underline{x}} f(\underline{x}) \log P(\underline{x}|\underline{\lambda})}_{\text{function of } \underline{\lambda}}$$

Hence minimize $D(f\|p)$ w.r.t. $\underline{\lambda}$ corresponds to minimizing $-\sum_{\underline{x}} f(\underline{x}) \log P(\underline{x}|\underline{\lambda})$

Geometric Interpretation : Information Geometry (Amari).

$$P(\underline{x}|\underline{\theta}) = \frac{e^{\underline{\lambda}\cdot\underline{\phi}(\underline{x})}}{Z[\underline{\lambda}]}$$ define a sub-manifold of distributions

$\underline{\lambda}$'s coordinate in the manifold.



Minimizing $D(f\|p)$ w.r.t. λ find distribution in submanifold closest to f. Best approximation.

(8) Relation to ML. Spring 2013.

Set $f(\underline{x}) = \frac{1}{N}\sum_{i=1}^{N} I(\underline{x}=\underline{x}_i)$ ← Indicator Function

This is the empirical distribution of the data $\{\underline{x}_i : i=1,\ldots,N\}$

(This is a special case of Parzen windows, later lecture.)

In this, minimizing the KL divergence corresponds to minimizing:

$$-\sum_{\underline{x}} f(\underline{x}) \log P(\underline{x}|\underline{\lambda}) = -\sum_{\underline{x}} \frac{1}{N}\sum_{i=1}^{N} I(\underline{x}=\underline{x}_i) \log P(\underline{x}|\underline{\lambda})$$

$$= -\frac{1}{N}\sum_{i=1}^{N} \log P(\underline{x}_i|\underline{\lambda}).$$

This proves the → same criteria as ML!

Claim: ML estimation of $\underline{\lambda}$ is equivalent to minimize $D(f \| P(\underline{x}|\underline{\lambda}))$ w.r.t. $\underline{\lambda}$, where $f(\underline{x})$ is the empirical distribution of the data.

Hence we can justify ML (for exponentials) as obtaining the distribution of form $\frac{e^{\underline{\lambda}\cdot\underline{\phi}(\underline{x})}}{Z[\underline{\lambda}]}$ which best approximates the data.

This also motivates the idea of model pursuit.

(1) Start by doing ML on an exponential distribution with statistic $\underline{\phi}_1(\underline{x})$. Get best approximation

(2) Get a better approximation by using more complex statistics → e.g. $\underline{\phi}_1(\underline{x}), \underline{\phi}_2(\underline{x})$ with parameters $\underline{\lambda}_1, \underline{\lambda}_2$

(3) Proceed by using increasingly complex stats. Beyond scope of course.

(9) Spring 2023

Another perspective on learning

Where do exponential distributions come from?

Jaynes (wiki) claim they come from a maximum entropy principle.

We have data $(\underline{x}_1 \ldots \underline{x}_N)$.

We have statistics $\underline{\phi}(\underline{x})$ of the data.

How to justify a distribution like $P(\underline{x}) = \frac{e^{\underline{\lambda}\cdot\underline{\phi}(\underline{x})}}{Z[\underline{\lambda}]}$?

And how to justify using ML to get $\underline{\lambda}$?

$$\mathcal{H}[p] = -\sum_{\underline{x}} P(\underline{x}) \log P(\underline{x})$$ Entropy of Distribution.

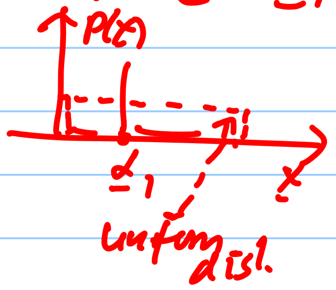
It is a measure of the amount of information that an observer expects to obtain by observing a sample $\underline{x}$ from a distribution $P(\underline{x})$

Info from sample $-\log P(\underline{x})$

Expected info $= -\sum_{\underline{x}} P(\underline{x}) \log P(\underline{x})$ } Shannon: Information Theory

Example: Suppose $\underline{x}$ can take M states $\underline{\alpha}_1, \ldots \underline{\alpha}_M$

If $P(\underline{x} = \underline{\alpha}_1) = 1$, $P(\underline{x} = \underline{\alpha}_i) = 0$, $i = 2, \ldots M$.



Then $\mathcal{H}[P] = 0$ (Note $0 \log 0 = 0$, $1 \log 1 = 0$)

So no. info is gained by observing the sample, because we know it can only be $\underline{\alpha}_1$ (other possibilities can't happen).

Alternatively. Suppose $P(x = \underline{\alpha}_i) = 1/M$, for all i. Uniform Distr.

