Lecture 4

Spring 2014

Note Title 4/9/2014

TOPICS:

(I) Learning probability distributions by Maximum Likelihood (ML)

(II) Exponential distributions, Sufficient Statistics, and ML learning.

(III) Kullback-Leibler divergence, learning "approximate" distributions.

(IV) Advanced Topics,
Maximum Entropy Principle.
Model Pursuit.
Model Selection.

(2) Maximum Likelihood (ML) learning of parametric probability distributions. (non-parametric, later lecture)

Note Title

Learn distributions like $P(x|y)$ likelihood function
or $P(y)$. prior.

For simplicity, we just study learning $P(x)$.
(Note: learning $P(y|x)$ – regression – later in the course.)

Parametric Distribution → $P(x|\theta)$, θ parameter

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

Data $\mathcal{X}_N = \{x_i : i = 1 \text{ to } N\}$

Assume data is i.i.d. from unknown $P(x)$

Maximum Likelihood (ML):

$$P(\mathcal{X}_N|\theta) = \prod_{i=1}^{N} P(x_i|\theta) \quad \text{(i.i.d. assumption)}$$

Set $\hat{\theta} = \underset{\theta}{\text{ARG MAX}}\, P(\mathcal{X}_N|\theta)$

$P(\mathcal{X}_N|\hat{\theta}) \geq P(\mathcal{X}_N|\theta)$ for any θ

This is equivalent to $\arg\max_\theta \log P(\mathcal{X}_N|\theta)$ (log = logarithm)

$$\hat{\theta} = \underset{\theta}{\text{ARG MAX}} \left\{ \sum_{i=1}^{N} \log P(x_i|\theta) \right\}$$

For Gaussian: $\sum_{i=1}^{N} \log P(x_i|\mu,\sigma) = -\sum_{i=1}^{N} \frac{(x_i-\mu)^2}{2\sigma^2} - \sum_{i=1}^{N} \log\sqrt{2\pi}\sigma$

To estimate $(\hat{\mu},\hat{\sigma})$ differentiate w.r.t. μ,σ (i.e. maximize $\log P(\mathcal{X}_N|\mu,\sigma)$ w.r.t. μ,σ.)

$$\frac{\partial}{\partial\mu} \log P(\mathcal{X}_N|\mu,\sigma) = \frac{1}{\sigma^2} \sum_{i=1}^{N} (x_i - \mu)$$

$$\frac{\partial}{\partial\sigma} \log P(\mathcal{X}_N|\mu,\sigma) = \frac{1}{\sigma^3} \sum_{i=1}^{N} (x_i-\mu)^2 - N/\sigma$$

Hence solution

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

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

(3) Note: the Gaussian is a special case because it gives a simple analytic formula for. $\hat{\mu}, \hat{\sigma}^2$.

This illustrates the ML estimator.

Note: What about MAP?

Why not use a prior $P(\theta)$?

$$\hat{\theta}_{MAP} = \arg\max_{\theta} \frac{P(\chi_N|\theta)P(\theta)}{P(\chi_N)} \quad \leftarrow \text{indep of } \theta$$

$$= \arg\max_{\theta} \{ \log P(\chi_N|\theta) + \log P(\theta) \}$$

$$= \arg\max_{\theta} \{ \sum_{i=1}^{N} \log P(x_i|\theta) + \log P(\theta) \}$$

N data terms; 1 prior term.

If N is large, then the prior will have little effect (except for special cases).

Note: we can also use loss functions and all the machinery of Bayes Decision Theory.

In practice, loss functions are often not used when learning probability distributions – but they are used for learning classifiers (later in course)

(4) Topic II: Exponential Distributions

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

Almost every named distribution can be expressed as an exponential distribution.

For Gaussian in 1-dimension.

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

$$P(x|\lambda) = \frac{1}{Z[\underline{\lambda}]} e^{\lambda_1 x + \lambda_2 x^2}$$

compare to $\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$.

Translation

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

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

(5)

Learning an Exponential Distribution

You can learn them by Maximum Likelihood,

Examples:

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

Maximize wrt $\underline{\lambda}$ ||

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

(6)

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

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

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

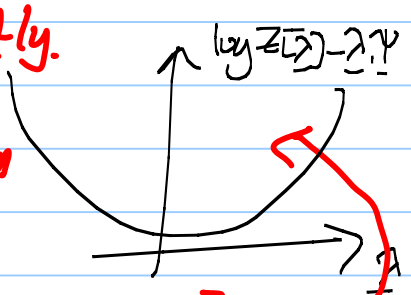
For exponential distributions this is

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

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

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



But for other exponential distributions 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)

(7)

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

(i) Steepest Descent:

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

Δ is a small constant.

