Lecture 5 Entropy and Model Selection

Note Title 1/26/2011

The entropy of a distribution $P(x)$ is

$$H[P] = -\sum_x P(x) \log P(x).$$

It is a measure of the information obtained by observing a sample x from $P(x)$.

Shannon. Information Theory.
Physics. Thermodynamics / Stat Physics

Examples: Let $x \in \{\alpha_1 \ldots \alpha_N\}$,
i.e. x has a finite set of values.

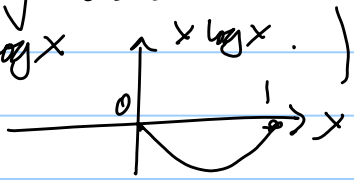
Case (i). Suppose $P(x=\alpha_1)=1$, $P(x=\alpha_i)=0$, $i \neq 1$
There is no uncertainty, we know that the sample from $P(x)$ will be α_1 before we observe it.

Entropy $H[P] = -\sum_{i=1}^{N} P(x=\alpha_i) \log P(x=\alpha_i)$

$= -\{ 1 \log 1 + (N-1)\, 0 \log 0 \}$,

$= 0$. no information is gained by making the observation.

(Note: $0 \log 0$ is defined by $\lim_{x \to 0} x \log x$ $x \log x$.)



Case (ii) $P(x=\alpha_i) = 1/N$, $i = 1 \ldots N$.

All samples are equally likely.

$H[P] = -N \cdot (1/N) \log (1/N) = \log N$.

we gain $\log N$ bits of information by making the observation.

(2) Note Title 1/26/2011

Shannon proposed that we encode information for transmission so that events x will be encoded by $-\log p(x)$ bits
(i.e. high probability x have short codes,
low probability x have long codes).
Then $-\sum_x p(x) \log p(x)$ is the expected coding length of events x from distribution $p(x)$.

Shannon considered the task of modelling the English language
→ sequences of letters.
What is the entropy of English?

How does entropy relate to the learning ideas we discussed last time?

$$p(x|\underline{\lambda}) = \frac{1}{Z[\underline{\lambda}]} e^{\underline{\lambda} \cdot \underline{\varphi}(x)}$$

$$\text{Entropy} = -\sum_x p(x|\underline{\lambda}) \log p(x|\underline{\lambda})$$
$$= \underline{\lambda} \cdot \sum_x \underline{\varphi}(x) p(x) + \log Z[\underline{\lambda}],$$

Suppose we select $\hat{\underline{\lambda}}$ by minimizing the entropy such that $\sum_x \underline{\varphi}\, p(x) = \underline{\nu}$ ← observed statistics

Then $\hat{\underline{\lambda}}$ is exactly the ML estimate $\underline{\lambda}_{ML}$.

(3) 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 multiplier μ ← constraint; lagrange multiplier $\underline{\lambda}$ ← constraint)

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

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

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

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

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

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

The maximum entropy principle recovers exponential distribution!

Note Title 10/7/2006

(4) An alternative viewpoint on ML learning of distributions. This gives deeper understand.

Suppose the data is generated by a distribution $f(x)$.

Define the Kullback-Leibler divergence between $f(x)$ and the model $P(x|\theta)$.

Kullback-Leibler $D(f||P) = \sum_x f(x) \log \frac{f(x)}{P(x|\theta)}$.

KL has the property that

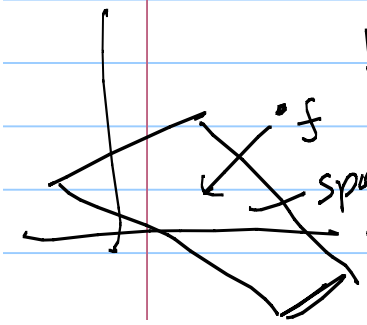
$D(f||p) \geq 0 \quad \forall f, p.$

$D(f||p) = 0$, if, and only if $f(x) = P(x|\theta)$.

So $D(f||p)$ is a measure of the similarity between $f(x)$ and $p(x|\theta)$

We can write

$$D(f||p) = \underbrace{\sum_x f(x) \log f(x)}_{\text{Independent of } \theta.} - \underbrace{\sum_x f(x) \log P(x|\theta)}_{\text{depends on } \theta}$$



(5) Now suppose we have samples (i, i, d) $x_1, \ldots, x_N$ from $f(x)$.

This gives us an empirical distribution

empirical → $f_{emp}(x) = \frac{1}{N} \sum_{i=1}^{N} \delta_{x, x_i}$ ← Kronecker-delta. Indicator Function.