$\mathcal{H}[P] = \log M$.

(10) Maximum Entropy Principle

Spring 2013.

Given statistics $\underline{\phi}(\underline{x})$ with observed value $\underline{\psi}$, choose the distribution $P(\underline{x})$ to maximize the entropy subject to constraints

$$-\sum_{\underline{x}} p(\underline{x})\log P(\underline{x}) + \overset{\text{lagrange multiplier}}{\mu}\overbrace{\left\{\sum_{\underline{x}} p(\underline{x}) - 1\right\}}^{\text{constraint}}$$

$$+ \underbrace{\underline{\lambda}}_{\text{lagrange multiplier}} \cdot \overbrace{\left\{\sum_{\underline{x}} p(\underline{x})\underline{\phi}(\underline{x}) - \underline{\psi}\right\}}^{\text{constraint}}$$

$$\frac{\delta}{\delta p(\underline{x})} \qquad -\log p(\underline{x}) - 1 + \mu + \underline{\lambda}\cdot\underline{\phi}(\underline{x}) = 0$$

solution. $\quad p(\underline{x}|\underline{\lambda}) = \dfrac{e^{\underline{\lambda}\cdot\underline{\phi}(\underline{x})}}{Z[\underline{\lambda}]}$

where $\underline{\lambda}, Z[\underline{\lambda}]$ are choosen to satisfy the Constraints:

$$\sum_{\underline{x}} p(\underline{x}) = 1, \quad \Rightarrow \quad Z[\underline{\lambda}] = \sum_{\underline{x}} e^{\underline{\lambda}\cdot\underline{\phi}(\underline{x})}$$

$$\sum_{\underline{x}} p(\underline{x})\underline{\phi}(\underline{x}) = \underline{\psi}, \quad \Rightarrow \quad \underline{\lambda} \text{ is choosen s.t.}$$

$$\sum_{\underline{x}} p(\underline{x}|\underline{\lambda})\,\underline{\phi}(\underline{x}) = \underline{\lambda}$$

The maximum entropy principle recovers exponential distribution!

(11) <u>Entropy and ML.</u> Spring 2023.

Suppose we have data $\{\underline{x}_i : i = 1 \text{ to } N\}$, and fit a probability distribution $P(\underline{x}|\underline{\lambda})$ by ML — to get parameter $\hat{\underline{\lambda}}$

The <u>prob of the data</u> — using $P(\underline{x}|\hat{\underline{\lambda}})$ ← best estimate of $\underline{\lambda}$.

is $\prod_{i=1}^{N} P(\underline{x}_i|\hat{\underline{\lambda}}) = \exp\{\hat{\underline{\lambda}} \cdot \sum_{i=1}^{N} \underline{\phi}(\underline{x}_i) - N \log Z[\hat{\underline{\lambda}}]\}$,

The <u>entropy</u> of $P(\underline{x}|\hat{\underline{\lambda}})$ is

$$-\sum_{\underline{x}} P(\underline{x}|\hat{\underline{\lambda}}) \log P(\underline{x}|\hat{\underline{\lambda}}) = \log Z[\hat{\underline{\lambda}}] - \sum_{\underline{x}} \hat{\underline{\lambda}} \cdot \underline{\phi}(\underline{x}) P(\underline{x}|\hat{\underline{\lambda}})$$

$$= \log Z[\hat{\underline{\lambda}}] - \frac{1}{N} \sum_{i=1}^{N} \hat{\underline{\lambda}} \cdot \underline{\phi}(\underline{x}_i)$$

(By defn. of $\hat{\underline{\lambda}}$)

<u>Hence</u> Prob of Data given $P(\underline{x}|\hat{\underline{\lambda}})$

$$= \exp\{-N \mathcal{H}[P(\underline{x}|\hat{\underline{\lambda}})]\} \quad //$$

So if the entropy of $P(\underline{x}|\hat{\underline{\lambda}})$ is small, then it does not describe the data well — it cannot predict it and there is a lot of uncertainty.

<u>Two</u> Related measures of the model. Its entropy. And the prob. of the data given the model.

Motivates Shannon search for the entropy of English. An example of model pursuit

<u>Shannon</u> starts with unary statistics — frequency of letters — fit to data (English text) estimated entropy.

then Shannon used more complex statistics — pairwise frequencies — entropy decreased (better fit) and so on. Compare to human entropy what is the next letter to "rgth."?

(12)

Spring 2023.