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

(ii) Generalized Iterative Scaling (GIS) $\rightsquigarrow$ Newton Raphson.

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

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 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.

(8) Examples of learning Exponential. $P(x|\lambda) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$

Gaussian: statistics $\underline{\phi}(x) = (x, x^2)$, N.B. $\underline{\phi}(\underline{x}) = (\underline{x}, \underline{x}\,\underline{x}^T)$ for Gaussian in N-dim. $\underline{x}$ N-D vector

$$\sum_x P(x|\lambda)(x, x^2) = \frac{1}{N}\sum_{i=1}^{N}(x_i, x_i^2)$$

L.H.S. $\int P(x|\lambda)\, x = \mu$

$\int P(x|\lambda)\, x^2 = \mu^2 + \sigma^2$,

Hence $\hat{\mu} = \frac{1}{N}\sum_{i=1}^{N} x_i$, $\hat{\mu}^2 + \hat{\sigma}^2 = \frac{1}{N}\sum_{i=1}^{N} x_i^2$, so $\hat{\sigma}^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \hat{\mu})^2$

Note: Really should be $\int d\underline{x}\, P(\underline{x}|\underline{\lambda})$ for Gaussian.

Letter Example: $x \in A = \{a, b, c, d, e, \ldots, y, z\}$

statistic $\underline{\phi}(x) = (\delta_{x,a}, \delta_{x,b}, \ldots, \delta_{x,z})$.

ie. $\underline{\phi}(c) = (0, 0, 1, 0, 0 \ldots 0)$

$\delta_{x,a} = 1$, if $x = a$
$= 0$, otherwise

(9) Dataset of letters $x_N = \{x_1, \ldots, x_N\}$

$$\frac{1}{N}\sum_{i=1}^{N} \underline{\phi}(x_i) = \left(\frac{\#a's}{N}, \frac{\#b's}{N}, \ldots, \frac{\#z's}{N}\right)$$

where $\#a's = \sum_{i=1}^{N} s_{x_i,a}$ = no. of a's in dataset.

$$\underline{\lambda} = (\lambda_a, \ldots, \lambda_z)$$

$$P(x|\underline{\lambda}) = \frac{1}{Z[\underline{\lambda}]} e^{\lambda_a s_{x,a} + \ldots + \lambda_z s_{z,a}}$$

$$\sum_x P(x|\underline{\lambda})\, s_{x,a} = \frac{1}{Z[\underline{\lambda}]} e^{\lambda_a}$$

Note: $P(x=a|\underline{\lambda}) = \frac{e^{\lambda_a}}{Z[\underline{\lambda}]}$

$$Z[\underline{\lambda}] = e^{\lambda_a} + \ldots + e^{\lambda_z}$$

Hence solution

$$\frac{1}{Z[\underline{\lambda}]}(e^{\lambda_a}, e^{\lambda_b}, \ldots, e^{\lambda_z}) = \left(\frac{\#a's}{N}, \frac{\#b's}{N}, \ldots \frac{\#z's}{N}\right)$$

Hence $\hat{\lambda}_a = \log \#a's - \log N$, $\hat{\lambda}_b = \log \#b's - \log N, \ldots, \hat{\lambda}_z = \log \#z's - \log N$.

$Z[\underline{\hat{\lambda}}] = \#a's + \#b's + \ldots + \#z's = 1$. //

(16) Topic III: Kullback-Leibler & Approximate Distributions

The theory of ML estimation in the statistics literature assumes that $P(x|\lambda)$ 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 $P(x|\lambda)$ and $f(x)$

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

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

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

We can express:

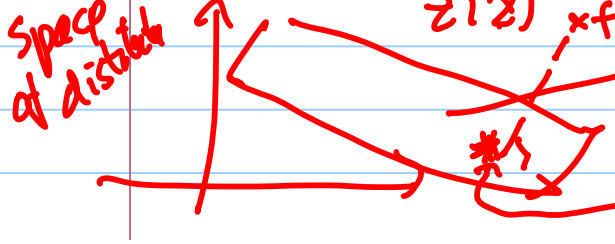
$$D(f\|p) = \underbrace{\sum_x f(x)\log f(x)}_{\text{indep of } \lambda} - \underbrace{\sum_x f(x)\log P(x|\lambda)}_{\text{function of } \lambda}$$

Hence minimize $D(f\|p)$ w.r.t. λ corresponds to minimizing $-\sum_x f(x)\log P(x|\lambda)$

Geometric Interpretation: Information Geometry (Amari).