The KL divergence between $f_{emp}(x)$ and $p(x|\theta)$ can be written as:

$$J(\theta) = -\sum_x f_{emp}(x) \log p(x|\theta) + K$$

$$J(\theta) = -\frac{1}{N} \sum_{i=1}^{N} \log p(x_i|\theta) + K$$ ← independent of θ

Minimizing $J(\theta)$ w.r.t. θ, finds the distribution $p(x|\hat{\theta})$ which is closest to $f_{emp}(x)$.

But minimizing $J(\theta)$ w.r.t. is exactly ML. $\hat{\theta} = \arg\max_\theta \sum_{i=1}^{N} \log p(x_i|\theta)$. //

So ML has meaning even if best fit to the model. Even if model is only an approximation.

Note Title 10/7/2006

(6) # Maximum Entropy + Model Selection

Two important papers (Della Pietra et al, Zhu et al) show how this principle is useful for learning complicated models.

(Della Pietra paper is available of the website.)

For a real problem, we do not know which statistics to use. (E.g., stats of letters, letter pairs etc.)

But we can define a large class of possible statistics.

$$\phi_1(\underline{x}), \ldots, \phi_C(\underline{x})$$

where C is very large

Two problems:

(i) selection: which of these statistics to use.

(ii) parameter estimation: how to determine the λ parameters of the model.

(7) Example: Della Pietra et al.
(Zhu, Wu, Mumford)

Want a model to generate strings of text $\underline{w} = (w_1, w_2 \ldots w_n)$

Each w_i can be a letter, a space, or commas, etc.

first order model.

Simplest model: (lower case letters only)

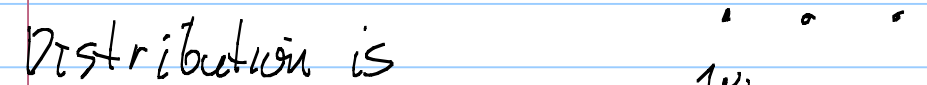
$$p(\underline{w}|N) = \frac{1}{Z} e^{\sum_{i=1}^{N} \underline{\lambda}_{[a-z]} \cdot \underline{\chi}_{[a-z]}(w_i)}$$

$\underline{\lambda}$: parameters; $\underline{\chi}$: potentials. In this case indicator function.

(length of string is generated by another process)

$$\underline{\chi}_{[a-z]}(w_i) = (1, 0, \ldots, 0), \text{ if } w_i = a$$
$$= (0, 0, \ldots, 0, 1), \text{ if } w_i = z$$
$$\underline{\lambda}_{[a-z]} = (\lambda_1, \lambda_2, \ldots, \lambda_{26}),$$

Represent as a graph.



Distribution is independent on letters

i.e. $p(\underline{w}|N) = \prod_{i=1}^{N} p(w_i)$ with $p(w_i) = \frac{1}{Z} e^{\underline{\lambda}_{[a-z]} \cdot \underline{\chi}_{(a-z)}}$

Learning the parameters corresponds to learning the frequencies of letters.

(2) This model does not fit the data (typical words). Because the letters are not independent – i.e. "t" is often followed by "h", "q" is almost followed by "u".

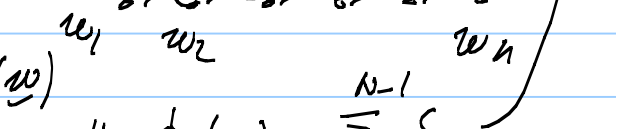
So second-order model using a new statistic. $\underline{X}_{[a-z][a-z]}(w_{ij})$ – indicator function of adjacent letters.

But there are many other features we could use. //

Stochastic sampling from the model is a generative way to check – model

Samples from first order model xevo, ijjiir,
Samples from second order model was, reaser, in,

Graphical Representation (dependent)



Kronecker Delta Indicator Function.

Suppose $P(\underline{w}) = \frac{1}{Z[\underline{\lambda}]} e^{\underline{\lambda} \cdot \underline{\phi}(\underline{w})}$ with $\phi_\mu(\underline{w}) = \sum_{i=1}^{N-1} \delta_{f(w_{i1},w_{i2}),\mu}$

$$\sum_{\mu=1}^{M} \lambda_\mu \phi_\mu(\underline{w}) = \sum_{\mu=1}^{M} \sum_{i=1}^{N-1} \delta_{f(w_{i+1},w_i),\mu} = \sum_{i=1}^{N-1} \lambda_{f(w_{i+1},w_i)}$$

$$P(\underline{w}) = \frac{1}{Z[\underline{\lambda}]} e^{\sum_{i=1}^{N-1} \lambda_{f(w_{i+1},w_i)}}$$

pairwise correlation.