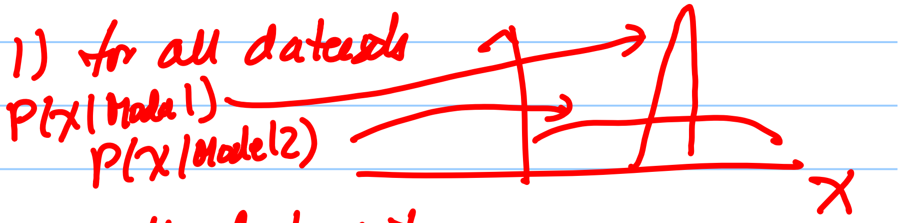
Which Models to Use?
How to select between them?

Suppose we have two models $P(\underline{x}|Model 1)$, $P(\underline{x}|Model 2)$
e.g. $P(\underline{x}|Model 1)$ uses stats $\underline{\phi}_1(\underline{x})$, parameter $\underline{\lambda}_1$
$P(\underline{x}|Model 2)$ " " $\underline{\phi}_2(\underline{x})$, parameter $\underline{\lambda}_2$.

Let $\chi = \{\underline{x}_i : i \text{ to } N\}$

Compute $P(\chi|Model 1)$ and $P(\chi|Model 2)$
Select Model 1 if $P(\chi|Model 1) > P(\chi|Model 2)$

Advantage – Occam Factor – this criterion favors simpler more specific models because the distribution must obey $\sum_{\chi} P(\chi|Model 1)$ for all datasets



Model 2 does okay on all datasets, but Model 1 does very well on some datasets and very badly on others. So if Model 1 does well on your dataset, then you should use it.

But there are complications. Model 1 corresponds to choice of statistics $\underline{\phi}_1(\underline{x})$

$$P(\chi|Model 1) = \sum_{\underline{\lambda}} P(\chi|\underline{\lambda}) P(\underline{\lambda}|Model 1)$$

$$P(\underline{x}|\underline{\lambda}) = \frac{e^{\underline{\lambda}\cdot\underline{\phi}(\underline{x})}}{Z[\underline{\lambda}]}$$

$$P(\chi|\underline{\lambda}) = \prod_{i=1}^{N} P(\underline{x}_i|\underline{\lambda}).$$

Usually difficult, or impossible, to perform $\sum_{\underline{\lambda}}$.

(13) Spring 2017

So try to approximate
$P(X \mid \text{Model 1})$ by $\max_{\underline{\lambda}} P(X \mid \underline{\lambda})$ (assume $P(\underline{\lambda} \mid \text{Model 1})$ is uniform)

gives $P(X \mid \text{Model 1}) \approx P(X \mid \hat{\underline{\lambda}})$ → the ML prob, we discussed earlier, which relates to the entropy.

But now we no longer have $\sum_{\underline{X}} P(X \mid \text{Model 1}) = 1$ (even if we include the prior $P(\hat{\underline{\lambda}} \mid \text{Model})$)

So Occam factor does not apply. //

Need to penalize more complex models.

→ e.g. Compare $-\log P(X \mid \hat{\underline{\lambda}}_1)$ model $P(\underline{x} \mid \underline{\lambda}) = e^{\hat{\underline{\lambda}}_1 \cdot \underline{\phi}_1(\underline{x})} / Z[\underline{\lambda}_1]$
with $-\log P(X \mid \hat{\underline{\lambda}}_2)$ model $\dfrac{e^{\hat{\underline{\lambda}}_2 \cdot \underline{\phi}_2(\underline{x})}}{Z[\underline{\lambda}_2]}$

Suppose model 2 has more parameters
→ e.g. $\underline{\phi}_1(\underline{x}) = (\phi^1_1(\underline{x}) \dots \phi^1_{10}(\underline{x}))$, $\underline{\lambda}_1 = (\lambda^1_1, \dots \lambda^1_{10})$
$\underline{\phi}_2(\underline{x}) = (\phi^2_1(\underline{x}), \dots, \phi^2_{100}(\underline{x}))$, $\underline{\lambda}_2 = (\lambda^2_1, \dots, \lambda^2_{100})$

Model 2 has 100 parameters, Model 1 has 10
So Model 2 can adjust to data better.

Penalize more complex model → AIC, BIC
Each model pays penalty $k_1 n$, or $k_2 \log n$
where n is the no. of parameters.

Other criteria → Minimum Description Length (MDL)

(14) Spring 2013

Training and Testing.

Critical issue is — how much data do you need to learn a distribution?

There is no perfect answer. A rule of thumb is that you need k × no. of parameter of the distribution, where k = 5 to 10.

In practice, train (learn) the model on a training dataset. Test it on a second dataset. If performance (e.g. Bayes Risk, ROC curves, etc) is the same on both — then you have learned or generalized.

If performance is good on the training set but bad on the testing set — then you have only memorized the training set.