$P(x|\theta) = \frac{e^{\lambda\cdot\psi(x)}}{Z(\lambda)}$ defines a sub-manifold of distributions

λ's coordinate in the manifold.



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

(11) Relation to ML.

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 $\{\underline{x}_i : i=1,\dots,N\}$

(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 minimizing $D(f \| P(\underline{x}|\underline{\lambda}))$ 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 <u>model pursuit</u>.

(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 statistics of course. Beyond scope of course.

(12) Example: Data consists of pairs of letters

$$X_N = ((x_1^1, x_2^1), (x_1^2, x_2^2), \ldots, (x_1^N, x_2^N))$$

First Model: assume independency

$$P(x_1, x_2) = P(x_1)P(x_2) \quad \text{with} \quad P(x) = \frac{1}{Z[\underline{\lambda}]} e^{\underline{\lambda} \cdot \underline{\phi}(x)}$$

exponential model for letters, as before.

Then $P(x_1, x_2) = \frac{e^{\underline{\lambda} \cdot (\underline{\phi}(x_1) + \underline{\phi}(x_2))}}{Z[\underline{\lambda}]}$ (frequencies of letters)

$\underline{\lambda}$ is estimated as before

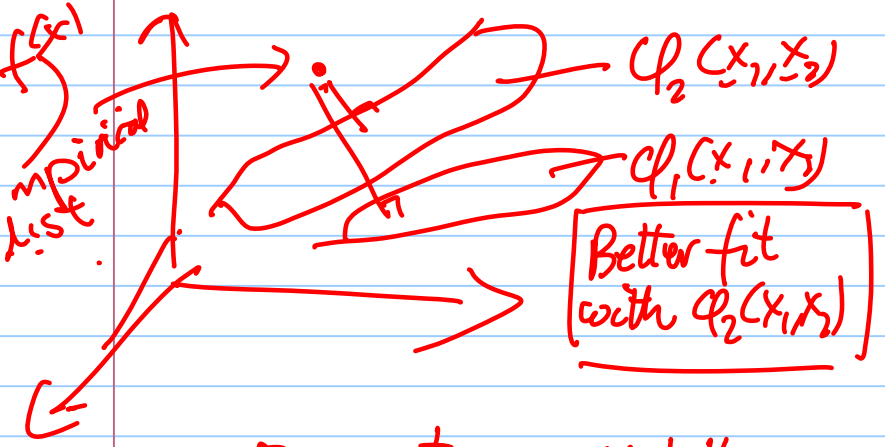
This gives best fit — in the Kullback–Leibler sense — to data, using statistic $\underline{\phi}_1(x_1, x_2) = \underline{\phi}(x_1) + \underline{\phi}(x_2)$.

But we can use a better statistic $\underline{\phi}_2(x_1, x_2)$

Pairwise frequencies of letters with

$$\underline{\phi}_2(x_1, x_2) = \begin{pmatrix} \delta_{x_1,a}\delta_{x_2,a}, & \delta_{x_1,a}\delta_{x_2,b}, \ldots, & \delta_{x_1,a}\delta_{x_2,z} \\ \vdots & & \\ \delta_{x_1,z}\delta_{x_2,a}, & \delta_{x_1,z}\delta_{x_2,b}, \ldots, & \delta_{x_1,z}\delta_{x_2,z} \end{pmatrix}$$

(13) The second model, with $\phi_2(x_1, x_2)$, gives a better fit to the data – if the pairs of letters come from English. Because there pairwise regularities → e.g. qu is frequent, qz is impossible



Note: If we have more letters eg $\chi = \{$ brown, smith, loves, hates, ghost, .. $\}$, then higher order statistics are best.

But higher order statistics require more parameters – M-letters requires 26^M parameters – so we don't usually have enough data. N.B. Shannon fit models like to data to estimate the entropy of English. See slide 16.

(14) Maximum Entropy: Advanced Topic.

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 $P(\underline{x})$

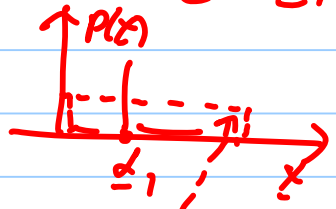
Info from sample $-\log P(\underline{x})$

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

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

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

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



uniform dist.

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

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

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

(15) Maximum Entropy Principle

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

$$-\sum_{\underline{x}} p(\underline{x}) \log p(\underline{x}) + \mu \left(\sum_{\underline{x}} p(\underline{x}) - 1 \right) + \underline{\lambda} \cdot \left(\sum_{\underline{x}} p(\underline{x}) \underline{\phi}(\underline{x}) - \underline{\psi} \right)$$

(Lagrange multipliers: μ for the normalization constraint, $\underline{\lambda}$ for the constraints)

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

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

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

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

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

The maximum entropy principle recovers exponential distribution!

(16) Entropy and ML.

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 $\hat{\underline{\lambda}}$ ← best estimate of $\underline{\lambda}$.

The prob of the data is using $P(\underline{x}|\hat{\underline{\lambda}})$

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

The entropy 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 $\hat{\underline{\lambda}}$)

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

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

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.

Two 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

Shannon 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."?

Which Models to Use? How to select between them?

Suppose we have two models $P(\underline{x}|\text{Model 1})$, $P(\underline{x}|\text{Model 2})$

e.g. $P(\underline{x}|\text{Model 1})$ uses stats $\underline{\phi}_1(\underline{x})$, parameter $\underline{\lambda}_1$

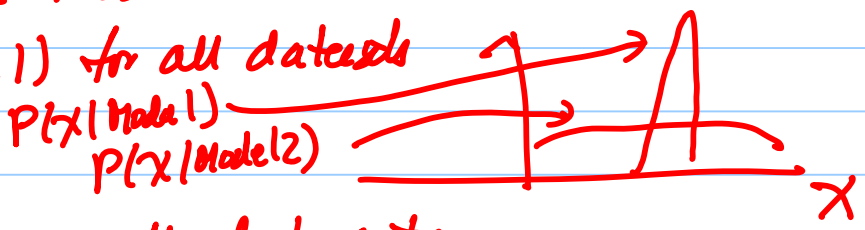
$P(\underline{x}|\text{Model 2})$ " " $\underline{\phi}_2(\underline{x})$, parameter $\underline{\lambda}_2$.

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

Compute $P(\mathcal{X}|\text{Model 1})$ and $P(\mathcal{X}|\text{Model 2})$

Select Model 1 if $P(\mathcal{X}|\text{Model 1}) > P(\mathcal{X}|\text{Model 2})$

Advantage – Occam Factor – this criterion favors simpler more specific models because the distribution must obey $\sum_{\mathcal{X}} P(\mathcal{X}|\text{Model 1})$ 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 $\underline{\phi}_1(\underline{x})$

$$P(\mathcal{X}|\text{Model 1}) = \sum_{\underline{\lambda}} P(\mathcal{X}|\underline{\lambda}) P(\underline{\lambda}|\text{Model 1})$$

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

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

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

(18)

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 $\sum_{\underline{x}} P(\underline{x} \mid \text{Model 1}) = 1$ (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 $P(\underline{x} \mid \underline{\lambda}) = \frac{e^{\underline{\lambda}_1 \cdot \underline{\phi}(\underline{x})}}{Z[\underline{\lambda}]}$

with $-\log P(X \mid \hat{\underline{\lambda}}_2)$ model $\frac{e^{\underline{\lambda}_2 \cdot \underline{\phi}_2(\underline{x})}}{Z[\underline{\lambda}_2]}$

Suppose model 2 has more parameters

→ e.g. $\underline{\phi}_1(\underline{x}) = (\phi_1^1(\underline{x}) \ldots \phi_{10}^1(\underline{x}))$, $\underline{\lambda}_1 = (\lambda_1^1, \ldots \lambda_{10}^1)$

$\underline{\phi}_2(\underline{x}) = (\phi_1^2(\underline{x}), \ldots, \phi_{100}^2(\underline{x}))$, $\underline{\lambda}_2 = (\lambda_1^2, \ldots, \lambda_{100}^2)$

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)