(9) The task of learning is to minimize $\log Z[\underline{\lambda}] - \underline{\lambda} \cdot \underline{\psi}$ w.r.t. $\underline{\lambda}$.

Equivalently, minimize $G[\underline{\lambda}] = Z[\underline{\lambda}]\, e^{-\underline{\lambda}\cdot\underline{\psi}}$

Write: $G[\underline{\lambda}] = \sum_x e^{\sum_\mu \underline{\lambda}_\mu \{\underline{\phi}_\mu(x) - \underline{\psi}_\mu\}}$. //

Initialize this by setting $\underline{\lambda}_\mu = 0$, $\forall \mu$.
(i.e. no statistics are selected)

Minimize $G[\underline{\lambda}]$ by a "feature pursuit" strategy. (This will be used later in the course also).

At time t, we have state $\{\underline{\lambda}_\mu^t\}$ $\left(\underline{\lambda}_\mu^{t=0} = 0\right)$

(A). Minimize w.r.t. each $\underline{\lambda}_\nu$ seperately to solve for $\underline{\lambda}_\nu^{t+1}$
by solving: $\dfrac{\partial G[\underline{\lambda}]}{\partial \underline{\lambda}_\nu} = 0$, keep other $\underline{\lambda}_\mu$ fixed.

(B). Select the feature ν, that minimizes $G\left[\{\underline{\lambda}_\mu^t : \mu \neq \nu,\ \underline{\lambda}_\nu^{t+1}\}\right]$

(C) Update $\underline{\lambda}_\nu^t \to \underline{\lambda}_\nu^{t+1}$.

Problem: both stages (A) & (B) are difficult

(10) New Page: Simply steps (A) & (B)

Minimize $G[\{\underline{\lambda}_\mu\}]$ w.r.t. $\underline{\lambda}_\nu$, with $\underline{\lambda}_\mu$ fixed, $\forall \mu \neq \nu$.

$\underline{\lambda}_\mu = \underline{\lambda}_\mu^t$, $\forall \mu \neq \nu$, $\underline{\lambda}_\nu = \underline{\lambda}_\nu^t + \underline{\Delta}_\nu$, with $\underline{\Delta}_\nu$ unknown.

$$\frac{\partial G}{\partial \underline{\lambda}_\nu} = \sum_x (\underline{\phi}_\nu(x) - \underline{\psi}_\nu)\, e^{\sum_\mu \underline{\lambda}_\mu \cdot (\underline{\phi}_\mu(x) - \underline{\psi}_\mu)}$$

write as $$\frac{\partial G}{\partial \underline{\lambda}_\nu} = \sum_x (\underline{\phi}_\nu(x) - \underline{\psi}_\nu)\, e^{\underline{\Delta}_\nu \cdot \underline{\phi}_\nu(x)}\, e^{\sum_\mu \underline{\lambda}_\mu^t \cdot (\underline{\phi}_\mu(x) - \underline{\psi}_\mu)}$$

Hence $\frac{\partial G}{\partial \underline{\lambda}_\nu} = 0$, is equivalent to solving

(A) $$\sum_x (\underline{\phi}_\nu(x) - \underline{\psi}_\nu)\, e^{\underline{\Delta}_\nu \cdot \underline{\phi}_\nu(x)}\, P(x \mid \{\underline{\lambda}_\mu^t\}) = 0, \text{ for } \underline{\Delta}_\nu$$

We drop the constant $e^{-\sum_\mu \underline{\lambda}_\mu^t \cdot \underline{\psi}_\mu}$ and normalize.

We can use a similar technique to represent

$$G[\{\underline{\lambda}_\mu\}] = \sum_x e^{\sum_\mu \underline{\lambda}_\mu \cdot (\underline{\phi}_\mu(x) - \underline{\psi}_\mu)}$$

$$= \sum_x e^{\underline{\Delta}_\nu \cdot (\underline{\phi}_\nu(x) - \underline{\psi}_\nu)}\, e^{\sum_\mu \underline{\lambda}_\mu^t \cdot \underline{\phi}_\mu(x)}\, e^{-\sum_\mu \underline{\lambda}_\mu^t \cdot \underline{\psi}_\mu}$$

where $\underline{\lambda}_\mu = \underline{\lambda}_\mu^t$, for all $\mu \neq \nu$, $\underline{\lambda}_\nu = \underline{\lambda}_\nu^t + \underline{\Delta}_\nu$.

Hence $$G[\{\underline{\lambda}_\mu\}] = \underbrace{\left(Z[\{\underline{\lambda}_\mu^t\}]\, e^{-\sum_\mu \underline{\lambda}_\mu^t \cdot \underline{\psi}_\mu}\right)}_{\text{indep of } \nu.} \sum_x e^{\underline{\Delta}_\nu \cdot (\underline{\phi}_\nu(x) - \underline{\psi}_\nu)}\, P(x \mid \{\underline{\lambda}_\mu(\nu)\})$$

(B) To minimize $G[\cdot]$ w.r.t. ν, need to minimize $\rightarrow$ $\sum_x e^{\underline{\Delta}_\nu \cdot (\underline{\phi}_\nu(x) - \underline{\psi}_\nu)}\, P(x \mid \{\underline{\lambda}_\mu^t\})$

(11) But there are some situations where this is not difficult. (See Della Pietra et al.)

Let the features $\phi_\mu(\underline{x})$ be binary-valued.

Define: $\chi_\nu^+ = \{\underline{x} : \phi_\nu(\underline{x}) = +1\}$

$\chi_\nu^- = \{\underline{x} : \phi_\nu(\underline{x}) = -1\}$.

Express (1) as: (for this special case)

$$\sum_{\underline{x} \in \chi_\nu^+} (1-\psi_\nu)\, e^{\Delta_\nu}\, p(\underline{x} | \langle \lambda_\mu^{\epsilon_\mu} \rangle) + \sum_{\underline{x} \in \chi_\nu^-} (-1-\psi_\nu)\, e^{-\Delta_\nu}\, p(\underline{x} | \langle \lambda_\mu^{\epsilon_\mu} \rangle) = 0$$

Hence $\hat{\Delta}_\nu = \frac{1}{2} \log \left\{ \dfrac{(1+\psi_\nu) \sum_{\underline{x} \in \chi_\nu^-} p(\underline{x} | \langle \lambda_\mu^{\epsilon_\mu} \rangle)}{(1-\psi_\nu) \sum_{\underline{x} \in \chi_\nu^+} p(\underline{x} | \langle \lambda_\mu^{\epsilon_\mu} \rangle)} \right\}$

Express (2) as:

Select ν to minimize $e^{\hat{\Delta}_\nu} \sum_{\underline{x} \in \chi_\nu^+} p(\underline{x} | \langle \lambda_\mu^{\epsilon_\mu} \rangle) + e^{-\hat{\Delta}_\nu} \sum_{\underline{x} \in \chi_\nu^-} p(\underline{x} | \langle \lambda_\mu^{\epsilon_\mu} \rangle)$.

So we need to evaluate $\sum_{\underline{x} \in \chi_\nu^+} p(\underline{x} | \langle \lambda_\mu^{\epsilon_\mu} \rangle)$ & $\sum_{\underline{x} \in \chi_\nu^-} p(\underline{x} | \langle \lambda_\mu^{\epsilon_\mu} \rangle)$

(12) It remains to compute:

(##) $\sum_{\underline{x} \in X_v^+} P(\underline{x} | \langle \underline{\lambda}_\mu^{t+1} \rangle)$, $\sum_{\underline{x} \in X_v^-} P(\underline{x} | \langle \underline{\lambda}_\mu^{t+1} \rangle)$

(The sum of both of them is 1) for every v.

If the distribution $P(\underline{x} | \langle \underline{\lambda}_\mu^{t+1} \rangle)$ is defined over a finite set of states (Della Pietra et al) then these can be computed easily. //

Otherwise: Stochastic Sampling (eg. Jun Liu's Book. Stat 202C)

Key Point: Stochastic Sampling is very effective at estimating summations/integrals in high dimensions.

(ie. generate data from current model.)

Also, for this problem, we only need to sample from $P(\underline{x} | \langle \underline{\lambda}_\mu^{t+1} \rangle)$ once to get sample $\{\underline{x}_1, \ldots, \underline{x}_M\}$ $M \gg 1$

Then we can compute ## from these samples without needing to sample separately for each v. //

(13) # Penalizing Additional Features.

If our algorithm selects a feature v that is already non-zero, then we are merely adjusting the parameter λ_v of that feature.

But if it selects a feature v with $\lambda_v = 0$, then we are activating a feature that has not been used yet.

It can be good to penalize the no. of features used. There are a variety of criteria used to do this AIC, BIC,

$K N$ or $K \log N$.

K - constant.

(14) # Model Selection

To compare two models:

Model 1:

$P(\underline{x} \mid \underline{\theta}, \text{Model } 1)$, $P(\underline{\theta} \mid \text{Model } 1)$

$P(\underline{x} \mid \underline{\phi}, \text{Model } 2)$, $P(\underline{\phi} \mid \text{Model } 2)$

$$P(\underline{x} \mid \text{Model } 1) = \sum_{\underline{\theta}} P(\underline{x} \mid \underline{\theta}, \text{Model } 1) \; P(\underline{\theta} \mid \text{Model } 1)$$

$$P(\underline{x} \mid \text{Model } 2) = \sum_{\underline{\phi}} P(\underline{x} \mid \underline{\phi}, \text{Model } 2) \; P(\underline{\phi} \mid \text{Model } 2)$$

Then use the log-likelihood ratio test:

$$\log \frac{P(\underline{x} \mid \text{Model } 1)}{P(\underline{x} \mid \text{Model } 2)} \gtrless T \quad \leftarrow \text{Threshold.}$$

Observe: that this takes into account all the different parameter values that could generate the data

Gives an Occam factor — penalizes models with unnecessary large no. of parameters (*)

(9) Minimax versus Model Selection.

Zhu et al. proposed learning distributions by minimax.

(1) Use Maximum Entropy to learn a set of distributions using different statistics.

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

$\underline{\phi}$ statistics

$\underline{\phi}(\underline{x})$ are chosen

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

$\underline{\psi}$ observed statistics.

(2) Use Minimum Entropy to select between different distributions – ie to compare

$p_1(\underline{x}|\underline{\lambda}_1) = \frac{1}{Z[\underline{\lambda}_1]} e^{\underline{\lambda}_1\cdot\underline{\phi}_1(\underline{x})}$ $\underline{\psi}_1$

$p_2(\underline{x}|\underline{\lambda}_2) = \frac{1}{Z[\underline{\lambda}_2]} e^{\underline{\lambda}_2\cdot\underline{\phi}_2(\underline{x})}$ $\underline{\psi}_2$

Entropy is $-\sum_{\underline{x}} p(\underline{x}|\underline{\lambda}) \log p(\underline{x}|\underline{\lambda})$

$= \log Z[\underline{\lambda}] - \underline{\lambda}\cdot\underline{\psi}$

(10) Hence using the minimum entropy principle is like maximum likelihood.

→ both involve minimizing

$$\log Z[\underline{\lambda}] - \underline{\lambda} \cdot \underline{\psi}$$

Consider: $$p(\underline{x} \mid \underline{\lambda}_1, \underline{\lambda}_2) = \frac{1}{Z[\underline{\lambda}_1, \underline{\lambda}_2]} e^{\underline{\lambda}_1 \cdot \underline{\phi}_1(\underline{x}) + \underline{\lambda}_2 \cdot \underline{\phi}_2(\underline{x})}$$

Using minimax entropy is choosing

$(\underline{\lambda}_1 = \underline{\lambda}_1, \underline{\lambda}_2 = 0)$ over $(\underline{\lambda}_1 = 0, \underline{\lambda}_2 = 0)$

provided $p(\underline{x} \mid \underline{\lambda}_1, 0) > p(\underline{x} \mid 0, \underline{\lambda}_2)$.

This differs from model selection. Where you should compare

$$\log \frac{\sum_{\underline{\lambda}_1} p(\underline{x} \mid \underline{\lambda}_1) p(\underline{\lambda}_1)}{\sum_{\underline{\lambda}_2} p(\underline{x} \mid \underline{\lambda}_2) p(\underline{\lambda}_2)} \quad //$$

(17) Occam Factor: Mackay.

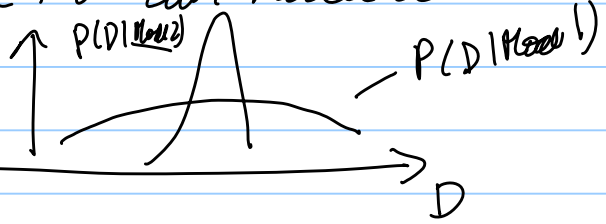
Automatic selection of complexity of the model.

P(D | Model 1)

P(D | Model 2)

Suppose Model 1 is flexible → ie. it can model a lot of datasets.

But Model 2 is more specific



Then Model selection will automatically favor the more specific model.

The Occam factor: