Advanced Machine Learning

Entropy concepts used for machine learning

Joan Andreu Sánchez

Departamento de Sistemas Informáticos y Computación
Universitat Politècnica de València

URL: http://www.prhlt.upv.es/∼jandreu
e-mail: jandreu@prhlt.upv.es

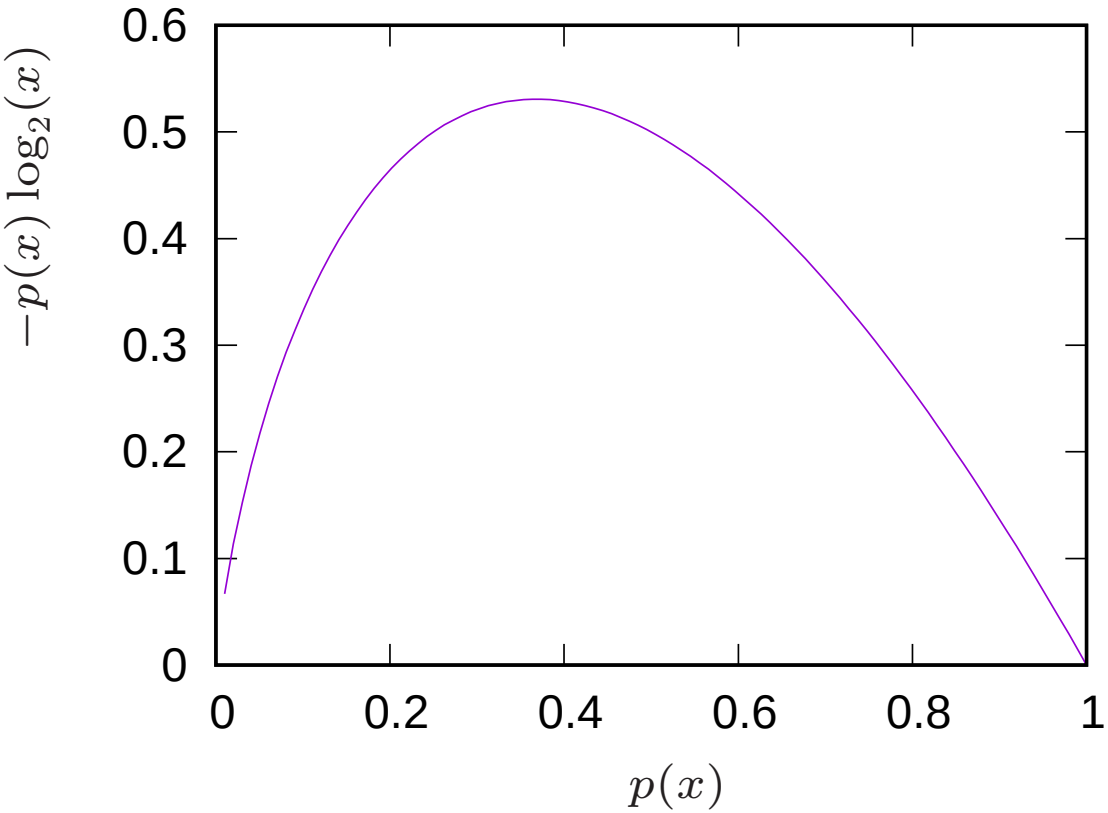
Index

Index

Entropy definitions

Entropy of a discrete variable:

$$H(X) = -\sum_x p(x) \ \log p(x) \tag{1}$$



Exercise (***): Compute the derivative of expr. (1). (see solution here)

Entropy definitions

If the log is to the base 2 then the entropy is measured in bits

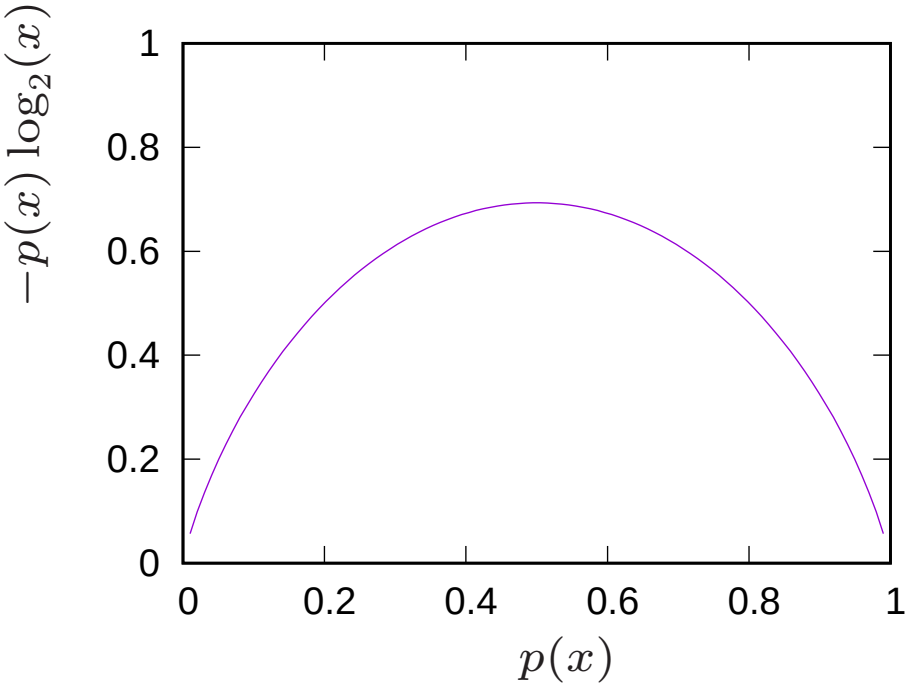
Interpretation 1: average number of bits needed to describe X

Interpretation 2: *uncertainty* about the outcome of a stochatic variable X

Property: entropy is a concave function (see demonstration here)

Property: entropy is maximum when p is uniform $\rightarrow$ maximun *uncertainty*

Example: a coin with 2 faces and identical probability



Entropy definitions

Example [Cover 1991]: Consider a dice with 8 faces (with numbers from 0 to 7), all of them with the same probabibility. A 3-bit string is necessary as label for each face. The entropy of this random variable is:

$$H(X) = -\sum_{i=1}^{8} p(i) \log p(i) = -\sum_{i=1}^{8} \frac{1}{8} \log \frac{1}{8} = \log 8 = 3 \text{ bits}$$

Intuition: let us supose that we send the winner face to a friend. The average number of bits we need to send to the friend to identify the face is 3 bits.

Let us supose that:

$$P(X=0) = \frac{1}{2}, \quad P(X=1) = \frac{1}{4}, \quad P(X=2) = \frac{1}{8}, \quad P(X=3) = \frac{1}{16}$$

$$P(X=4) = P(X=5) = P(X=6) = P(X=7) = \frac{1}{64}$$

Then, the entropy is: $H(X) = 2$ bits

Intuition: the most probable outcomes need less bits to be coded. Consider the following coding for the faces: $\{0, 10, 110, 1110, 111100, 111101, 111110, 111111\}$

$$\frac{1}{2} + 2\,\frac{1}{4} + 3\,\frac{1}{8} + 4\,\frac{1}{16} + 4 \cdot 6\,\frac{1}{64} = 2$$

Entropy definitions

Joint entropy of a pair of discrete random variables (X, Y) with a joint distribution $p(x, y)$ [Cover 1991]:

$$H(X,Y) = -\sum_x \sum_y p(x,y) \ \log p(x,y) \tag{2}$$

Conditional entropy of a pair of discrete random variables (X, Y):

$$\begin{aligned} H(Y|X) &= \sum_x p(x) \ H(Y|X=x) && (3)\\ &= -\sum_x p(x) \sum_y p(y|x) \ \log p(y|x) && (4)\\ &= -\sum_x \sum_y p(x,y) \ \log p(y|x) && (5) \end{aligned}$$

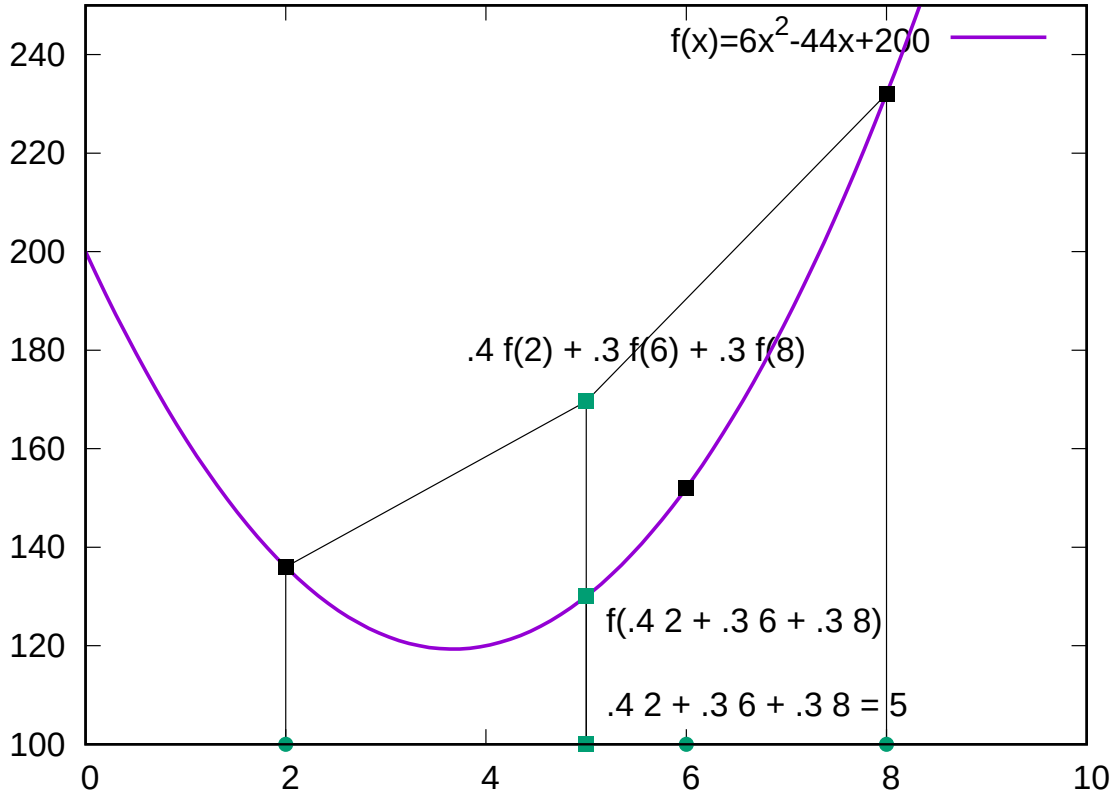
Theorem [Cover 1991]

$$H(X,Y) = H(X) + H(Y|X) \tag{6}$$

Entropy definitions

Theorem. (Jensen's inequality) Let f be a convex function defined on an interval I. If $x_1, x_2, \ldots, x_n \in I$ and $\lambda_1, \lambda_2, \ldots, \lambda_n \geq 0$ with $\sum_{i=1}^{n} \lambda_i = 1$, then,

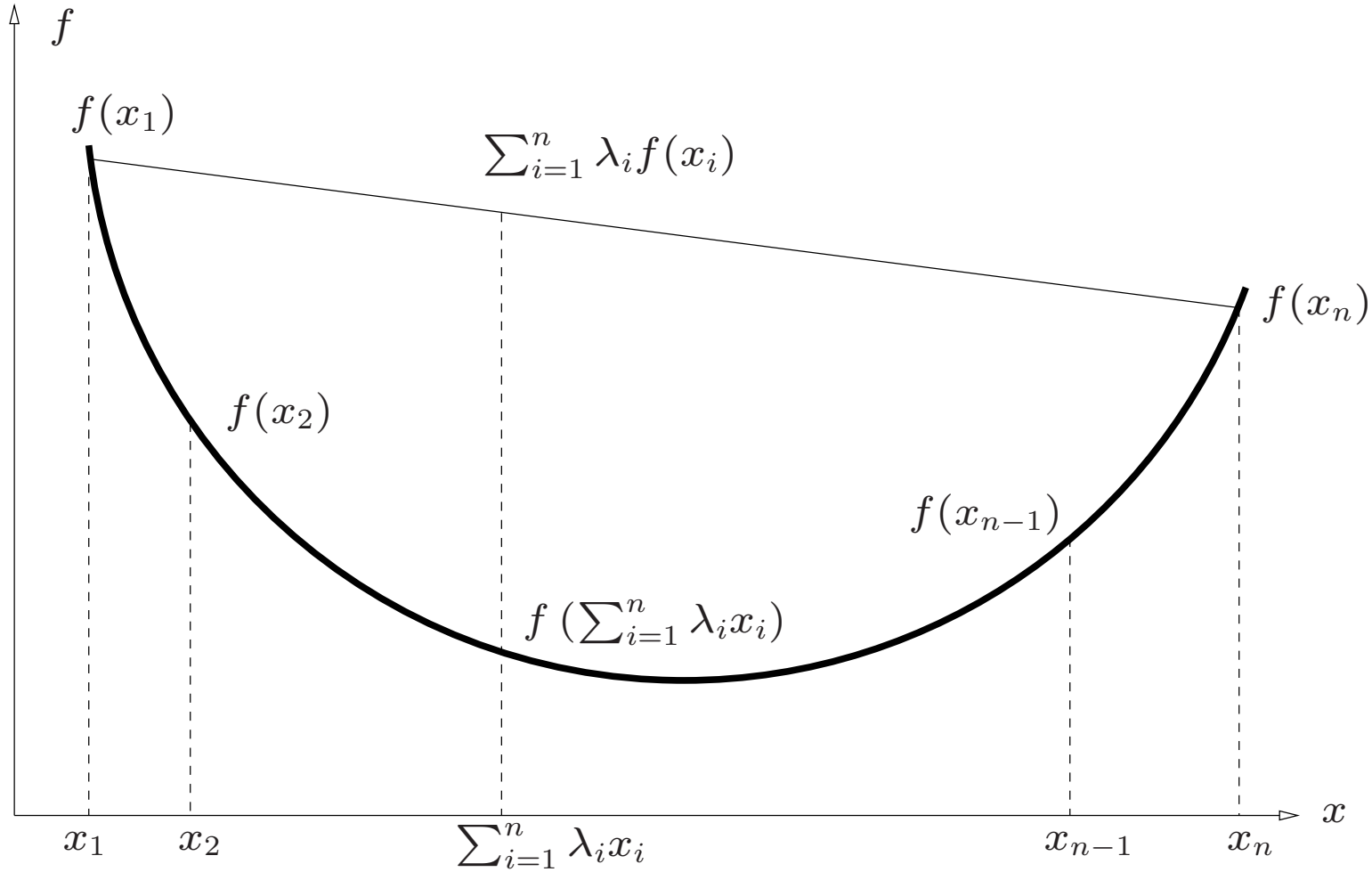
$$f\left(\sum_{i=1}^{n} \lambda_i x_i\right) \leq \sum_{i=1}^{n} \lambda_i f(x_i). \tag{7}$$



$$\lambda_1 = .4 \quad \lambda_1 = .3 \quad \lambda_1 = .3$$

Entropy definitions

Graphical interpretation of Jensen's inequality



Entropy definitions

Relative entropy or *Kullback-Leibler distance* between two probability mass functions $p(x)$ and $q(x)$:

$$D(p||q) = \sum_x p(x) \log \frac{p(x)}{q(x)} \tag{8}$$

- Non-negative

$$\begin{aligned} -D(p||q) &= -\sum_x p(x) \log \frac{p(x)}{q(x)} = \sum_x p(x) \log \frac{q(x)}{p(x)} \leq \log \sum_x p(x) \frac{q(x)}{p(x)} \\ &= \log \sum_x q(x) = \log 1 = 0 \end{aligned}$$

The inequality follows from Jensen's inequality

- Non-symetric and does not satisfy the triangle inequality

Exercise (**): Demonstrate if $D(p||q) + D(q||p)$ is or is not symetric and satisfy the triangle inequality.

Entropy definitions

Mutual information between a pair of discrete random variables (X, Y):

$$I(X;Y) = \sum_x \sum_y p(x,y) \log \frac{p(x,y)}{p(x)\,p(y)} \tag{9}$$

Interpretation: It measures how much knowing one of these variables reduces uncertainty about the other[1]

$$I(X;Y) = H(X) - H(X|Y) \tag{10}$$

$$= H(Y) - H(Y|X) \tag{11}$$

Property:

$$I(X;Y) = D(p(x,y)||p(x)p(y)) \geq 0$$

[1] From wikipedia: For example, if X and Y are independent, then knowing X does not give any information about Y and vice versa, so their mutual information is zero. At the other extreme, if X is a deterministic function of Y and Y is a deterministic function of X then all information conveyed by X is shared with Y: knowing X determines the value of Y and vice versa. As a result, in this case the mutual information is the same as the uncertainty contained in Y (or X) alone, namely the entropy of Y (or X)

Entropy definitions

Interpretation of mutual information in special cases

- X and Y independent:

$$\begin{aligned} I(X;Y) &= H(Y) - H(Y|X) = H(Y) - \sum_x p(x)\ H(Y|X=x) \\ &= H(Y) + \sum_x p(x) \sum_y p(y|x)\ \log p(y|x) = H(Y) + \sum_x p(x) \sum_y p(y)\ \log p(y) \\ &= H(Y) - H(Y) \sum_x p(x) = 0 \end{aligned}$$

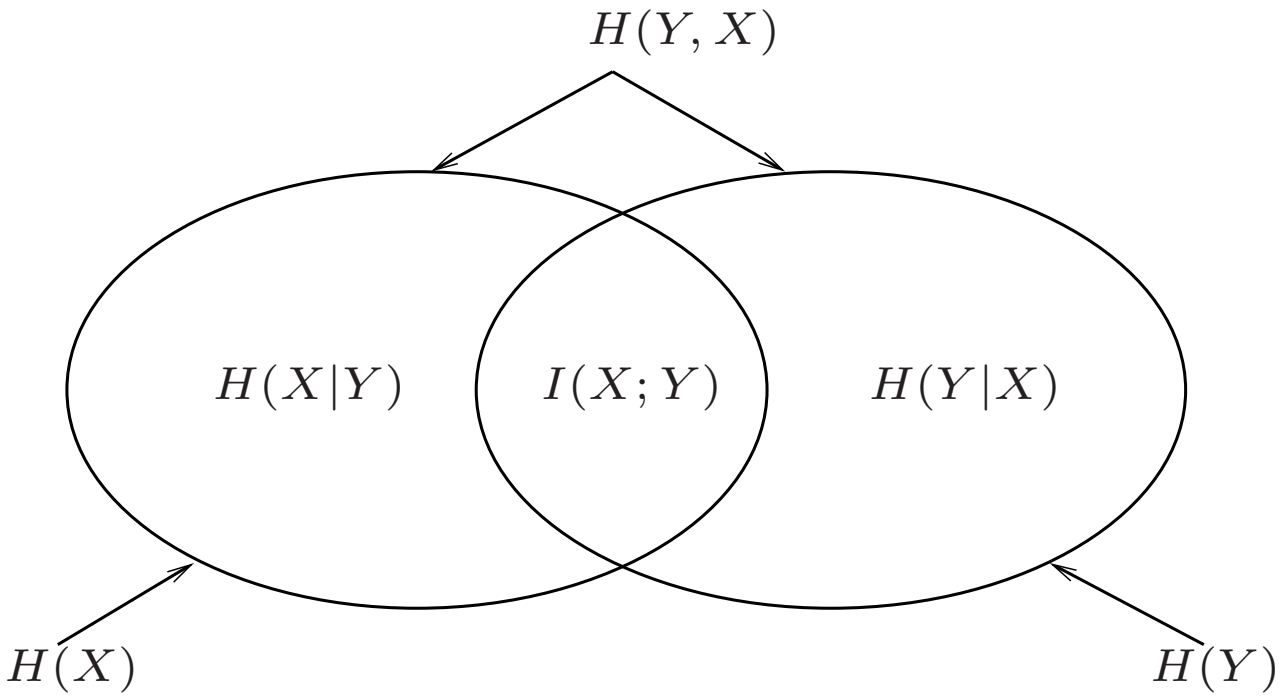
- X and Y are "totally dependent" (or the same variable, see comments above):

$$\begin{aligned} I(X;X) &= H(X) - H(X|X) = H(X) - \sum_x p(x)\ H(X|X=x) \\ &= H(X) - \sum_x p(x) \sum_x p(x|x)\ \log \underbrace{p(x|x)}_{=1^{(*)}} = H(X) \end{aligned}$$

$^{(*)} P(A|B) = \frac{P(A \cap B)}{P(B)} \underset{A=B}{=} \frac{P(B)}{P(B)} = 1$

Entropy definitions

$$
\begin{aligned}
I(X;Y) &= H(X) - H(X|Y) && (12)\\
&= H(Y) - H(Y|X) && (13)\\
&= H(X) + H(Y) - H(Y,X) && (14)
\end{aligned}
$$



Entropy definitions

Example [Cover 1991]: Let (X, Y) have the following joint distribution:

$Y \setminus X$	1	2	3	4	$\sum$
1	$1/8$	$1/16$	$1/32$	$1/32$	$1/4$
2	$1/16$	$1/8$	$1/32$	$1/32$	$1/4$
3	$1/16$	$1/16$	$1/16$	$1/16$	$1/4$
4	$1/4$	0	0	0	$1/4$
$\sum$	$1/2$	$1/4$	$1/8$	$1/8$	1

Marginal of X: $(\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8})$ Marginal of Y: $(\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4})$

$$
\begin{aligned}
H(X|Y) &= \sum_{i=1}^{4} p(Y=i)\, H(X|Y=i) \\
&= \frac{1}{4}H(\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8}) + \frac{1}{4}H(\frac{1}{4}, \frac{1}{2}, \frac{1}{8}, \frac{1}{8}) + \frac{1}{4}H(\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}) + \frac{1}{4}H(1,0,0,0) \\
&= \frac{11}{8} = 1.375 \quad \text{bits} \\
I(X;Y) &= 0.375 \quad \text{bits}
\end{aligned}
$$

Entropy definitions

Interesting properties [Cover 1991; Murphy 2022]

- $I(X;Y) \geq 0$
- $H(X|Y) \leq H(X)$

Exercise (*): For a given value of $Y = y$, is it possible that $H(X|Y = y) \geq H(X)$? Provide a demonstration.

Exercise (*): Write an example similar to the example in the previous slide but changing the joint distributions, where variables have a real meaning (real sense) (e.g. alcohol in blood versus car accident, weather versus car accident, ...). Then compute the $H(X), H(X,Y), H(X|Y)$ and $I(X;Y)$.

Entropy definitions

Exercise (taken from SIN-ETSINF)

Random variables:

- *Climatología* (C): *despejado* (DES), *nublado* (NUB), *lluvia* (LLU)
- *Luminosidad* (L): *dia* (DIA), *noche* (NOC)
- *Seguridad* (S): desplazamiento *seguro* (SEG), o con *accidente* (ACC)

$P(c,s)$	DES	NUB	LLU	
SEG	0.43	0.30	0.13	
ACC	0.03	0.03	0.08	
				$\sum=1$

$P(l,s)$	DIA	NOC	
SEG	0.57	0.29	
ACC	0.05	0.09	
			$\sum=1$

$P(c,l)$	DES	NUB	LLU	
DIA	0.31	0.21	0.10	
NOC	0.15	0.12	0.11	
				$\sum=1$

	DIA			NOC			
$P(s,c,l)$	DES	NUB	LLU	DES	NUB	LLU	
SEG	0.30	0.20	0.07	0.13	0.10	0.06	
ACC	0.01	0.01	0.03	0.02	0.02	0.05	
							$\sum=1$

Entropy definitions

Exercise (*): (choose only one of them)

Compute $H(C), H(S), H(C|S), H(S|C), H(C,S)$, and $I(C;S)$

Compute $H(L), H(S), H(L|S), H(S|L), H(L,S)$, and $I(L;S)$

Compute $H(L), H(C), H(L|C), H(C|L), H(L,C)$, and $I(C;L)$

Exercise (**): (choose only one of them)

Compute $H(C|L,S), H(L|C,S), H(S|C,L)$

Compute $H(C,L|S), H(L,S|C), H(S,C|L)$

Compute $I(C;L;S)$

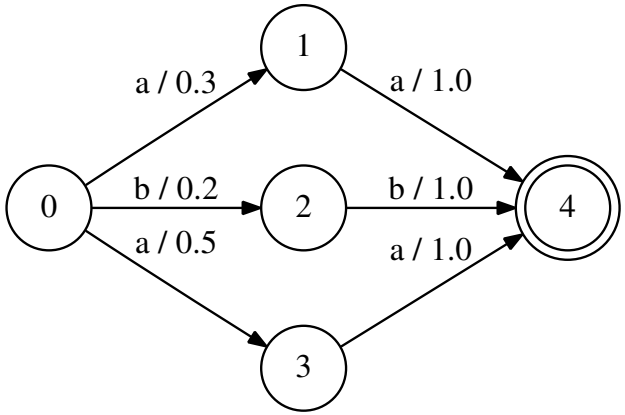
Index

Entropy measures for grammatical models

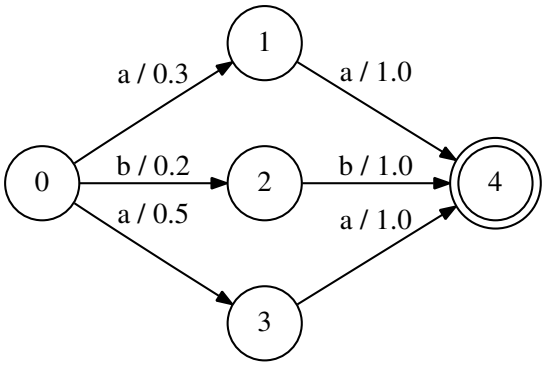
Definition A *PFA* is a tuple $\mathcal{A} = \langle Q, \Sigma, \delta, I, F, P \rangle$, where: Q is a finite set of states; Σ is the alphabet; $\delta \subseteq Q \times \Sigma \times Q$ is a set of transitions; $I : Q \to \mathbb{R}^{\geq 0}$ is the probability function of a state being an initial state; $P : \delta \to \mathbb{R}^{\geq 0}$ is a probability function of probabilities transition between states; $F : Q \to \mathbb{R}^{\geq 0}$ is the probability function of a state being a final state.
I, P, and F are functions such that:

$$\sum_{i \in Q} I(i) = 1 \tag{15}$$

$$\forall i \in Q, F(i) + \sum_{v \in \Sigma, j \in Q} P(i, v, j) = 1\,. \tag{16}$$



Entropy measures for grammatical models



- Sequence entropy [Grenander 1967]

$$H_{\mathcal{A}}(W) = -\sum_{w \in L(\mathcal{A})} p_{\mathcal{A}}(w) \log p_{\mathcal{A}}(w) \quad (17)$$

$$\begin{aligned} H_{\mathcal{A}}(W) &= -p_{\mathcal{A}}(aa) \log p_{\mathcal{A}}(aa) - p_{\mathcal{A}}(bb) \log p_{\mathcal{A}}(bb) \\ &= -0.8 \log 0.8 - 0.2 \log 0.2 = 0.72 \text{ bits} \end{aligned}$$

- Entropy on the paths given an observation sequence aa [Hernando 2005]

$$H_{\mathcal{A}}(\theta|w) = -\sum_{\theta \in \Theta_{\mathcal{A}}(w)} p_{\mathcal{A}}(\theta|w) \log p_{\mathcal{A}}(\theta|w) \quad (18)$$

$$\begin{aligned} H_{\mathcal{A}}(\theta|aa) &= -p_{\mathcal{A}}(034|aa) \log p_{\mathcal{A}}(034|aa) - p_{\mathcal{A}}(014|aa) \log p_{\mathcal{A}}(014|aa) \\ &= -\frac{0.5}{0.8} \log \frac{0.5}{0.8} - \frac{0.3}{0.8} \log \frac{0.3}{0.8} = 0.95 \text{ bits} \end{aligned}$$

- Derivational entropy [Grenander 1967]

$$H_{\mathcal{A}}(\Theta) = -\sum_{\theta \in \Theta(\mathcal{A})} p_{\mathcal{A}}(\theta) \log p_{\mathcal{A}}(\theta) \quad (19)$$

$$\begin{aligned} H_{\mathcal{A}}(\Theta) &= -p_{\mathcal{A}}(034) \log p_{\mathcal{A}}(034) - p_{\mathcal{A}}(014) \log p_{\mathcal{A}}(014) - p_{\mathcal{A}}(024) \log p_{\mathcal{A}}(024) \\ &= -0.5 \log 0.5 - 0.3 \log 0.3 - 0.2 \log 0.2 = 1.49 \text{ bits} \end{aligned}$$

Entropy measures for grammatical models

Computation of the derivational entropy

Given a *PFA* $\mathcal{A}$, we define the characteristic matrix M [Thompson 74] of dimensions $|Q| \times |Q|$ as[2]:

$$M(i,j) = \sum_{v \in \Sigma} P(i,v,j) \ . \tag{20}$$

The following vector ξ, $0 \leq i < |Q| - 1$ is defined [Soule 1974]:

$$\xi(i) = - \sum_{\substack{v \in \Sigma \\ 0 \leq j \leq |Q|-1}} P(i,v,j) \log P(i,v,j) \tag{21}$$

and $\xi(|Q| - 1) = 0$.

Theorem [Grenander 1967] If the largest eigenvalue of the characteristic matrix M is strictly less than 1, then the derivational entropy of the model can be computed as: $H_{\mathcal{A}}(\Theta) = (I - M)^{-1}\xi$.

---

[2] Let's supose that states are named with integers and there is only one initial state, 0, and one final state, $|Q| - 1$

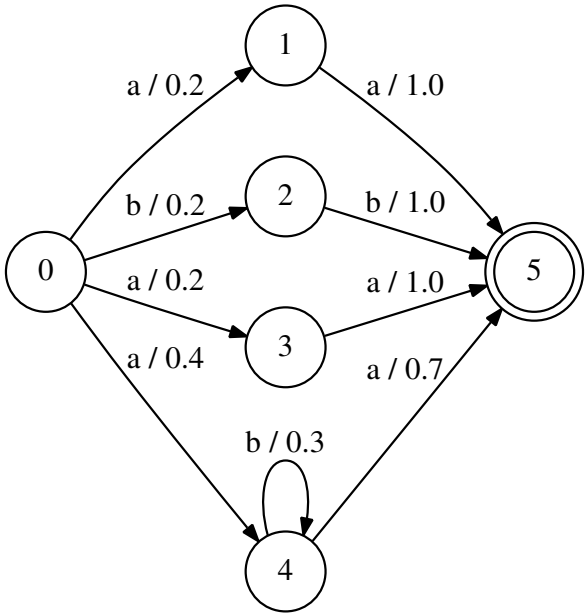
Entropy measures for grammatical models

Comments and additional properties:

- $H_{\mathcal{A}}(W) \leq H_{\mathcal{A}}(\Theta)$ with equality if the automaton is non-ambiguous [Grenander 1967]
- Computation of $H_{\mathcal{A}}(W)$ is no feasible [Grenander 1967] in general (see example below)
- Computation of $H_{\mathcal{A}}(\Theta)$ is feasible in general with time complexity $O(|Q|^3)$ [Grenander 1967]. Less time complexity if the automaton is acyclic [Sánchez 2018]
- Computation of $H_{\mathcal{A}}(\theta|w)$ requires time complexity $O(|w||Q|^2)$ [Hernando 2005]

Entropy measures for grammatical models

A particular case for the computation of the sequence entropy $H_{\mathcal{A}}(W)$[1]



$$
\begin{aligned}
aa &\longrightarrow -0.68 \log 0.68 = .3783 \\
bb &\longrightarrow -0.2 \ \log 0.2 = .4644 \\
ab^n a &\longrightarrow -\sum_{n>0} p(ab^n a) \log p(ab^n a) = 1.78
\end{aligned}
$$

Exercise (**): Provide a much more complex example that has to include loops, cycles, . . .

Exercise (****): Define an algorithm to compute the sequence entropy on an acyclic PFA and/or on left-to-right discrete HMM. Hist: DP with states and prefixes. (Open research problems!)

[1] See Appendix 1 for some details.

Entropy measures for grammatical models

Exercise (****): Adapt the computation of the entropy of the paths for a given sequence for PFA and apply the developed algorithm to a the given example. Take the following definitions as starting point and follow [Hernando 2005].

$$H_{\mathcal{A}}(\theta|w) \quad = \quad - \sum_{\theta \in \Theta_{\mathcal{A}}(w)} p_{\mathcal{A}}(\theta|w) \log p_{\mathcal{A}}(\theta|w)$$

Definitions:

- $H_t(j) = H(\theta_{1,t-1}|\theta_t = j, w_{1,t})$: entropy of all paths (state sequences) that lead to state j at time t, given the observation up to time t.
- $c_t(j) = p_{\mathcal{A}}(\theta_t = j|w_{1,t})$: probability of being in state j at time t, given the word sequence up to time t.

$$c_t(j) = \frac{\sum_{i=0}^{|\mathcal{Q}|-1} c_{t-1}(i) P(i, w_t, j)}{\sum_{k=0}^{|\mathcal{Q}|-1} \sum_{i=0}^{|\mathcal{Q}|-1} c_{t-1}(i) P(i, w_t, k)} \tag{22}$$

Auxiliary computations (see (6) in [Hernando 2005] for details):

$$p(\theta_{t-1} = i|\theta_t = j, w_{1,t}) \quad = \quad \frac{P(i, w_t, j) c_{t-1}(i)}{\sum_{k=0}^{|\mathcal{Q}|-1} P(k, w_t, j) c_{t-1}(k)} \tag{23}$$

Exercise (**): Demonstrate (23) for PFA.

Entropy measures for grammatical models

Initialization. For $0 \leq j < |\mathcal{Q}|$

$$H_0(j) = 0 \tag{24}$$

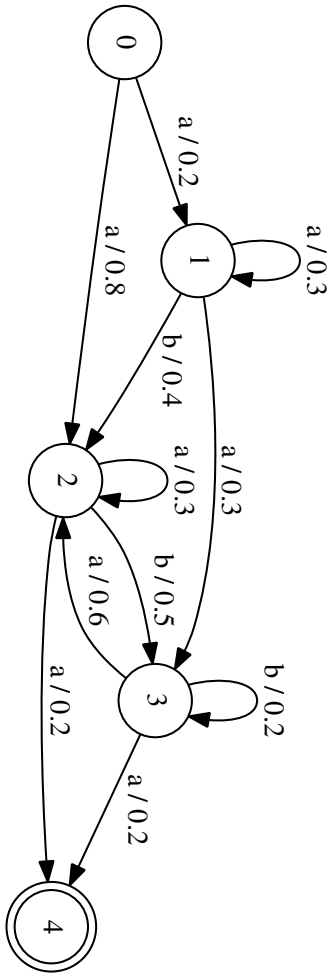
$$c_0(j) = I(j) \tag{25}$$

Recursion. For $0 \leq j < |\mathcal{Q}| - 1; 1 \leq t \leq |w|$:

$$\begin{aligned} c_t(j) &= \frac{\sum_{i=0}^{|\mathcal{Q}|-1} c_{t-1}(i) P(i, w_t, j)}{\sum_{k=0}^{|\mathcal{Q}|-1} \sum_{i=0}^{|\mathcal{Q}|-1} c_{t-1}(i) P(i, w_t, k)} \\ p(\theta_{t-1} = i | \theta_t = j, w_{1,t}) &= \frac{c_{t-1}(i) P(i, w_t, j)}{\sum_{k=0}^{|\mathcal{Q}|-1} c_{t-1}(k) P(k, w_t, j)} \\ H_t(j) &= \sum_{i=0}^{|\mathcal{Q}|-1} H_{t-1}(i) p(\theta_{t-1} = i | \theta_t = j, w_{1,t}) \\ &\quad - \sum_{i=0}^{|\mathcal{Q}|-1} p(\theta_{t-1} = i | \theta_t = j, w_{1,t}) \log p(\theta_{t-1} = i | \theta_t = j, w_{1,t}) \end{aligned} \tag{26}$$

Termination (to be completed).

Entropy measures for grammatical models



	a	a	b	a	
0.0					$H_t(0)$
1.0					$c_t(0)$
0.0					$H_t(1)$
0.0	.2	.06/.52=.115			$c_t(1)$
0.0				.8143	$H_t(2)$
0.0	.8	.24/.52=.462	.154	.735	$c_t(2)$
0.0			.438		$H_t(3)$
0.0		.06/.52=.115	.846		$c_t(3)$
0.0				.987	$H_t(4)$
0.0		.16/.52=.308		.265	$c_t(4)$

Entropy measures for grammatical models

Exercise (*): Define a PFA similar to the one that is in slide 15 and compute expressions (17)-(19). Choose a string with more than two derivations for (17).

Exercise (*): Compute the exact relation between the values in blue that are below expressions (17)-(19).

Exercise (**): Read [Hernando 2005]. Then, define a discrete HMM and compute $H_{\mathcal{A}}(\theta|w)$. The example, both the model and the string must be different from the example in [Hernando 2005].

Entropy measures for grammatical models

Exercise (***): (Practical exercise) Compute an approximation to expression (17) for an acyclic automaton $\mathcal{A}$ as follows:

1. Compute a list of n-best hyptheses ($\mathcal{B}$) for the automaton $\mathcal{A}$.
2. Compute $H_{\mathcal{A}}(\Theta)$ (19) (see [Sánchez 2018] for details).
3. Approximate $H_{\mathcal{A}}(\Theta|W) \approx -\sum_{w\in\mathcal{B}'} p_{\mathcal{A}}(w) H_{\mathcal{A}}(\theta|w)$ with the n-best hypotheses of step 1. $\mathcal{B}'$ is obtained from $\mathcal{B}$ by removing repetitions.
4. Approximate expression (17) as follows:

$$
\begin{aligned}
H_{\mathcal{A}}(\Theta|W) + H_{\mathcal{A}}(W) &= H_{\mathcal{A}}(W|\Theta) + H_{\mathcal{A}}(\Theta) \\
H_{\mathcal{A}}(\Theta|W) + H_{\mathcal{A}}(W) &= \sum_{\theta\in\Theta(\mathcal{A})} p_{\mathcal{A}}(\theta) H_{\mathcal{A}}(W|\theta) + H_{\mathcal{A}}(\Theta) \\
H_{\mathcal{A}}(\Theta|W) + H_{\mathcal{A}}(W) &= \sum_{\theta\in\Theta(\mathcal{A})} p_{\mathcal{A}}(\theta) \sum_{w\in L(\mathcal{A})} p_{\mathcal{A}}(w|\theta) \underbrace{\log p_{\mathcal{A}}(w|\theta)}_{\approx 0} + H_{\mathcal{A}}(\Theta) \\
H_{\mathcal{A}}(W) &= H_{\mathcal{A}}(\Theta) - H_{\mathcal{A}}(\Theta|W) \qquad (27)
\end{aligned}
$$

Index

Maximum entropy models

- **Problem**: estimate $p(y|x)$ where $x = (x_1, \ldots, x_D)$ is D-tuple of discrete (some times, categoric) observations

$$p(y|x_1, \ldots, x_D)$$

- **Maximum entropy solution:** log-linear combination

- **Pending problems:**
  - Selection of features
  - Smoothing: no closed-form solutions for optimal parameters

Maximum entropy models

Conditional entropy (see (4))

$$H(X|Y) = \sum_y p(y)\ H(X|Y=y) \tag{28}$$

$$= -\sum_x \sum_y p(x,y)\ \log p(x|y) \tag{29}$$

Mutual information (see (9))

$$I(X;Y) = \sum_x \sum_y p(x,y)\ \log \frac{p(x,y)}{p(x)\ p(y)} \tag{30}$$

Note that when X and Y are independent, Y can tell us nothing about X and then $I(X;Y) = 0$ in this case.

Maximum entropy models

- **Goal**: estimate p
- Choose p with maximum entropy (or "uncertainty" subject to some constraints)
- **Entropy** is a mathematic measure of the uniformity (uncertainty) of a distribution $p(x, y)$ (see (2))

$$H(p) = -\sum_{x,y} p(x, y) \log p(x, y) \tag{31}$$

- For a conditional distribution $p(y|x)$ its conditional entropy is (see (4))

$$H(p) = -\sum_{x,y} \widetilde{p}(x) p(y|x) \log p(y|x) \tag{32}$$

Maximum entropy models

- Collect (x, y) pairs from training data:
  - y: label to be predicted
  - x: the context
- Learn the probability p of each (x, y)
- Maximum entropy

  Model all that is known and assume nothing about what is unknown
  - to satisfy a set of constraints
  - to assume the most “uniform” distribution

Maximum entropy models

Example 1: MT

"We wish to model an expert translator's decisions concerning the proper French rendering of the English word **in**"

Let p our model of the translator's decisions and let (x, y) a set of samples: $\{(\textit{in}, \textit{dans}), (\textit{in}, \textit{en}), \ldots\}$

"Suppose that the expert translator always chooses among the following five French phrases: $\{\textit{dans}, \textit{en}, \textit{à}, \textit{au cours de}, \textit{pendenat}\}$"

$$p(\textit{dans}) + p(\textit{en}) + p(\textit{à}\ ) + p(\textit{au cours de}) + p(\textit{pendant}) \quad = \quad 1$$

Maximum entropy models

"Suppose we notice that the expert chose either *dans* or *en* 30% of the time"

$$\begin{aligned} p(\textit{dans}) + p(\textit{en}) &= 3/10 \\ p(\textit{dans}) + p(\textit{en}) + p(\textit{à}) + p(\textit{au cours de}) + p(\textit{pendant}) &= 1 \end{aligned}$$

". . . in half the cases, the expert chose either *dans* or *à*"

$$\begin{aligned} p(\textit{dans}) + p(\textit{en}) &= 3/10 \\ p(\textit{dans}) + p(\textit{en}) + p(\textit{à }) + p(\textit{au cours de}) + p(\textit{pendant}) &= 1 \\ p(\textit{dans}) + p(\textit{à }) &= 1/2 \end{aligned}$$

Maximum entropy models

Example 2: POS tagging

- Let say we have the following event space

NN	NNS	NNP	NNPS	VBZ	VBD

- Empirical data

3	5	11	13	3	1

- Maximize the entropy

$1/e$	$1/e$	$1/e$	$1/e$	$1/e$	$1/e$

- $E[\mathrm{NN}, \mathrm{NNS}, \mathrm{NNP}, \mathrm{NNPS}, V\mathrm{BZ}, \mathrm{VBD}] = 1$

$1/6$	$1/6$	$1/6$	$1/6$	$1/6$	$1/6$

Maximum entropy models

- N* are more common than V*. Add feature $f_N = \{\text{NN}, \text{NNS}, \text{NNP}, \text{NNPS}\}$ with $E[f_N] = 32/36$

NN	NNS	NNP	NNPS	VBZ	VBD
$8/36$	$8/36$	$8/36$	$8/36$	$2/36$	$2/36$

- Proper nouns are more frequent than common nouns. Add feature $f_P = \{\text{NNP}, \text{NNPS}\}$ with $E[f_P] = 24/36$

NN	NNS	NNP	NNPS	VBZ	VBD
$4/36$	$4/36$	$12/36$	$12/36$	$2/36$	$2/36$

Maximum entropy models

Problem: Construct a stochastic model that accurately represents the behaviour of a random process: $p(y|x)$.

Training data

Given a training sample (x, y), its empirical probability distribution $\widetilde{p}$ is defined by:

$$\widetilde{p}(x, y) \equiv \frac{1}{N} \times \text{number of times that } (x, y) \text{ occurs in the sample}$$

Features

If *April* is the word following *in*, then the translation of *in* is *en* with frequency $9/10$.

$$f(x, y) = \begin{cases} 1 & \text{if } y = \textit{en} \text{ and } \textit{April} \text{ follows } \textit{in} \\ 0 & \text{otherwise} \end{cases} \tag{33}$$

Maximum entropy models

Constraints

Expected value of f with respect to $\widetilde{p}(x, y)$:

$$\widetilde{p}(f) \equiv \sum_{x,y} \widetilde{p}(x, y) f(x, y) \tag{34}$$

Expected value of f with respect to the model $p(y|x)$:

$$p(f) \equiv \sum_{x,y} \widetilde{p}(x) p(y|x) f(x, y) \tag{35}$$

$$\sum_{x,y} \widetilde{p}(x) p(y|x) f(x, y) = \sum_{x,y} \widetilde{p}(x, y) f(x, y) \tag{36}$$

Maximum entropy models

Given n features f_i, we would like p lie in the subset $\mathcal{C}$ of $\mathcal{P}$ defined by

$$\mathcal{C} \equiv \{p \in \mathcal{P} \mid p(f_i) = \widetilde{p}(f_i) \text{ for } i \in \{1, 2, \ldots, n\}\} \tag{37}$$

Conditional entropy

$$H(p) = -\sum_{x,y} \widetilde{p}(x) p(y|x) \log p(y|x) \tag{38}$$

To select a model from a set $\mathcal{C}$ of allowed probability distributions, choose the model $p_* \in \mathcal{C}$ with maximum entropy $H(p)$:

$$p_* = \arg\max_{p \in \mathcal{C}} H(p) \tag{39}$$

Maximum entropy models

Solution to the primal problem:

http://www.cs.cmu.edu/afs/cs/user/aberger/www/html/tutorial/node7.html#SECTION00024000000000000000

The Maximum entropy solution p_* have the form:

$$p(y|x) = \frac{1}{Z(x)} \exp\left(\sum_{i=1}^{k} \lambda_i f_i(x,y)\right) \tag{40}$$

$$Z(x) = \sum_{y} \exp\left(\sum_{i=1}^{k} \lambda_i f_i(x,y)\right) \tag{41}$$

where k is the number of features
Let λ^* be

$$\lambda^* = \arg\max_{\lambda} \frac{1}{Z(x)} \exp\left(\sum_{i=1}^{k} \lambda_i f_i(x,y)\right) \tag{42}$$

Then $p_{\lambda^*} = p_*$

Maximum entropy models

Conditional likelihood of the training data (X, Y)

$$\log p(Y|X,\lambda) = \log \prod_{(x,y)\in(X,Y)} p(y|x,\lambda) \tag{43}$$

$$= \sum_{(x,y)\in(X,Y)} \log \frac{\exp\left(\sum_{i=1}^{k} \lambda_i f_i(x,y)\right)}{\sum_y \exp\left(\sum_{i=1}^{k} \lambda_i f_i(x,y)\right)} \tag{44}$$

$$= \sum_{(x,y)\in(X,Y)} \sum_{i=1}^{k} \lambda_i f_i(x,y) - \sum_{(x,y)\in(X,Y)} \log \sum_y \exp\left(\sum_{i=1}^{k} \lambda_i f_i(x,y)\right) \tag{45}$$

Solution: Compute each derivative independently and equal to 0

Maximum entropy models

$$\frac{\partial \sum_{(x,y)\in(X,Y)} \sum_{i=1}^{k} \lambda_i f_i(x,y)}{\partial \lambda_i} = \sum_{(x,y)\in(X,Y)} f_i(x,y) \tag{46}$$

$$\frac{\partial \sum_{(x,y)\in(X,Y)} \log \sum_{y} \exp\left(\sum_{i=1}^{k} \lambda_i f_i(x,y)\right)}{\partial \lambda_i} \tag{47}$$

$$= \sum_{(x,y)\in(X,Y)} \frac{1}{\sum_{y''} \exp\left(\sum_{i=1}^{k} \lambda_i f_i(x,y'')\right)} \sum_{y'} \exp\left(\sum_{i=1}^{k} \lambda_i f_i(x,y')\right) \frac{\partial \sum_{i=1}^{k} \lambda_i f_i(x,y')}{\partial \lambda_i} \tag{48}$$

$$= \sum_{(x,y)\in(X,Y)} \sum_{y'} \frac{\exp\left(\sum_{i=1}^{k} \lambda_i f_i(x,y')\right)}{\sum_{y''} \exp\left(\sum_{i=1}^{k} \lambda_i f_i(x,y'')\right)} \frac{\partial \sum_{i=1}^{k} \lambda_i f_i(x,y')}{\partial \lambda_i} = \sum_{(x,y)\in(X,Y)} \sum_{y'} p_\lambda(y' \mid x) f_i(x,y') \tag{49}$$

The optimum is achieved when (see Eq. (36)):

$$\sum_{(x,y)\in(X,Y)} f_i(x,y) = \sum_{(x,y)\in(X,Y)} \sum_{y'} p_\lambda(y' \mid x) f_i(x,y') \tag{50}$$

Maximum entropy models

Solution to the dual problem: IIS algorithm Details here and here

1. Start with some (arbitrary) value for each λ_i
2. Repeat until convergence:

$$\text{(a) Solve} \quad \frac{\partial \mathcal{B}(\delta)}{\partial \delta_i} = 0 \quad \text{for} \quad \delta_i \tag{51}$$

(b) Set $\lambda_i = \lambda_i + \delta_i$

where:

$$\frac{\partial \mathcal{B}(\delta)}{\partial \delta_i} = \overbrace{\sum_{x,y} \widetilde{p}(x,y) f_i(x,y)}^{\widetilde{p}(f_i)} - \overbrace{\sum_{x,y} \widetilde{p}(x) p_\lambda(y|x) f_i(x,y)}^{p_\lambda(f_i)} e^{\delta_i f^\#(x,y)} \tag{52}$$

If $f^\#(x,y) = M$ then

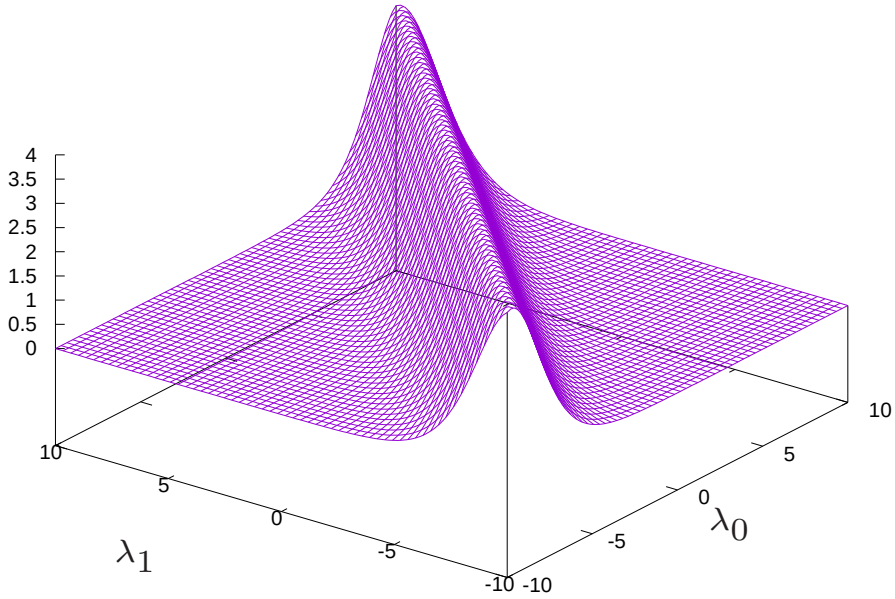
$$\delta_i = \frac{1}{M} \log \frac{\widetilde{p}(f_i)}{p_\lambda(f_i)} \tag{53}$$

Maximum entropy models

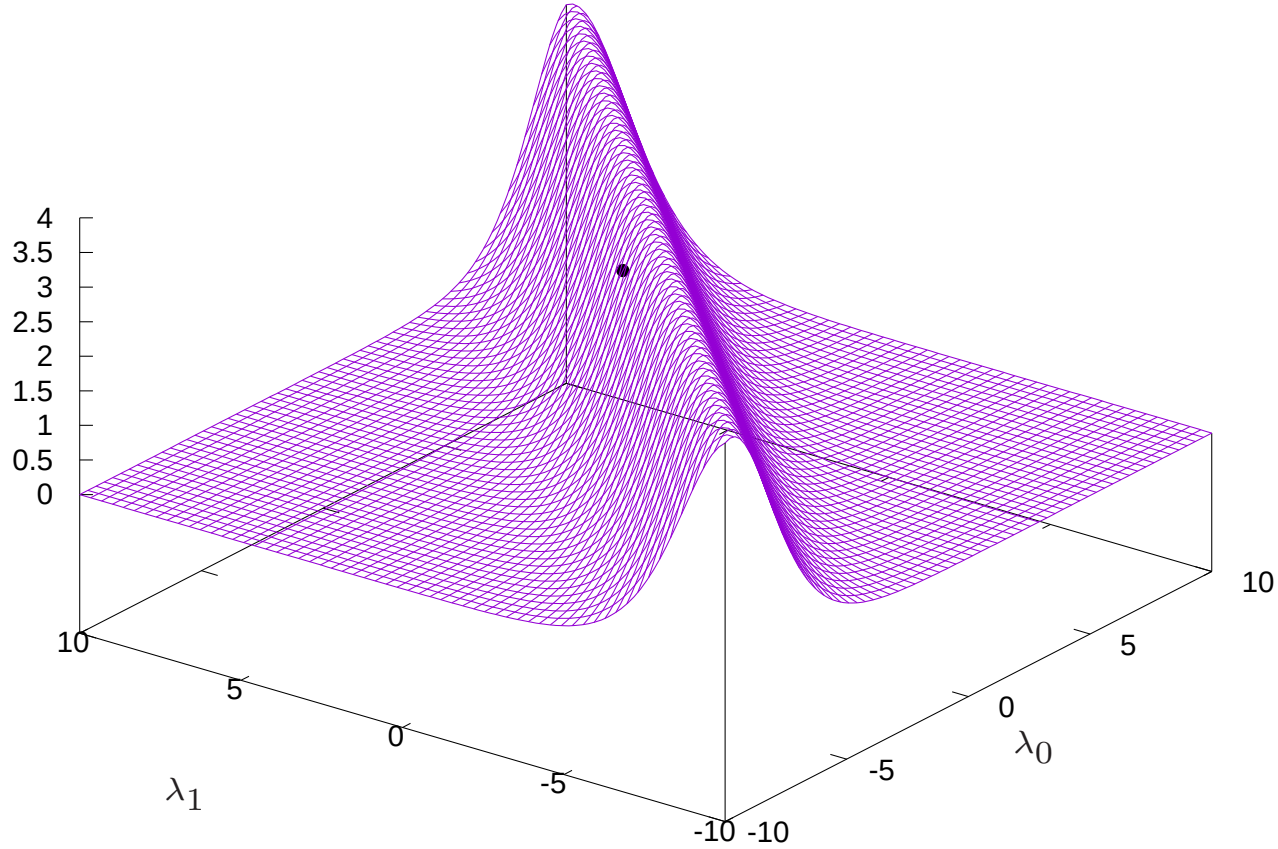
Example to illustrate de optimization problem (just two features)

$$C = \{(w_0, c_0), (w_0, c_0), (w_0, c_0), (w_1, c_1), (w_1, c_1), (w_1, c_1), (w_1, c_1), (w_1, c_1), (w_1, c_1), (w_1, c_1)\}$$
$$\lambda_0 = f(w_0, c_0) \quad \lambda_1 = f(w_1, c_1)$$

$H(p)$ with $\lambda_0 = 0, \lambda_1 = 0$



$H(p)$ with $\lambda_0 = 0, \lambda_1 = 0 \rightarrow H(p) = 2.84, \lambda_0 = .69, \lambda_1 = .69$



Maximum entropy models. Smoothing

- Lots of features
- Lots of sparsity
- Overfitting very easy
- Many features at test time could not be seen in training time
- Feature weights can be infinite and optimizer can take a long time to get to those infinites
- Smoothing is needed!

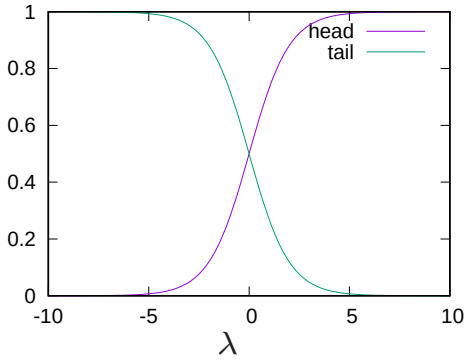
Maximum entropy models. Smoothing

- Asume the following empirical distribution: a coin with 2 faces and identical probability
- Features: head, tail
- Model distribution

$$p_h = \frac{e^{\lambda_h}}{e^{\lambda_h} + e^{\lambda_t}} \qquad p_t = \frac{e^{\lambda_t}}{e^{\lambda_h} + e^{\lambda_t}}$$

- Only one degree of freedom: $\lambda = \lambda_h - \lambda_t$

$$p_h = \frac{e^{\lambda_h} e^{-\lambda_t}}{e^{\lambda_h} e^{-\lambda_t} + e^{\lambda_t} e^{-\lambda_t}} = \frac{e^{\lambda}}{e^{\lambda} + e^{0}} = \frac{e^{\lambda}}{e^{\lambda} + 1} \qquad p_t = \frac{1}{1 + e^{\lambda}}$$

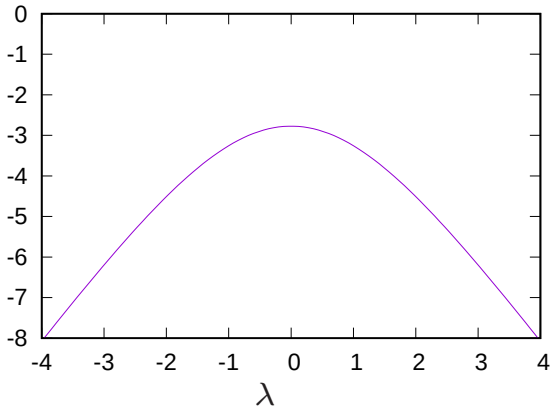


Maximum entropy models. Smoothing

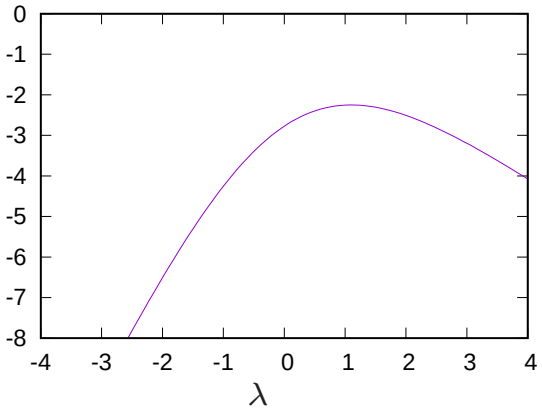
- For a sample with h and t, the log likelihood is:

$$\begin{aligned} h \log p_h + t \log p_t &= h(\lambda - \log(1 + e^{\lambda})) + t(\log 1 - \log(1 + e^{\lambda})) \\ &= h\lambda - h \log(1 + e^{\lambda})) + t \log(1 + e^{\lambda}) \\ &= h\lambda - (t + h) \log(1 + e^{\lambda})) \end{aligned}$$

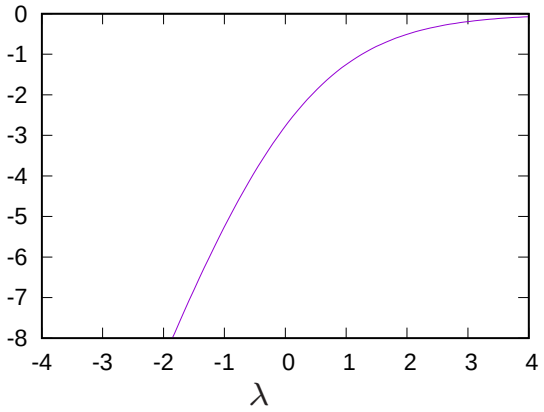
$h = 2 \quad t = 2$



$h = 3 \quad t = 1$



$h = 4 \quad t = 0$



- λ goes to $\infty \longrightarrow$ long time to convergence

Maximum entropy models. Smoothing

- To include a prior on the parameter values and change the optimization objective:

$$\log p(Y, \lambda \mid X) \approx \log p(\lambda) + \log p(Y \mid X, \lambda)$$

- Gaussian prior: each parameter will be distributed according to a gaussian mean μ and variance σ^2.
- Usually $\mu = 0$ and $2\sigma^2 = 1$
- New optimization function

$$\log p(Y, \lambda \mid X) = \sum_{(x,y)\in(X,Y)} p(y \mid x, \lambda) - \sum_i \frac{(\lambda_i - \mu_i)^2}{2\sigma_i^2} + k \qquad (54)$$

- Change the derivative

$$\sum_{(x,y)\in(X,Y)} f_i(x, y) - \sum_{(x,y)\in(X,Y)} \sum_{y'} p_\lambda(y' \mid x) f_i(x, y') - \frac{(\lambda_i - \mu_i)}{\sigma_i^2} \qquad (55)$$

Maximum entropy models

Exercise 1. Given a set of words, such that each word has associated a class label c_0 or c_1:

$$\begin{aligned} C \quad = \quad & \{(w_0, c_0), (w_0, c_0), (w_1, c_0), (w_1, c_0), (w_1, c_0), (w_1, c_0), \\ & (w_0, c_1), (w_2, c_1), (w_2, c_1), (w_2, c_1)\} \end{aligned}$$

Obtain a maximum entropy model for classifying a text containg those words. The following features have been defined for this purpose:

$$f(x, y) = \begin{cases} 1 & \text{if } y = c_i \text{ and it has the word } y = w_j \\ 0 & \text{otherwise} \end{cases}$$

1. Suppose that all λ_i are initialized to 0, then compute the increasing δ_0 for the feature $f_0 = f(w_0, c_0)$ with IIS algorithm.
2. Suppose that a ME model has been learned and the following values have been obtained:

$$\begin{aligned} \lambda_0 = \quad & 0.3 \text{ associated to the feature } f_0 = f(w_0, c_0), \\ \lambda_1 = \quad & 3.0 \text{ associated to the feature } f_1 = f(w_1, c_0), \\ \lambda_2 = -& 0.4 \text{ associated to the feature } f_2 = f(w_0, c_1), \\ \lambda_3 = \quad & 3.0 \text{ associated to the feature } f_3 = f(w_2, c_1). \end{aligned}$$

Compute in which class is classified a text if it contains the word w_0.

Maximum entropy models

Solution

a)

$$\widetilde{p}(w_0, c_0) = \frac{2}{10} = \frac{1}{5}, \quad \widetilde{p}(w_0) = \frac{3}{10} \quad \text{y} \quad p_\lambda(c_0|w_0) = \frac{1}{2}$$

Then:

$$\delta_0 = \log \frac{\frac{1}{5}}{\frac{3}{10}\frac{1}{2}} = \log \frac{4}{3} = 0.288$$

b)

$$p_\lambda(c_0|w_0) = \frac{\exp(0.3)}{\exp(0.3) + \exp(-0.4)} = 0.668$$

$$p_\lambda(c_1|w_0) = \frac{\exp(-0.4)}{\exp(0.3) + \exp(-0.4)} = 0.332$$

and therefore that sample is classified in class c_0.

Exercise (*): Suppose that all λ_i are initialized to 0, then compute the increasing δ_{21} for the feature $f_{21} = f(w_2, c_1)$ with the IIS algorithm. Compute in which class is classified a text if it contains the words w_0, w_0, w_2.

Maximum entropy models

Exercise 2. Let a classification problem into three classes A, B and C, where the classification is carried out with a Maximum Entropy classifier. The classification of each samples is performed according to 3 features out of 5, noted as $(c_0, c_1, c_2, c_3, c_4)$.

The features of the Maximum Entropy classifier are defined as:

$$f(y, x) = \begin{cases} 1 & \text{if } y = S \text{ and the feature } c_j \text{ is active in } x \\ 0 & \text{otherwise} \end{cases}$$

where $S \in \{A, B, C\}$.

Supose that a model has been trained and the obtained parameters are:

$$\begin{array}{lllll} \lambda_{A,c_0} = 0.37 & \lambda_{A,c_1} = 0.0 & \lambda_{A,c_2} = -0.04 & \lambda_{A,c_3} = 0.08 & \lambda_{A,c_4} = 0.0 \\ \lambda_{B,c_0} = 0.0 & \lambda_{B,c_1} = 0.0 & \lambda_{B,c_2} = -0.04 & \lambda_{B,c_3} = -0.05 & \lambda_{B,c_4} = -0.28 \\ \lambda_{C,c_0} = 0.0 & \lambda_{C,c_1} = 0.23 & \lambda_{C,c_2} = 0.06 & \lambda_{C,c_3} = -0.05 & \lambda_{C,c_4} = 0.32 \end{array}$$

Compute the class of a sample if the features c_0, c_2 and c_3 has been observed.

Maximum entropy models

Solution

$$p_\lambda(A|(c_0,c_2,c_3)) = \frac{\exp(0.37-0.04+0.08)}{exp(0.37-0.04+0.0)+\exp(0.0-0.04-0.05)+\exp(0.0+0.06-0.0)} = 0.43$$

$$p_\lambda(B|(c_0,c_2,c_3)) = \frac{\exp(0.0-0.04-0.05)}{exp(0.37-0.04+0.0)+\exp(0.0-0.04-0.05)+\exp(0.0+0.06-0.0)} = 0.27$$

$$p_\lambda(C|(c_0,c_2,c_3)) = \frac{\exp(0.0+0.06-0.05)}{exp(0.37-0.04+0.0)+\exp(0.0-0.04-0.05)+\exp(0.0+0.06-0.0)} = 0.30$$

and therefore the samples is classified in class A.

Maximum entropy models

Available ME tools:

- MALLET: http://mallet.cs.umass.edu/
- NLTK: http://www.nltk.org/
- WEKA: http://www.cs.waikato.ac.nz/ml/weka/

Recommended web pages:

- http://www.cs.cmu.edu/afs/cs/user/aberger/www/html/tutorial/tutorial.html
- http://en.wikipedia.org/wiki/Principle_of_maximum_entropy
- http://www.inference.phy.cam.ac.uk/hmw26/crf/

Exercise (***): Choose a toolkit and a dataset and develop a classifier trained with that toolkit. The mark on this exercise will depend on the obtained results compared with other state-of-the-art results.

Exercise (**): Consider the set of sentences: $S = \{(\text{"a a"}, C_0), (\text{"b b"}, C_0), (\text{"a b"}, C_1), (\text{"b a"}, C_1)\}$. Compute a Maximum Entropy classifier with features $f_{C_i,x}$ with $i \in \{1, 2\}, x \in \{a, b\}$ and compute the classification error. Describe carefully the conclusion that you get and justify them.

Exercise (***): Modify mallet to compute (55) and report experiments on a dataset.

Maximum entropy models

Feature selection: Information Gain. The idea is to use the Mutual Information for choosing the features

$$I(X;Y) = H(Y) - H(Y|X) = H(Y) - \sum_x p(X=x)H(Y|X=x)$$

Choose those features that reduce the class entropy, H(Y).

Example. Supose that we have two type of texts A and B, and each text has only one word. We have 6 instances of text of type A: $\{a, a, a, a, b, c\}$; and 4 instances of text of type B: $\{a, b, b, c\}$.

$$H(Y) = -\sum_{A,B} p(Y)\log p(Y) = -3/5\log 3/5 - 2/5\log 2/5 = .97 \text{ bits}$$

$$p(X=a) = \frac{1}{2} \qquad p(A|a) = \frac{p(A,a)}{p(a)} = \frac{4}{5} \qquad p(B|a) = \frac{p(B,a)}{p(a)} = \frac{1}{5}$$

$$p(X=b) = \frac{3}{10} \qquad p(A|b) = \frac{p(A,b)}{p(b)} = \frac{1}{3} \qquad p(B|b) = \frac{p(B,b)}{p(b)} = \frac{2}{3}$$

$$p(X=c) = \frac{1}{5} \qquad p(A|c) = \frac{p(A,c)}{p(c)} = \frac{1}{2} \qquad p(B|c) = \frac{p(B,c)}{p(c)} = \frac{1}{2}$$

Maximum entropy models

The words can be sorted according to entropy $H(Y)$ reduction:

$$-p(X=x)(p(Y=A|X=x)\log p(Y=A|X=x) + p(Y=B|X=x)\log p(Y=B|X=x))$$

$$X=a \longrightarrow -\frac{1}{2}\left(\frac{4}{5}\log\frac{4}{5}+\frac{1}{5}\log\frac{1}{5}\right)=.25$$

$$X=b \longrightarrow -\frac{3}{10}\left(\frac{1}{3}\log\frac{1}{3}+\frac{2}{3}\log\frac{2}{3}\right)=.19$$

$$X=c \longrightarrow -\frac{1}{5}\left(\frac{1}{2}\log\frac{1}{2}+\frac{1}{2}\log\frac{1}{2}\right)=.13$$

Application: Extracting Descriptive Words from Untranscribed Handwritten Images

Index

EM algorithm

Goal: find θ such that $\mathcal{P}(\mathbf{X}|\theta)$ is maximum (Maximum Likelihood estimate for θ)

Log likelihood function:

$$L(\theta) = \ln \mathcal{P}(\mathbf{X}|\theta). \tag{56}$$

Supose that after iteration n^{th} the current parameters are θ_n and we want to estimate these parameters. Then, we want to compute an updated estimate θ such that,

$$L(\theta) > L(\theta_n). \tag{57}$$

or to maximize the difference

$$L(\theta) - L(\theta_n) = \ln \mathcal{P}(\mathbf{X}|\theta) - \ln \mathcal{P}(\mathbf{X}|\theta_n). \tag{58}$$

EM algorithm

Denote the hidden random vector by $\mathbf{Z}$ and a given realization by $\mathbf{z}$:

$$\mathcal{P}(\mathbf{X}|\theta) = \sum_{\mathbf{z}} \mathcal{P}(\mathbf{X}, \mathbf{z}|\theta) = \sum_{\mathbf{z}} \mathcal{P}(\mathbf{X}|\mathbf{z}, \theta)\mathcal{P}(\mathbf{z}|\theta). \tag{59}$$

Then, Equation (58) can be rewritten as:

$$L(\theta) - L(\theta_n) = \ln\left(\sum_{\mathbf{z}} \mathcal{P}(\mathbf{X}|\mathbf{z}, \theta)\mathcal{P}(\mathbf{z}|\theta)\right) - \ln \mathcal{P}(\mathbf{X}|\theta_n). \tag{60}$$

EM algorithm

Then, from Equation (7):

$$
\begin{aligned}
L(\theta) - L(\theta_n) &= \ln\left(\sum_{\mathbf{z}} \mathcal{P}(\mathbf{X}|\mathbf{z},\theta)\mathcal{P}(\mathbf{z}|\theta)\right) - \ln \mathcal{P}(\mathbf{X}|\theta_n) && (61)\\
&= \ln\left(\sum_{\mathbf{z}} \mathcal{P}(\mathbf{X}|\mathbf{z},\theta)\mathcal{P}(\mathbf{z}|\theta)\frac{\mathcal{P}(\mathbf{z}|\mathbf{X},\theta_n)}{\mathcal{P}(\mathbf{z}|\mathbf{X},\theta_n)}\right) - \ln \mathcal{P}(\mathbf{X}|\theta_n) && (62)\\
&= \ln\left(\sum_{\mathbf{z}} \mathcal{P}(\mathbf{z}|\mathbf{X},\theta_n)\frac{\mathcal{P}(\mathbf{X}|\mathbf{z},\theta)\mathcal{P}(\mathbf{z}|\theta)}{\mathcal{P}(\mathbf{z}|\mathbf{X},\theta_n)}\right) - \ln \mathcal{P}(\mathbf{X}|\theta_n) && (63)\\
&\geq \sum_{\mathbf{z}} \mathcal{P}(\mathbf{z}|\mathbf{X},\theta_n) \ln\left(\frac{\mathcal{P}(\mathbf{X}|\mathbf{z},\theta)\mathcal{P}(\mathbf{z}|\theta)}{\mathcal{P}(\mathbf{z}|\mathbf{X},\theta_n)}\right) - \ln \mathcal{P}(\mathbf{X}|\theta_n) && (64)\\
&= \sum_{\mathbf{z}} \mathcal{P}(\mathbf{z}|\mathbf{X},\theta_n) \ln\left(\frac{\mathcal{P}(\mathbf{X}|\mathbf{z},\theta)\mathcal{P}(\mathbf{z}|\theta)}{\mathcal{P}(\mathbf{z}|\mathbf{X},\theta_n)}\right) - \sum_{\mathbf{z}} \mathcal{P}(\mathbf{z}|\mathbf{X},\theta_n) \ln \mathcal{P}(\mathbf{X}|\theta_n) && (65)\\
&= \sum_{\mathbf{z}} \mathcal{P}(\mathbf{z}|\mathbf{X},\theta_n) \ln\left(\frac{\mathcal{P}(\mathbf{X}|\mathbf{z},\theta)\mathcal{P}(\mathbf{z}|\theta)}{\mathcal{P}(\mathbf{z}|\mathbf{X},\theta_n)\mathcal{P}(\mathbf{X}|\theta_n)}\right) && (66)\\
&\triangleq \Delta(\theta|\theta_n) && (67)
\end{aligned}
$$

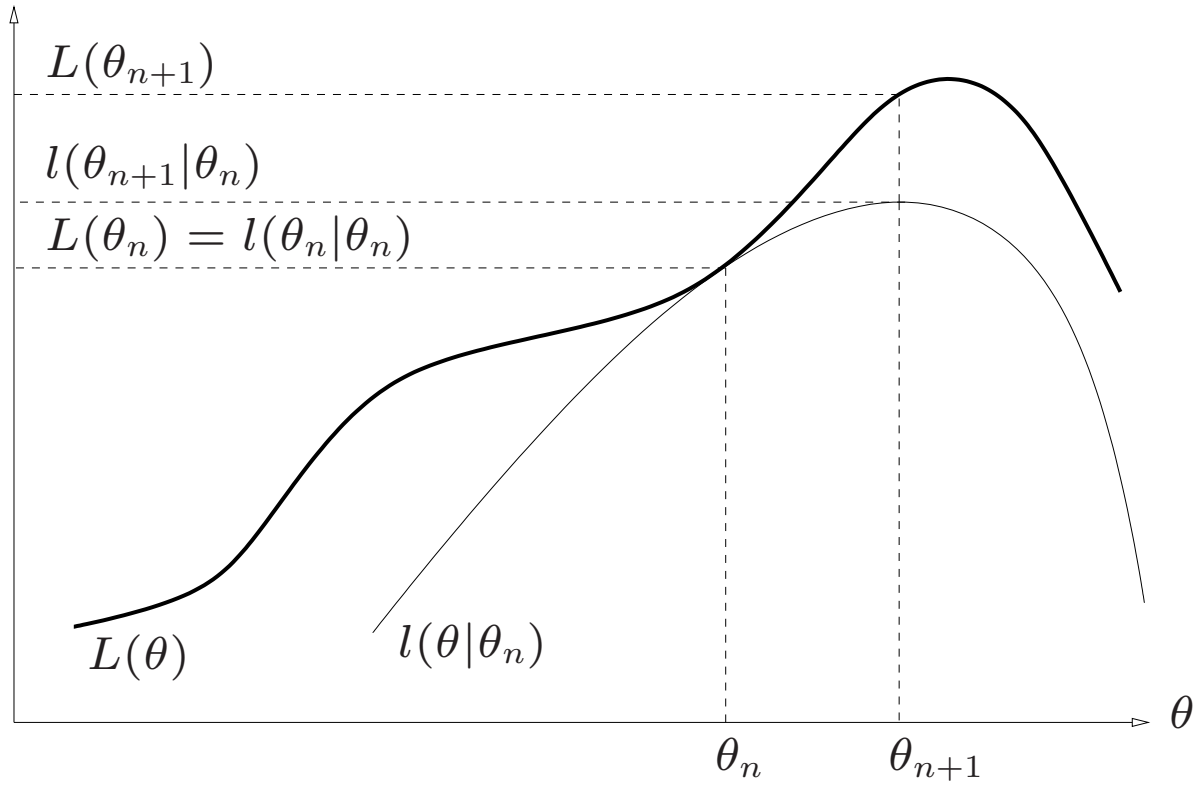
EM algorithm

Then: $$L(\theta) \geq L(\theta_n) + \Delta(\theta|\theta_n).$$

If we define:

$$l(\theta|\theta_n) \triangleq L(\theta_n) + \Delta(\theta|\theta_n),$$ note taht $L(\theta_n)$ is a known value

then $$L(\theta) \geq l(\theta|\theta_n).$$



EM algorithm

Since $l(\theta|\theta_n)$ is bounded above by $L(\theta)$, any θ that increases $l(\theta|\theta_n)$ will also increase $L(\theta)$.

$$\theta_{n+1} = \arg\max_{\theta} \{l(\theta|\theta_n)\} \tag{68}$$

$$= \arg\max_{\theta} \left\{ L(\theta_n) + \sum_{\mathbf{z}} \mathcal{P}(\mathbf{z}|\mathbf{X},\theta_n) \ln \frac{\mathcal{P}(\mathbf{X}|\mathbf{z},\theta)\mathcal{P}(\mathbf{z}|\theta)}{\mathcal{P}(\mathbf{z}|\mathbf{X},\theta_n)\mathcal{P}(\mathbf{X}|\theta_n)} \right\} \tag{69}$$

$$= \arg\max_{\theta} \left\{ \sum_{\mathbf{z}} \mathcal{P}(\mathbf{z}|\mathbf{X},\theta_n) \ln \mathcal{P}(\mathbf{X}|\mathbf{z},\theta)\mathcal{P}(\mathbf{z}|\theta) \right\} \tag{70}$$

$$= \arg\max_{\theta} \left\{ \sum_{\mathbf{z}} \mathcal{P}(\mathbf{z}|\mathbf{X},\theta_n) \ln \mathcal{P}(\mathbf{X},\mathbf{z}|\theta) \right\} \text{ (Usually called } \mathcal{Q}(\theta|\theta_n)) \tag{71}$$

$$= \arg\max_{\theta} \left\{ \mathbf{E}_{\mathbf{Z}|\mathbf{X},\theta_n} \{\ln \mathcal{P}(\mathbf{X},\mathbf{z}|\theta)\} \right\} \tag{72}$$

1. *E-step*: Determine $\mathcal{P}(\mathbf{Z}|\mathbf{X},\theta_n)$
2. *M-step*: Maximize (72) with respecto to θ.

EM algorithm: example with a Gaussian mixture

Simplication: unidimensional Gaussian mixture

$$p(\mathbf{x}) = \sum_{\mathbf{z}} p(\mathbf{x}, \mathbf{z}) \tag{73}$$

Let supose that $\mathbf{z}$ is a K-dimensional binary random variable where some value $\mathbf{z}_k$ is equal to 1 and all the other values are equal to 0. Variable $\mathbf{z}$ has K possible states.

$$p(\mathbf{z}_k = 1) = \pi_k \ , \qquad \sum_{k=1}^{K} \pi_k = 1$$

Alternative notation:

$$p(\mathbf{z}) = \prod_{k=1}^{K} \pi_k^{\mathbf{z}_k}$$

In addition

$$p(\mathbf{x}|\mathbf{z}_k = 1) = p(\mathbf{x}|\mu_k, \sigma_k)$$

or

$$p(\mathbf{x}|\mathbf{z}) = \prod_{k=1}^{K} p(\mathbf{x}|\mu_k, \sigma_k)^{\mathbf{z}_k}$$

EM algorithm: example with a Gaussian mixture

In this way:

$$
\begin{aligned}
p(\mathbf{x}) &= \sum_{\mathbf{z}} p(\mathbf{z}) p(\mathbf{x}|\mathbf{z}) = \sum_{k=1}^{K} \pi_k \; p(\mathbf{x}|\mu_k, \sigma_k) \\
&= \sum_{k=1}^{K} \pi_k \; \frac{1}{\sigma_k \sqrt{2\pi}} \; e^{-\frac{(\mathbf{x}-\mu_k)^2}{2\sigma_k^2}}
\end{aligned}
\tag{74}
$$

Given a sample S, the $\log$ likelihood is defined as:

$$
L_S(\theta) = \ln \prod_{m=1}^{M} p(\mathbf{x}_m|\theta) \tag{75}
$$

$$
= \sum_{m=1}^{M} \ln \sum_{k=1}^{K} \pi_k \; p(\mathbf{x}_m|\mu_k, \sigma_k) \tag{76}
$$

EM algorithm: example with a Gaussian mixture

Simplication: unidimensional Gaussian mixture with unknown parameters π_k. The other parameters are all known and constant along all iterations (see [Bishop 2006] pp. 430–443).

1. *E-step*:

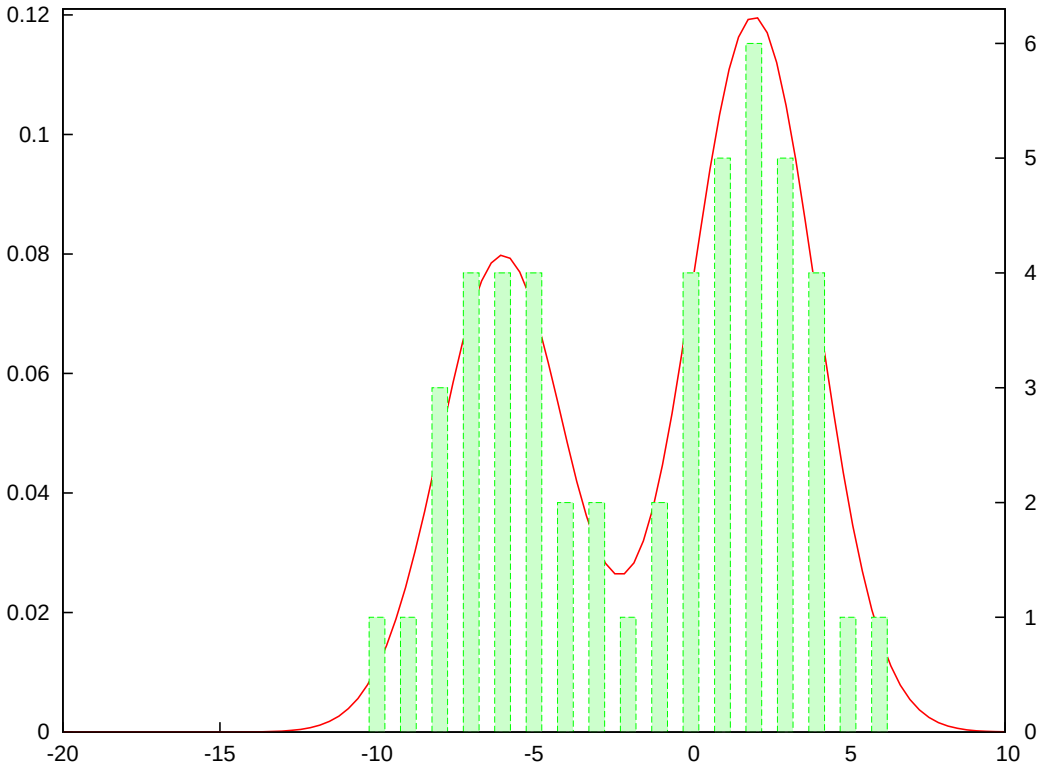
$$\hat{z}_{km} = \frac{\pi_k^{(t)}\ p(\mathbf{x}_m|\theta_k)}{\sum_{k'} \pi_{k'}^{(t)}\ p(\mathbf{x}_m|\theta_{k'})} \tag{77}$$

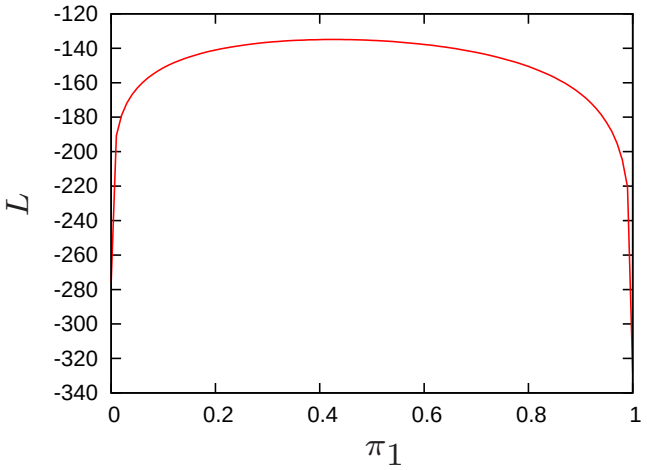
2. *M-step*:

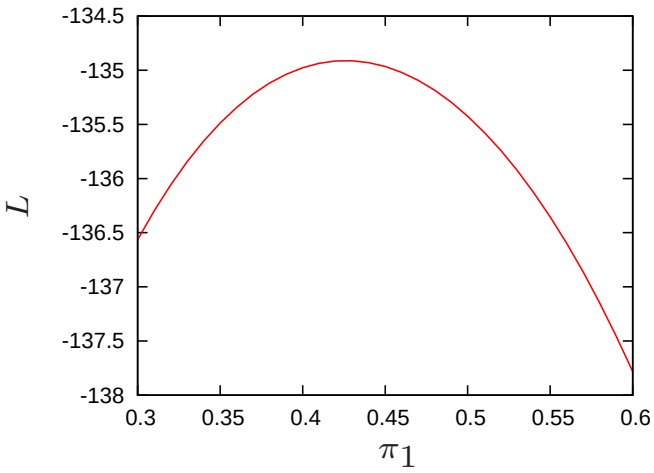
$$\pi_k^{(t+1)} = \frac{\sum_{m=1}^{M} \hat{z}_{km}}{M} \tag{78}$$

EM algorithm: example with a Gaussian mixture

Let a gaussian mixture of two unidimensional distributions, with known mean (-6 and 2) and equal and known variance (4) where only π_1, π_2 (0.4 and 0.6) are unknown. Let supose that a training sample of 50 unidimensional vectors is available:

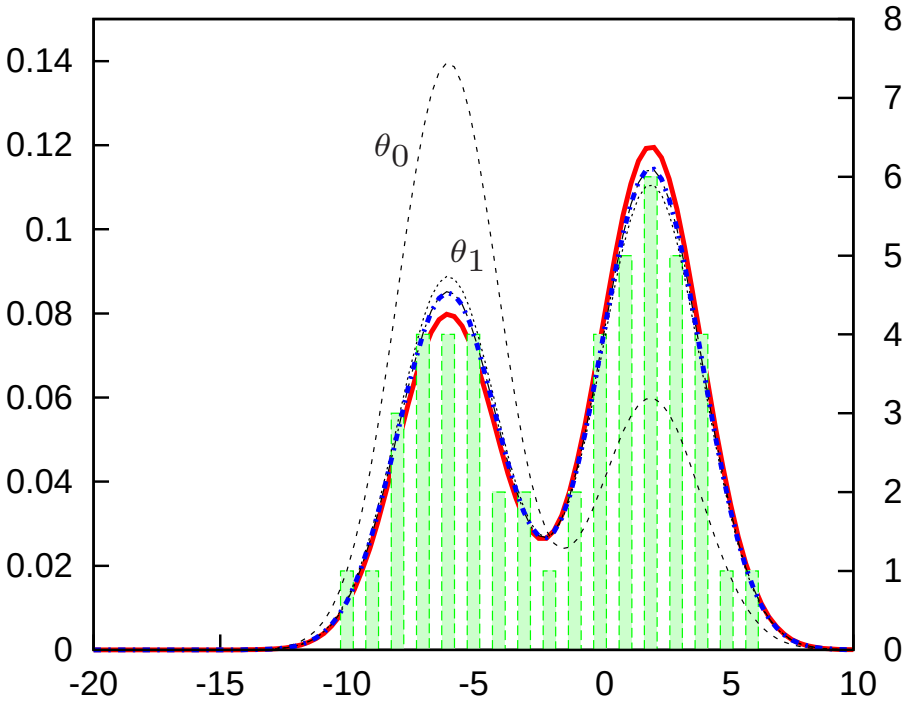




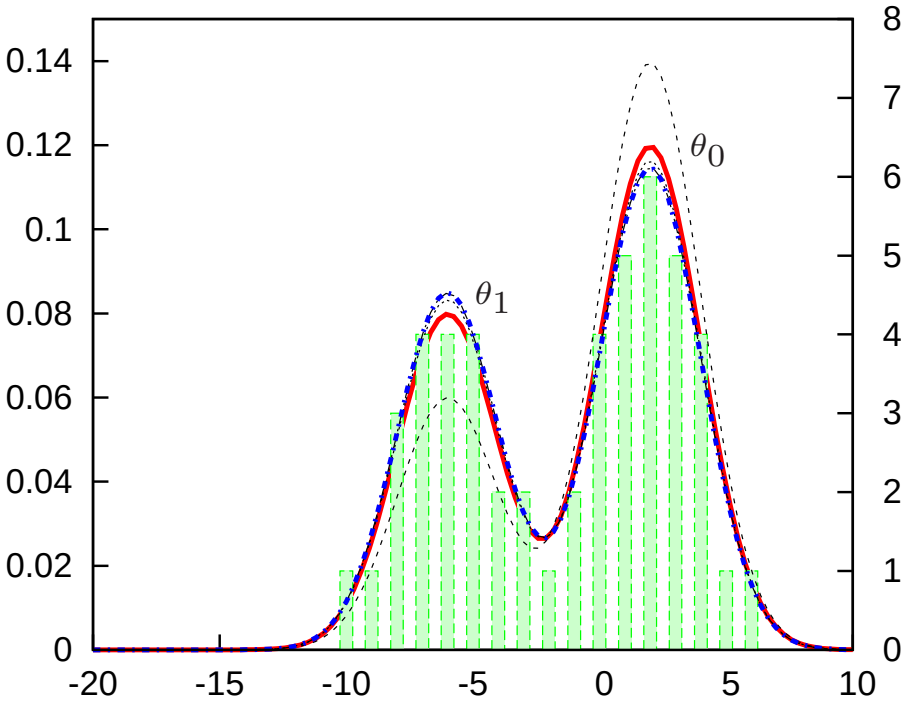


EM algorithm: example with a Gaussian mixture

$$\theta^{(0)} = (\pi_1 = 0.700, \pi_2 = 0.300)$$
$$\theta^{(1)} = (\pi_1 = 0.445, \pi_2 = 0.555)$$
$$\theta^{(2)} = (\pi_1 = 0.427, \pi_2 = 0.573)$$
$$\theta^{(3)} = (\pi_1 = 0.426, \pi_2 = 0.574)$$
$$\theta^{(4)} = (\pi_1 = 0.426, \pi_2 = 0.574)$$



$$\theta^{(0)} = (\pi_1 = 0.300, \pi_2 = 0.700)$$
$$\theta^{(1)} = (\pi_1 = 0.417, \pi_2 = 0.583)$$
$$\theta^{(2)} = (\pi_1 = 0.425, \pi_2 = 0.575)$$
$$\theta^{(3)} = (\pi_1 = 0.426, \pi_2 = 0.574)$$
$$\theta^{(4)} = (\pi_1 = 0.426, \pi_2 = 0.574)$$



EM algorithm: example with a Gaussian mixture

Exercise (**): Reproduce an example similar to the previous example with three unidimensional distributions, with known mean and equal and known variance where only π_1, π_2, π_3 are unknown. But the data have to be generated with two gaussians. Obtain the corresponding plots.

Exercise (**): Reproduce an example similar to the previous example with three unidimensional distributions, with known mean and equal and known variance where only π_1, π_2, π_3 are unknown. The data have to be generated with three gaussians. Obtain the corresponding plots.

Exercise (**): Reproduce an example similar to the previous example with two bidimensional distributions, with known mean vector and equal and known variance vector where only π_1, π_2, π_3 are unknown. Obtain the corresponding plots.

Exercise (***): Reproduce an example similar to the previous example with two unidimensional distributions, with equal and known variance where the mean of each distribtuion and π_1, π_2 are unknown. Obtain the corresponding plots.

Exercise (***): Reproduce an example similar to the previous example with two unidimensional distributions, with equal and known mean where the variances and π_1, π_2 are unknown. Obtain the corresponding plots.

EM algorithm: Regularization

Motivation: To choose a suitable missing data.
"This question is not addressed in the EM algorithm because the likelihood function does not reflect any influence of the missing data." [Li, 2005]

Approach: *"To regularize the likelihood function with a suitable functional of the distribution of the complete data."*

Regularizated EM:

$$\tilde{L}(\mathbf{X}|\theta) = L(\mathbf{X}|\theta) + \gamma\, R(\mathbf{X}, \mathbf{Z}|\theta) \tag{79}$$

Ideal situation: the missing data has little uncertainty given the observations $\longrightarrow$ Mutual information

Regularization function: Mutual information

$$\tilde{L}(\mathbf{X}|\theta) = L(\mathbf{X}|\theta) + \gamma\, I(\mathbf{X}, \mathbf{Z}|\theta) \tag{80}$$

If we assume that $\mathbf{Z}$ follows a uniform distribution $(I(\mathbf{X};\mathbf{Z}) = H(\mathbf{Z}) - H(\mathbf{Z}|\mathbf{X}))$:

$$\tilde{L}(\mathbf{X}|\theta) = L(\mathbf{X}|\theta) - \gamma\, H(\mathbf{Z}|\mathbf{X};\theta) \tag{81}$$

EM algorithm: Regularization

For a Gaussian mixture:

$$\tilde{L}(\theta;\mathbf{X}) = \sum_{m=1}^{M} \ln \sum_{k=1}^{K} \pi_k \; p(\mathbf{x}_m|\mu_k,\sigma_k)$$

$$+\gamma \underbrace{\int p(\mathbf{x}|\mu,\sigma) \sum_{k=1}^{K} \frac{\pi_k \; p(\mathbf{x}|\mu_k,\sigma_k)}{p(\mathbf{x}|\mu,\sigma)} \; \ln\left(\frac{\pi_k \; p(\mathbf{x}|\mu_k,\sigma_k}{p(\mathbf{x}|\mu,\sigma)}\right) \; d\mathbf{x}}_{\text{See (4)}} \qquad (82)$$

$$\tilde{\mathcal{Q}}(\theta|\theta^{(t)}) = \sum_{k=1}^{K} \sum_{m=1}^{M} p(\mathbf{z}_k|\mathbf{x}_m,\mu^{(t)},\sigma^{(t)}) \; \ln \pi_k$$

$$+\sum_{k=1}^{K} \sum_{m=1}^{M} p(\mathbf{z}_k|\mathbf{x}_m,\mu^{(t)},\sigma^{(t)}) \; \ln p(\mathbf{x}_m|\mu_k,\sigma_k) \quad \text{(See (66))}$$

$$+\gamma \int p(\mathbf{x}|\mu,\sigma) \sum_{k=1}^{K} \frac{\pi_k \; p(\mathbf{x}|\mu_k,\sigma_k)}{p(\mathbf{x}|\mu,\sigma)} \; \ln\left(\frac{\pi_k \; p(\mathbf{x}|\mu_k,\sigma_k}{p(\mathbf{x}|\mu,\sigma)}\right) \; d\mathbf{x} \qquad (83)$$

EM algorithm: Regularization

Introducing a Lagrangian and solving [Li, 2005] (Compare with (77) and (78)):

$$\pi_k^{(t+1)} = \frac{\sum_{m=1}^{M} p(\mathbf{z}_k|\mathbf{x}_m, \mu^{(t)}, \sigma^{(t)})(1 + \gamma \ \ln p(\mathbf{z}_k|\mathbf{x}_m, \mu^{(t)}, \sigma^{(t)}))}{\sum_{m=1}^{M}\sum_{k=1}^{K} p(\mathbf{z}_k|\mathbf{x}_m, \mu^{(t)}, \sigma^{(t)})(1 + \gamma \ \ln p(\mathbf{z}_k|\mathbf{x}_m, \mu^{(t)}, \sigma^{(t)}))} \quad (84)$$

where

$$p(\mathbf{z}_k|\mathbf{x}_m, \mu^{(t)}, \sigma^{(t)}) = \frac{\pi_k^{(t)} \ p(\mathbf{x}_m|\theta_k^{(t)})}{\sum_{k'} \pi_{k'}^{(t)} \ p(\mathbf{x}_m|\theta_{k'}^{(t)})} \quad (85)$$

The other parameters:

$$\mu_k^{(t+1)} = \frac{\sum_{m=1}^{M} \mathbf{x}_m \ p(\mathbf{z}_k|\mathbf{x}_m, \mu^{(t)}, \sigma^{(t)})(1 + \gamma \ \ln p(\mathbf{z}_k|\mathbf{x}_m, \mu^{(t)}, \sigma^{(t)}))}{\sum_{m=1}^{M} p(\mathbf{z}_k|\mathbf{x}_m, \mu^{(t)}, \sigma^{(t)})(1 + \gamma \ \ln p(\mathbf{z}_k|\mathbf{x}_m, \mu^{(t)}, \sigma^{(t)}))} \quad (86)$$

$$\sigma_k^{(t+1)} = \frac{\sum_{m=1}^{M} \mathbf{d}_{mk} \ p(\mathbf{z}_k|\mathbf{x}_m, \mu^{(t)}, \sigma^{(t)})(1 + \gamma \ \ln p(\mathbf{z}_k|\mathbf{x}_m, \mu^{(t)}, \sigma^{(t)}))}{\sum_{m=1}^{M} p(\mathbf{z}_k|\mathbf{x}_m, \mu^{(t)}, \sigma^{(t)})(1 + \gamma \ \ln p(\mathbf{z}_k|\mathbf{x}_m, \mu^{(t)}, \sigma^{(t)}))} \quad (87)$$

where $\mathbf{d}_{mk} = (\mathbf{x}_m - \mu_k)(\mathbf{x}_m - \mu_k)^T$

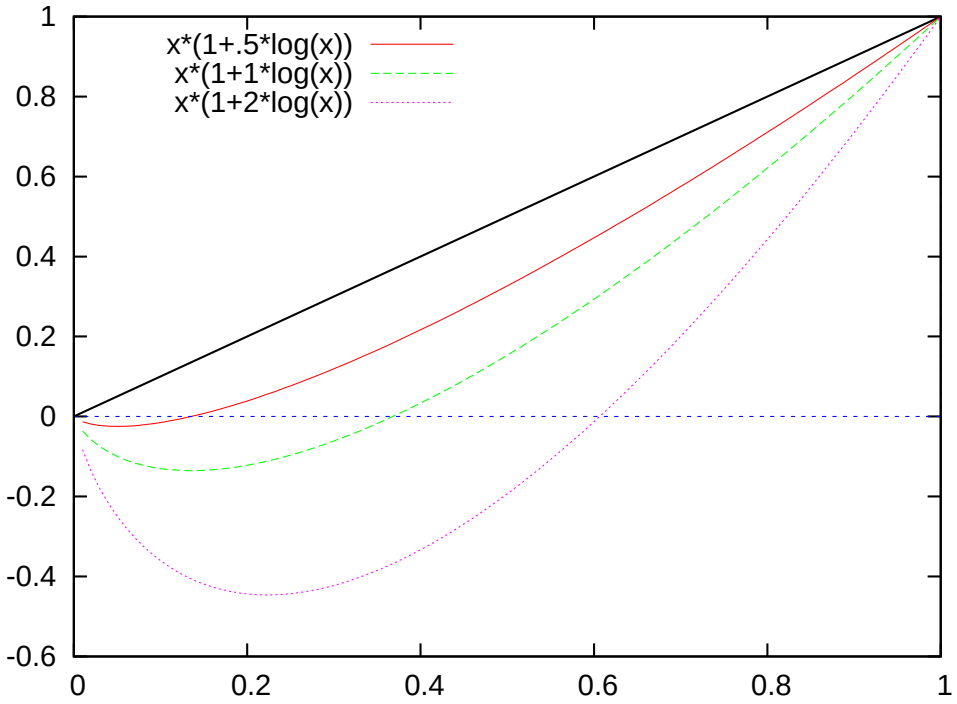
EM algorithm: Regularization

Comparing (78) and (84) just for parameter π_k:

$$\pi_k^{(t+1)} = \frac{\sum_{m=1}^{M} p(\mathbf{z}_k|\mathbf{x}_m, \pi_k^{(t)}, \mu_k, \sigma_k)}{\sum_{m=1}^{M}\sum_{k=1}^{K} p(\mathbf{z}_k|\mathbf{x}_m, \pi_k^{(t)}, \mu_k, \sigma_k)} = \frac{\sum_{m=1}^{M} p(\mathbf{z}_k|\mathbf{x}_m, \pi_k^{(t)}, \mu_k, \sigma_k)}{M}$$

$$\pi_k^{(t+1)} = \frac{\sum_{m=1}^{M} p(\mathbf{z}_k|\mathbf{x}_m, \pi_k^{(t)}, \mu_k, \sigma_k)(1+\gamma \ \ln p(\mathbf{z}_k|\mathbf{x}_m, \pi_k^{(t)}, \mu_k, \sigma_k))}{\sum_{m=1}^{M}\sum_{k=1}^{K} p(\mathbf{z}_k|\mathbf{x}_m, \pi_k^{(t)}, \mu_k, \sigma_k)(1+\gamma \ \ln p(\mathbf{z}_k|\mathbf{x}_m, \pi_k^{(t)}, \mu_k, \sigma_k))}$$

Contribution of each sample in the numerator



EM algorithm: Regularization

No regularization

$\theta^{(0)} = (\pi_1 = 0.700, \pi_2 = 0.100, \pi_3 = 0.200)$

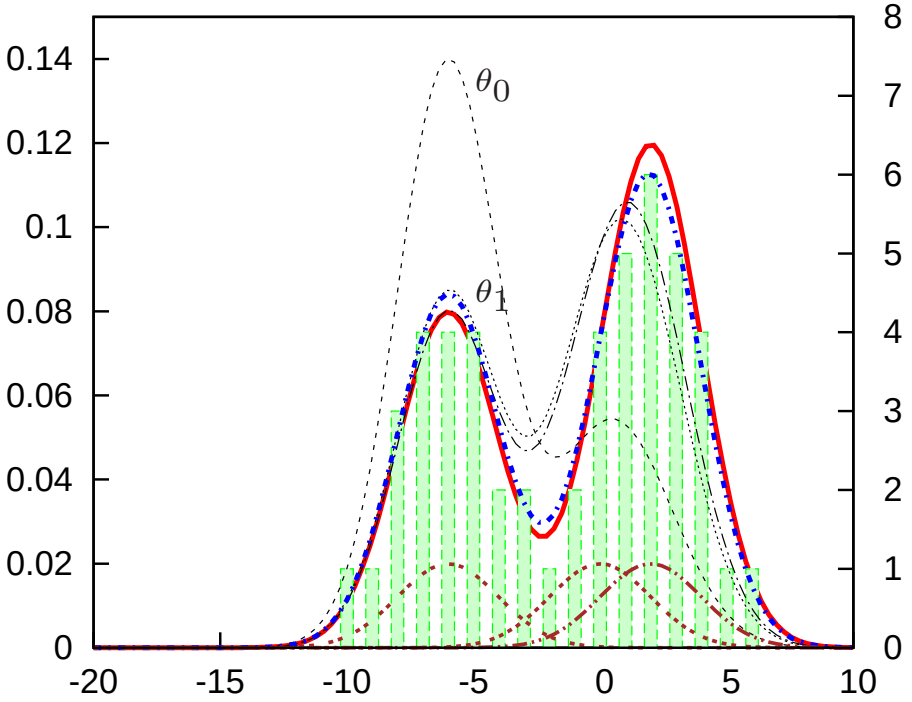
$\theta^{(1)} = (\pi_1 = 0.423, \pi_2 = 0.259, \pi_3 = 0.318)$

$\theta^{(2)} = (\pi_1 = 0.399, \pi_2 = 0.320, \pi_3 = 0.281)$

$\theta^{(3)} = (\pi_1 = 0.402, \pi_2 = 0.391, \pi_3 = 0.207)$

... (26 iterations later)

$\theta^{(26)} = (\pi_1 = 0.421, \pi_2 = 0.543, \pi_3 = 0.036)$



Regularization $\gamma = .1$

$\theta^{(0)} = (\pi_1 = 0.700, \pi_2 = 0.100, \pi_3 = 0.200)$

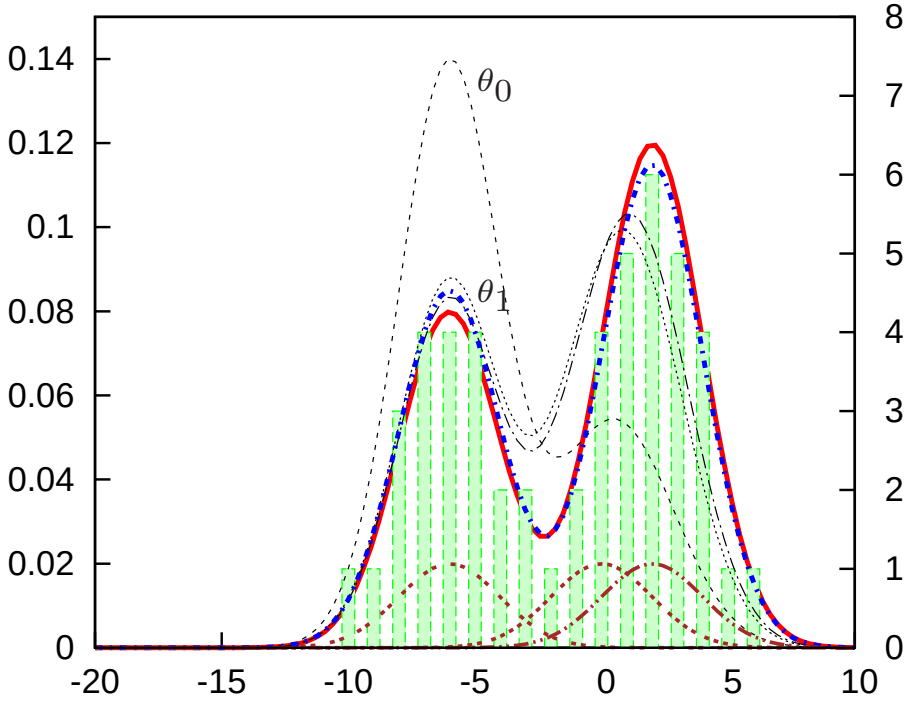
$\theta^{(1)} = (\pi_1 = 0.438, \pi_2 = 0.251, \pi_3 = 0.311)$

$\theta^{(2)} = (\pi_1 = 0.415, \pi_2 = 0.315, \pi_3 = 0.270)$

$\theta^{(3)} = (\pi_1 = 0.418, \pi_2 = 0.385, \pi_3 = 0.197)$

... (13 iterations later)

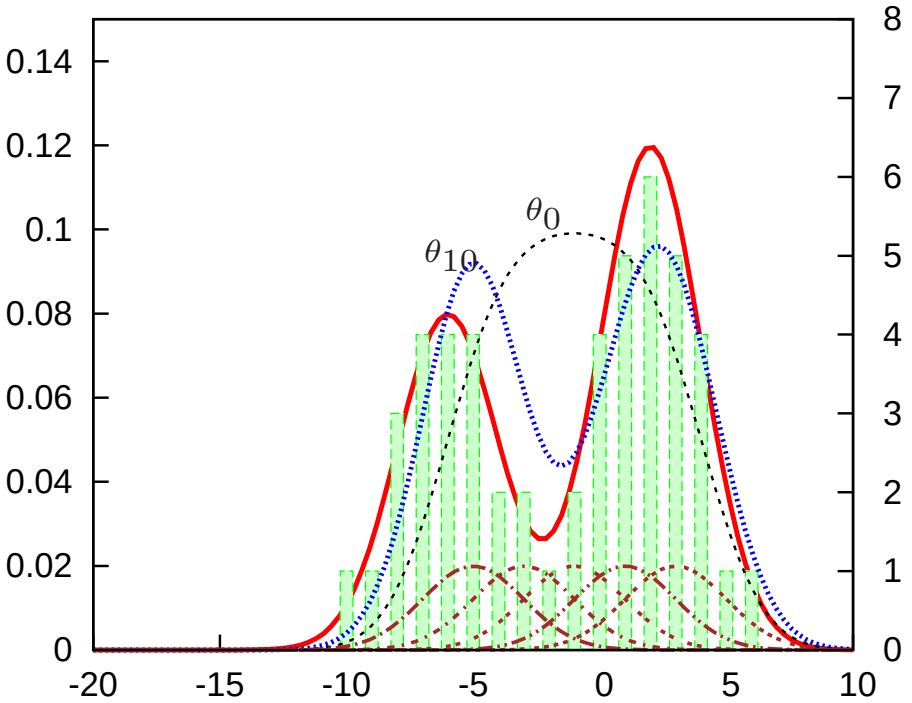
$\theta^{(13)} = (\pi_1 = 0.426, \pi_2 = 0.574, \pi_3 = 0.000)$



EM algorithm: Regularization

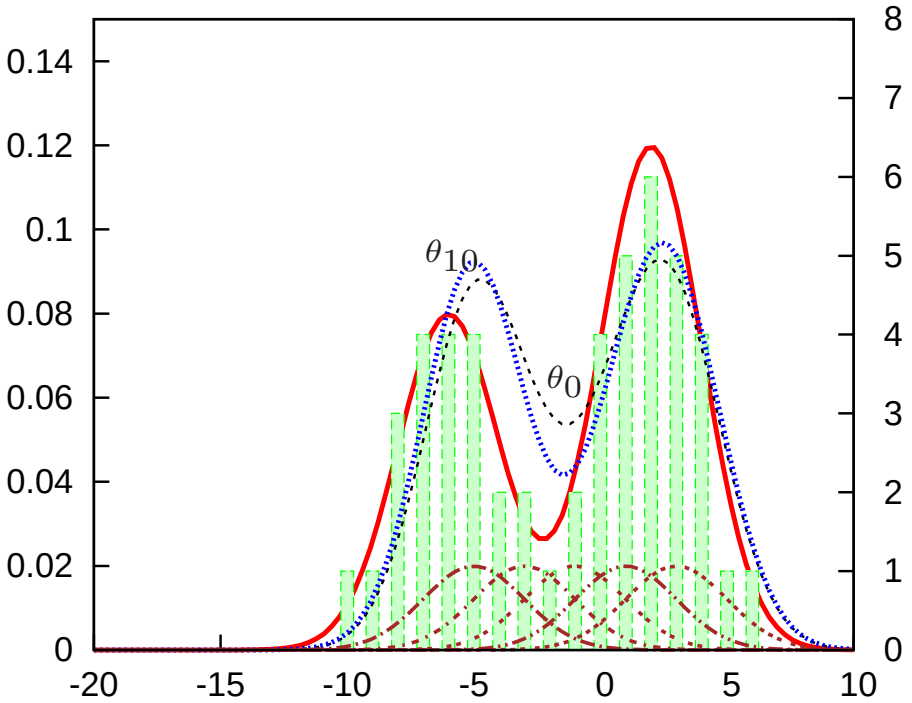
Regularization $\gamma = .1$

$\theta^{(0)} = (\pi_1{=}0.200, \pi_2{=}0.200, \pi_3{=}0.200, \pi_4{=}0.200, \pi_5{=}0.200;$
$\mu_1 = -5.0, \mu_2 = -3.0, \mu_3 = -1.0, \mu_4 = 1.0, \mu_5 = 3.0)$
$\theta^{(10)} = (\pi_1{=}0.459, \pi_2{=}0.000, \pi_3{=}0.000, \pi_4{=}0.204, \pi_5{=}0.336)$



Regularization $\gamma = .1$

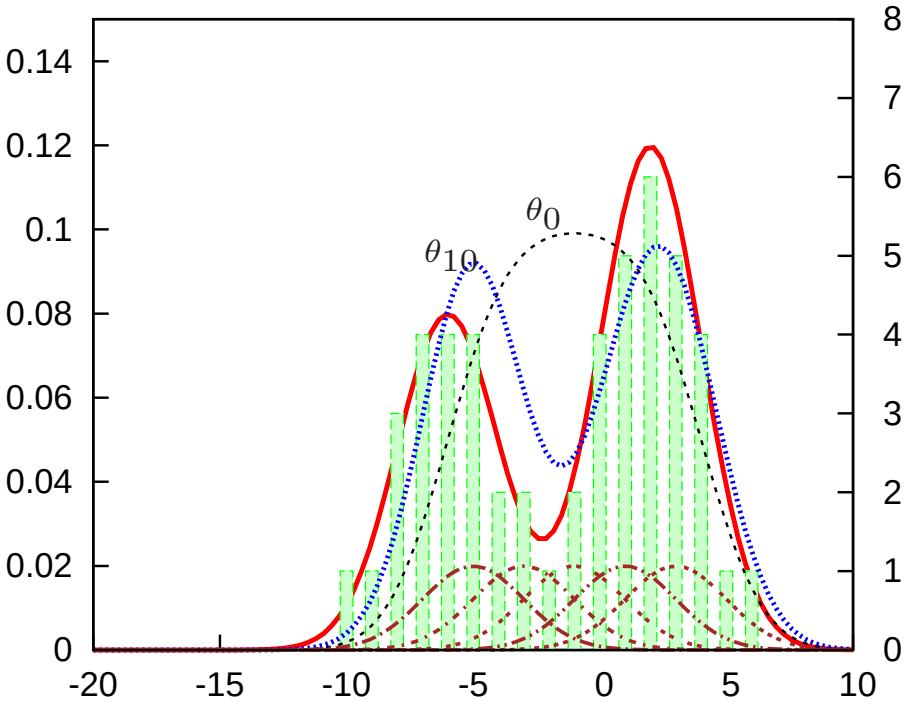
$\theta^{(0)} = (\pi_1{=}0.400, \pi_2{=}0.050, \pi_3{=}0.050, \pi_4{=}0.150, \pi_5{=}0.350;$
$\mu_1 = -5.0, \mu_2 = -3.0, \mu_3 = -1.0, \mu_4 = 1.0, \mu_5 = 3.0)$
$\theta^{(10)} = (\pi_1{=}0.460, \pi_2{=}0.000, \pi_3{=}0.000, \pi_4{=}0.194, \pi_5{=}0.346)$



EM algorithm: Regularization

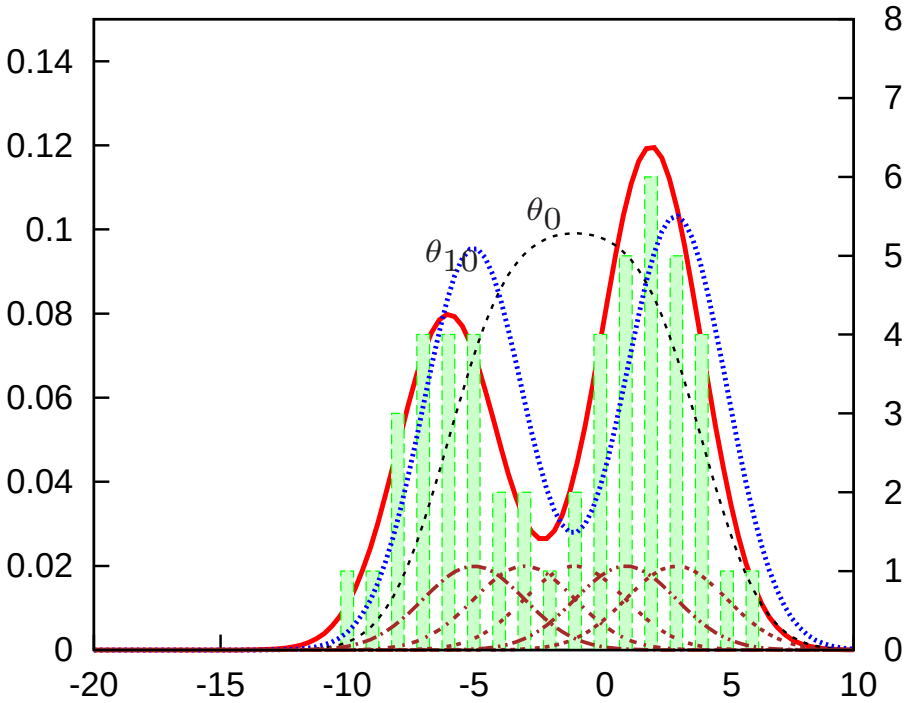
Regularization $\gamma = .1$

$\theta^{(0)} = (\pi_1{=}0.200, \pi_2{=}0.200, \pi_3{=}0.200, \pi_4{=}0.200, \pi_5{=}0.200;$
$\mu_1 = -5.0, \mu_2 = -3.0, \mu_3 = -1.0, \mu_4 = 1.0, \mu_5 = 3.0)$
$\theta^{(10)} = (\pi_1{=}0.459, \pi_2{=}0.000, \pi_3{=}0.000, \pi_4{=}0.204, \pi_5{=}0.336)$



Regularization $\gamma = 0.3$

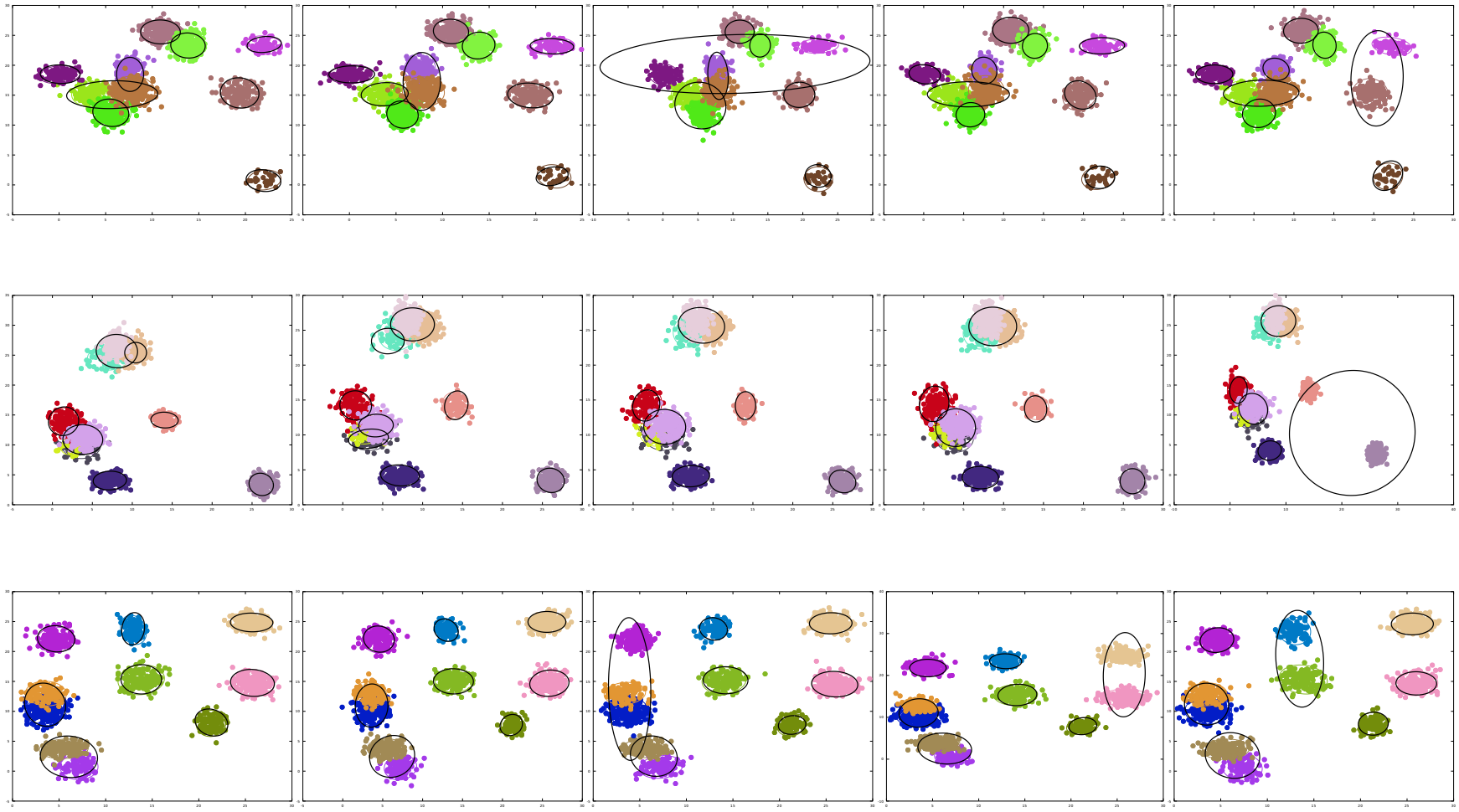
$\theta^{(0)} = (\pi_1{=}0.200, \pi_2{=}0.200, \pi_3{=}0.200, \pi_4{=}0.200, \pi_5{=}0.200;$
$\mu_1 = -5.0, \mu_2 = -3.0, \mu_3 = -1.0, \mu_4 = 1.0, \mu_5 = 3.0)$
$\theta^{(10)} = (\pi_1{=}0.479, \pi_2{=}0.000, \pi_3{=}0.000, \pi_4{=}0.009, \pi_5{=}0.512)$

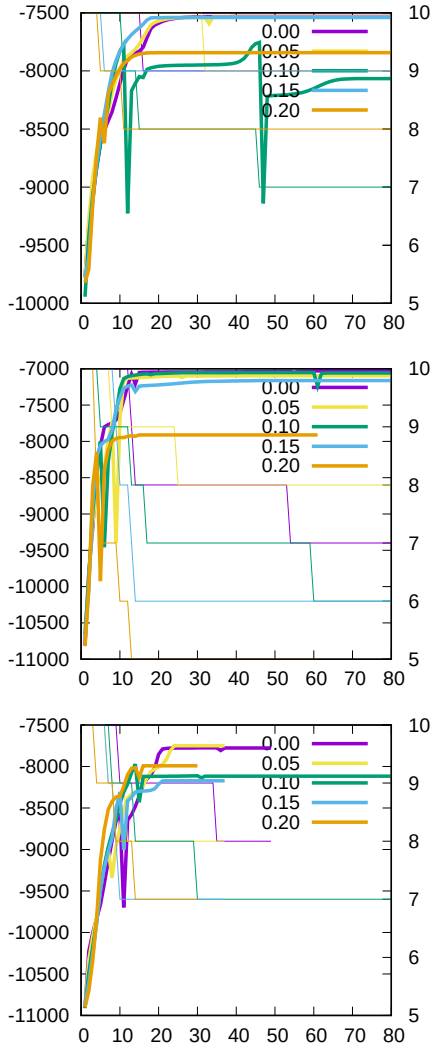


EM algorithm: Regularization

2D. EM with regularization. 10 initial clusters.

$\gamma = 0.00$ $\gamma = 0.05$ $\gamma = 0.10$ $\gamma = 0.15$ $\gamma = 0.20$

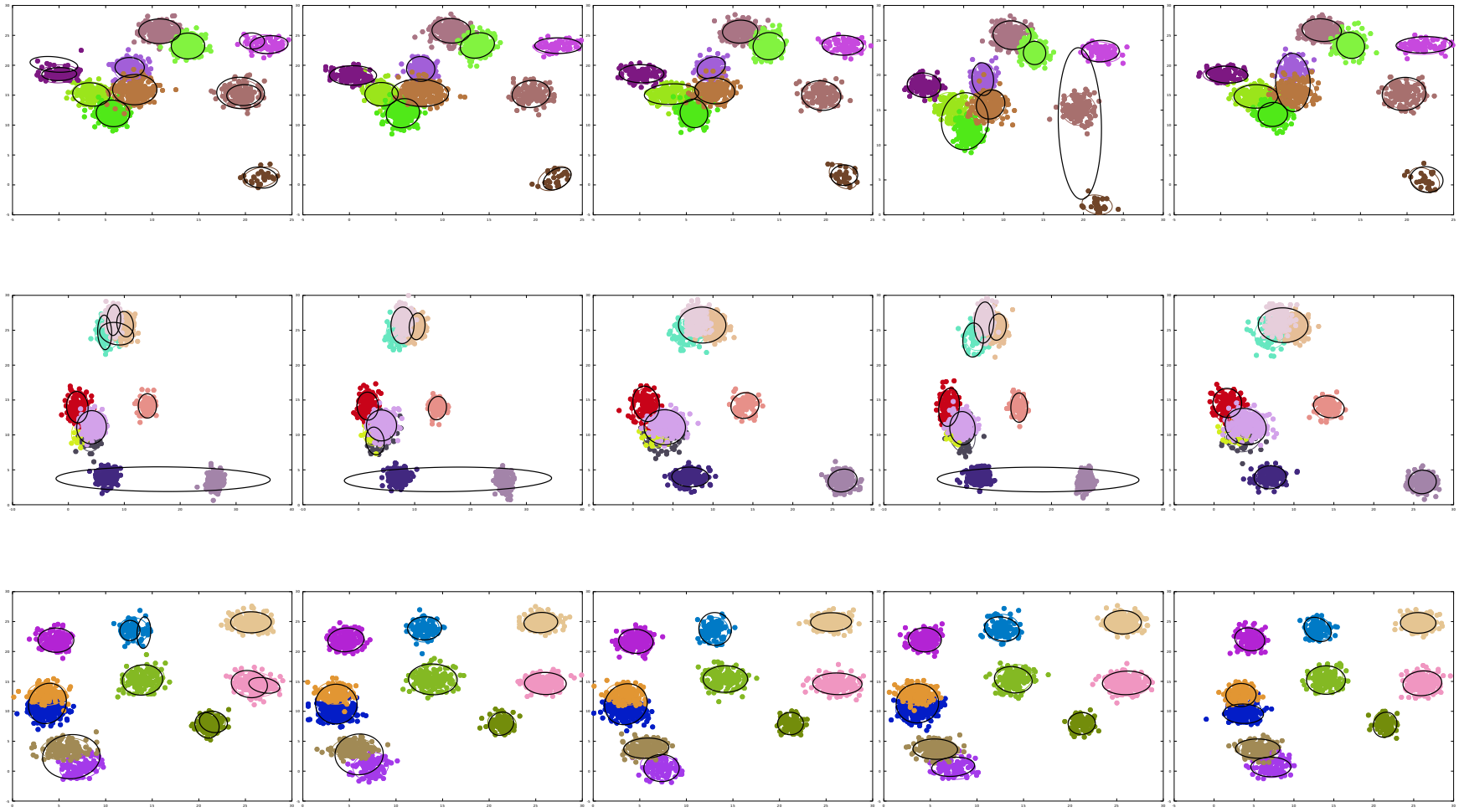


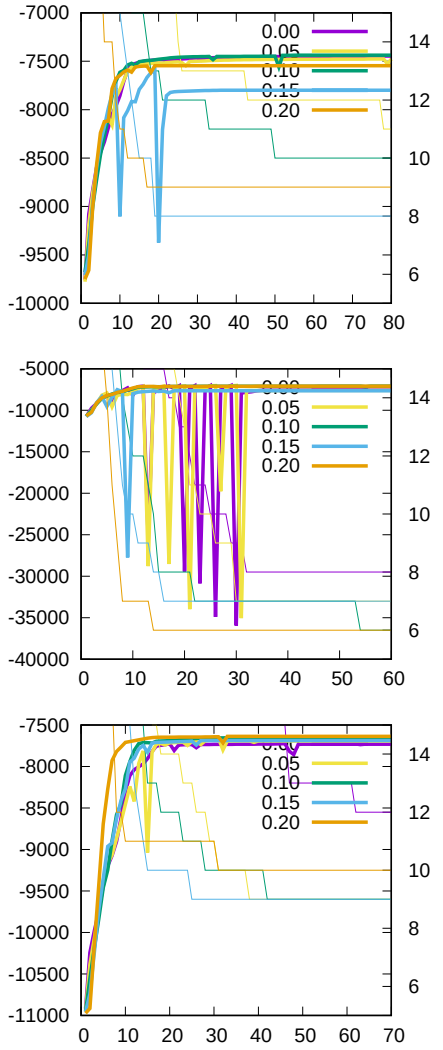


EM algorithm: Regularization

2D. EM with regularization. 20 initial clusters.

$\gamma = 0.00$ $\gamma = 0.05$ $\gamma = 0.10$ $\gamma = 0.15$ $\gamma = 0.20$





EM algorithm: Regularization

Pending issues:

- Decide the value of γ: a validation set, ...
- Criterion to remove a gaussian
- Initialize the algorithm: c-means, ...
- Dimensionality

Additional references: [Yi, 15; Houdoui, 23]

Exercises

Exercise (***): Reproduce an example similar to the unidimensional example with three unidimensional distributions, with known mean and equal and known variance where only π_1, π_2, π_3 are unknown. Obtain the corresponding plots.

Exercise (***): Reproduce an example similar to the unidimensional example with two bidimensional distributions, with known mean vector and equal and known co-variance matrx where only π_1, π_2, π_3 are unknown. Obtain the corresponding plots.

Exercise (****): Reproduce an example similar to the unidimensional example with two unidimensional distributions, with equal and known variance where the mean of each distribtuion and π_1, π_2 are unknown. Obtain the corresponding plots.

Exercise (****): Reproduce an example similar to the unidimensional example with two unidimensional distributions, with equal and known mean where the variances and π_1, π_2 are unknown. Obtain the corresponding plots.

Exercise (****): Reproduce an example similar to the bi-dimensional example, but the inicialization has to be performed with the c-means algorithm. Obtain the corresponding plots.

Exercise (*****): Reproduce an example similar to the bi-dimensional example, but the inicialization has to be performed with the c-means algorithm. Apply the algorithm to a real classification task.

Exercise (*****): Apply the described regularization method to other type of models (e.g. HMM)

EM algorithm: Regularization

Regularization applied to grammatical models (PFA)
From (80), let W a variable associated to the strings of a PFA $\mathcal{A}$ and Θ a variable associated to a state sequence:

$$\tilde{L}_{\mathcal{A}}(W) = L_{\mathcal{A}}(W) + \gamma\, I_{\mathcal{A}}(W,\Theta) = L_{\mathcal{A}}(X) + \gamma\, (H_{\mathcal{A}}(\Theta) - H_{\mathcal{A}}(\Theta \mid W))$$

Using (27), then:

$$\tilde{L}_{\mathcal{A}}(W) = L_{\mathcal{A}}(W) + \gamma\, H_{\mathcal{A}}(W) \tag{88}$$

Another point of view: Since W is a deterministic function of Θ (the contrary is not true) then the mutual information coincides with $H_{\mathcal{A}}(W)$ (see plot in page 11)

Estimation expression: (Hypothesis!)

$$\bar{P}(i,v,j) = \frac{\sum_w \frac{1}{p_{\mathcal{A}}(w)} \sum_{\theta_w} \mathrm{N}((i,v,j),\theta_w) p_{\mathcal{A}}(w,\theta_w)(1+\gamma \log p_{\mathcal{A}}(\theta_w|w))}{\sum_w \frac{1}{p_{\mathcal{A}}(w)} \sum_{\theta_w} \mathrm{N}(i,\theta_w) p_{\mathcal{A}}(w,\theta_w z)(1+\gamma \log p_{\mathcal{A}}(\theta_w|w))} \tag{89}$$

Exercise (****): Proof if the previous expression is true or false!

Index

Discriminative training criterion

H-criterion based training [Gopalakrishnan 88]. Let a, b, c constants with $a > 0$:

$$H_{a,b,c}(\theta) = -\frac{1}{n}\sum_{i=1}^{n} \log p_\theta^a(\mathbf{x}_i, \mathbf{z}_i)\; p_\theta^b(\mathbf{x}_i)\; p_\theta^c(\mathbf{z}_i) \tag{90}$$

Observe the close relation between the previous expression and the Mutual Information (9)

Maximal Mutual Information

$$\begin{aligned} F(\theta) &= \frac{1}{n}\sum_{i=1}^{n} \log \frac{p_\theta(\mathbf{x}_i|\mathbf{z}_i)}{p_\theta(\mathbf{x}_i)} && (91)\\ &= \frac{1}{n}\sum_{i=1}^{n} \log \frac{p_\theta(\mathbf{x}_i, \mathbf{z}_i)}{p_\theta(\mathbf{x}_i)\; p_\theta(\mathbf{z}_i)} && (92)\\ &\approx H_\theta(X) + H_\theta(Z) - H_\theta(X, Z) && (93) \end{aligned}$$

Thus $F(\theta) = H_{1,-1,-1}(\theta)$ in expression (90)

The join distribution $p(\mathbf{x}, \mathbf{z})$ in expression (9) is "replaced" in (92) by an empirical distribution $\tilde{p}(\mathbf{x}, \mathbf{z})$

Discriminative training criterion

MMI Criterion. The aim is to estimate the parameters in such a way as to (approximately) reduce the error rate on the training data.

$$\hat{\theta} = \arg\max_{\theta} \sum_{i=1}^{n} \log \frac{p_\theta(\mathbf{x}_i|\mathbf{z}_i)}{p_\theta(\mathbf{x}_i)} = \arg\max_{\theta} \sum_{i=1}^{n} \log \frac{p_\theta(\mathbf{x}_i|\mathbf{z}_i)}{\sum_j p_\theta(\mathbf{x}_i|\mathbf{z}_j)\ p_\theta(\mathbf{z}_j)} \tag{94}$$

MMI Criterion - Alternative: Use $F(\theta) = H_{1,-h,0}(\theta)$ in expression (90) with $0 \leq h \leq 1$. Value h aims to establish the degree that competing derivation discriminate against the reference derivation. The optimization of the H-criterion aims to maximize the numerator while decreasing the denominator.[1]

$$\hat{\theta} = \arg\max_{\theta} \sum_{i=1}^{n} \log \frac{\sum_{j'} \left(p_\theta(\mathbf{x}_i, \mathbf{z}_{j'})\right)^\nu}{\left(\sum_j \left(p_\theta(\mathbf{x}_i, \mathbf{z}_j)\right)^\nu\right)^h} \tag{95}$$

with $\nu > 0$

[1] **Important:** The sum in the denominator has to include the terms in the numerator.

Discriminative training criterion

Problem: expression (95) has a rational function

Solution: Growth transformation for rational functions [Gopalakrishnan 1991]

Example: SCFG with parameters ϑ [Maca 2021]

$$\bar{P}(A \to \alpha) = \frac{\sum_x \frac{1}{p^{\nu}_{\vartheta}(x,\Delta')} \sum_{t'_x \in \Delta'} \mathrm{N}(A \to \alpha, t'_x)\; p^{\nu}_{\vartheta}(t'_x) - h \sum_x \frac{1}{p^{\nu}_{\vartheta}(x,\Delta)} \sum_{t_x \in \Delta} \mathrm{N}(A \to \alpha, t_x)\; p^{\nu}_{\vartheta}(t_x) + p(A \to \alpha) C^*(\vartheta)}{\sum_x \frac{1}{p^{\nu}_{\vartheta}(x,\Delta')} \sum_{t'_x \in \Delta'} \mathrm{N}(A, t'_x)\; p^{\nu}_{\vartheta}(t'_x) - h \sum_x \frac{1}{p^{\nu}_{\vartheta}(x,\Delta)} \sum_{t_x \in \Delta} \mathrm{N}(A \to \alpha, t_x)\; p^{\nu}_{\vartheta}(t_x) + C^*(\vartheta)} \tag{96}$$

where $\Delta' \subseteq \Delta$ are subset of derivation of string x and

$$C^*(\vartheta) = \max\left(\max_{p(A\to\alpha)}\left(-\frac{\sum_x \frac{1}{p^{\nu}_{\vartheta}(x,\Delta')} \sum_{t'_x \in \Delta'} \mathrm{N}(A \to \alpha, t'_x)\; p^{\nu}_{\vartheta}(t'_x) - h \sum_x \frac{1}{p^{\nu}_{\vartheta}(x,\Delta)} \sum_{t_x \in \Delta} \mathrm{N}(A \to \alpha, t_x)\; p^{\nu}_{\vartheta}(t_x)}{p(A \to \alpha)}\right), 0\right) \tag{97}$$

Discriminative training criterion

MMI Criterion for CHMM [Povey 2001]. For M observation sequences $\{O_1, O_2, \ldots, O_M\}$ choose λ that maximizes:

$$\mathcal{F}_{\text{mmi}}(\lambda) = \frac{1}{M} \sum_{i=1}^{M} \log p_\lambda(w_i^{\text{ref}}|O_i) = \frac{1}{M} \sum_{i=1}^{M} \log \frac{p_\lambda(O_i|w_i^{\text{ref}})\ P(w_i^{\text{ref}})}{\sum_{\hat{w}} p_\lambda(O_i|\hat{w})\ P(\hat{w})} \tag{98}$$

where w is a composite model corresponding to the word sequence represented in O. Note that summation for $\hat{w}$ is not feasible. In practice this is restricted to the set of confusable hypotheses (see also [Schlüter 2001])

Discriminative training criterion

MPE Criterion for CHMM [Povey 2002]. To minimize the function:

$$\mathcal{F}_{\text{mpe}}(\lambda) = \sum_{i=1}^{M} \sum_{w} p_\lambda(w|O_i)\mathcal{L}(w, w_i^{\text{ref}}) \tag{99}$$

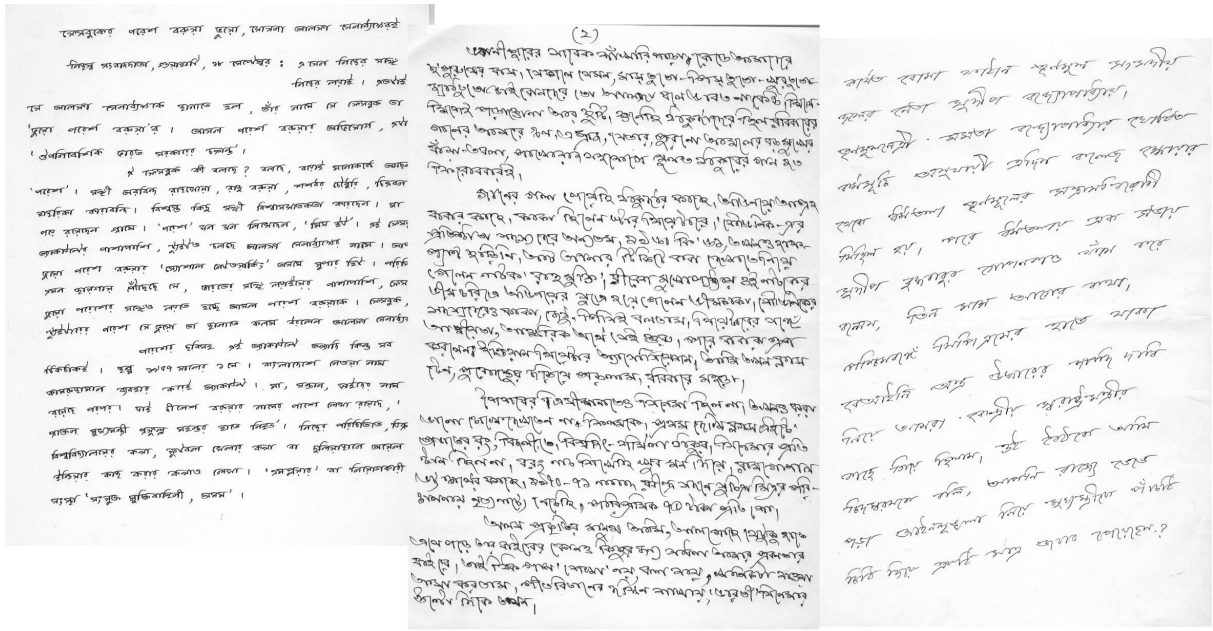
where $\mathcal{L}(w, w_i^{\text{ref}})$ is the "loss" between the hypothesis w and the reference w_i^{ref}. In MPE the "loss" is based on the Levenshtein edit distance between the phone sequences of the reference and the hypothesis (see HTK manual for details)

Discriminative training criterion

Experiments [Sánchez 2016]: Handwritten Text Recognition of Bengali using holistic segmentation-free Handwritten Text Recognition (HTR) technology

Characteristics of Bengali:

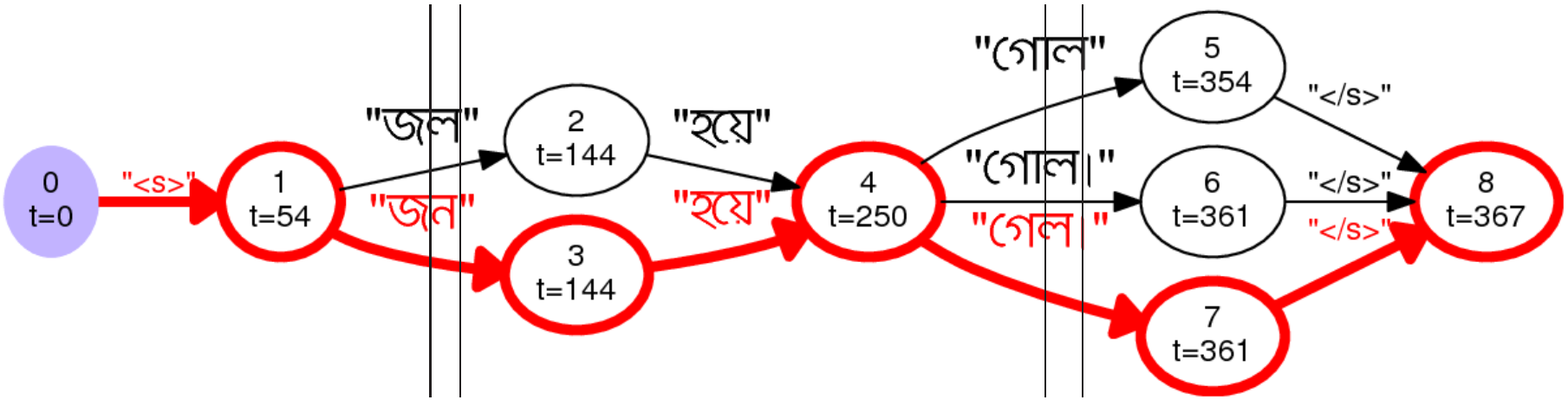
- About 300 different characters
- About 90 characters appers 3 or less times in the dataset



Discriminative training criterion

Training Criterion: First ML, and then MPE

- The numerator is the real transcript and it is exactly the log-likelihood
- The denominator includes the competitor transcripts (restricted to be the most probable transcripts according to a less restrictive LM: word-based LM or character-based LM)



Discriminative training criterion

Number of:	P0	Total
Pages	20	98
Lines	465	1,785
Run. words	4,407	15,694
Run. OOV	1,503	-
Lexicon	1,953	4,962
Char. set size	217	291
Run. Characters	5,148	42,712

- Cross-validation: 5 partitions
- About 30% OOV in average

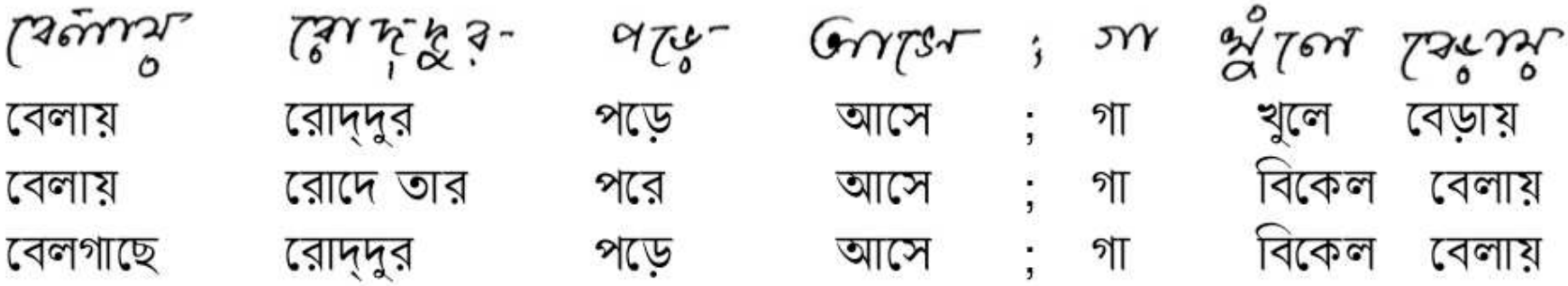
- Feature extraction based on moments computed from a sliding window
- No special treatment for modifiers placed out of the main body:  ,  , ...
  Left to the LM model
- Two-stroked characters () modeled with two HMM. The lexical model take care of reversing the problem

Discriminative training criterion

Results. Recognition with word-based LM (2-gram)

	All Parts.		P0	
	WER	CER	WER	CER
Baseline	63.4	39.3	67.9	42.8
+ Gen. HMM	62.7	38.5	66.7	40.7
+ CV	49.7	31.8	52.2	32.8
+ WL-MPE	-		50.4	29.8
+ CL-MPE-1	-		43.8	25.8
+ CL-MPE-2	-		43.0	25.4

Problem: 88 characters out of approximately 290 appeared 3 times or less in the dataset
Solution: to substitute by a generic HMM



Discriminative training criterion: Practice

This practical exercise is devoted to develop a PCFG for modeling geometric figures, namely, triangles. The toolkit includes a program for estimating a SCFG with the Inside-Outside algorithm, with the Viterbi-Score algorithm, with and without bracketed samples, **with and without discriminative learning**. For example, in the following commands:

```
bin/scfg_cgr -g M/Gm1 -f M/Gm1-new
bin/scfg_learn_mmi -g M/Gm1-new -f M/Gm1-new-20 -i 20 -p D/D
bin/scfg_learn_mmi -g M/Gm1-new -f M/Gm1-new-br-20 -i 20 -p D/Dpar
```

The first command creates a new SCFG, the second trains the grammar with the Inside-Outside algorithm with plain and positive samples, and the third command trains the grammar with the Inside-Outside algorithm with bracketed positive samples. The second command could take a lot of time given the time complexity and should be used taking into account this restriction. Therefore if the sample has many repeated strings then it can trained also as follows:

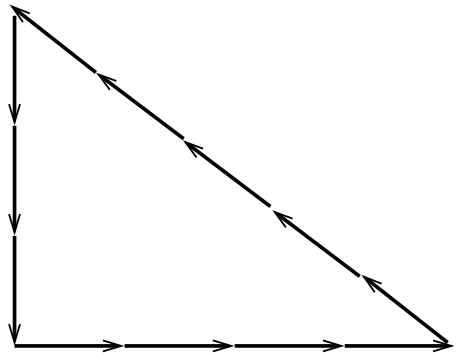
```
sort D/Dpar | uniq -c | sed 's/^ \+//g;s/ \+/ /g;' > D/Dpar-s-u
bin/scfg_learn_mmi -g M/Gm1-new -f M/Gm1-new-br-20 -i 20 -p D/Dpar-s-u -H 0.1 -U
```

With the `-U` option, each uniq string is parsed only once and absolute frequencies are taken into account. `-H` **option involves MMI training**.

Discriminative training criterion: Practice

PCFG can be used to represent geometric figures. Consider the following primitives:

$$a = \nearrow,\ \ b = \rightarrow,\ \ c = \searrow,\ \ d = \downarrow, e = \swarrow,\ \ f = \leftarrow,\ \ g = \nwarrow,\ \ h = \uparrow .$$



Then, a rectangle triangle like the one that can be seen to the right can be represented with the string $bbbbggggggddd$ (left bottom corner is assumed to be the begining of the string).

A dataset `SampleTriangle-10K` representing rectangle triangles is provided. The samples are bracketed as $(bbbb)(ggggg)(ddd)$. A PCFG can be trained and tested as follows:

```
bin/scfg_cgr -g M/G-triangle -f M/Gt-0
sort D/Tr-right | uniq -c | sed 's/^ \+//g;s/ \+/ /g;' > D/Tr-right-s-u
bin/scfg_learn -g M/Gt-0 -f M/Gt-100 -i 100 -p D/Tr-right-s-u -U
bin/scfg_gstr -g M/Gt-100 -c 1000 > tri-test
awk -f bin/checkTriangle tri-test | grep Y | wc -l
```

The last command outputs the number of rectangle triangles in 1000 generated strings. The number of non-terminal symbols in the PCFG is a critical point since the more non-terminal symbols the more flexibility the PCFG has to learn the samples, but it takes more time and more training samples are needed.

Discriminative training criterion: Practice

```
# train
bin/scfg_learn_mmi -g M/Gt-0 -f M/right-0.10 -p D/Tr-right-s-u -n D/Tr-right-nsu -H 0.1 -i 1 -U
bin/scfg_learn_mmi -g M/Gt-0 -f M/equil-0.10 -p D/Tr-equil-s-u -n D/Tr-equil-nsu -H 0.1 -i 1 -U
bin/scfg_learn_mmi -g M/Gt-0 -f M/isosc-0.10 -p D/Tr-isosc-s-u -n D/Tr-isosc-nsu -H 0.1 -i 1 -U
# classify with the trained models and get results
bin/scfg_prob -g M/right-0.10 -m D/Ts-right > r
bin/scfg_prob -g M/equil-0.10 -m D/Ts-right > e
bin/scfg_prob -g M/isosc-0.10 -m D/Ts-right > i
paste r e i | awk '{m=$1;argm="right"; if ($2>m) {m=$2;argm="equil";}
              if ($3>m) {m=$3;argm="isosc";}printf("right %s\n",argm);}' > results
bin/scfg_prob -g M/right-0.10 -m D/Ts-equil > r
bin/scfg_prob -g M/equil-0.10 -m D/Ts-equil > e
bin/scfg_prob -g M/isosc-0.10 -m D/Ts-equil > i
paste r e i | awk '{m=$2;argm="equil"; if ($1>m) {m=$1;argm="right";}
              if ($3>m) {m=$3;argm="isosc";} printf("equil %s\n",argm);}' >> results
bin/scfg_prob -g M/right-0.10 -m D/Ts-isosc > r
bin/scfg_prob -g M/equil-0.10 -m D/Ts-isosc > e
bin/scfg_prob -g M/isosc-0.10 -m D/Ts-isosc > i
paste r e i | awk '{m=$3;argm="isosc"; if ($1>m) {m=$1;argm="right";}
              if ($2>m) {m=$2;argm="equil";} printf("isosc %s\n",argm);}' >> results
cat results | bin/confus
#       equi isos righ  Err Err%
#  equi  721  279    0  279 27.9
#  isos  392  590   18  410 41.0
#  righ    0  134  866  134 13.4
#
# Error: 823/3000 = 27.43%
```

Index

Semi-supervised learning

Problem: Classification

- Supervised training: both label y and object representation $\mathbf{x}$ are available
- Unsupervised training: only object representation $\mathbf{x}$ are available

- Generative models: $p(y, \mathbf{x})$
- Discriminative models: $p(y|\mathbf{x})$

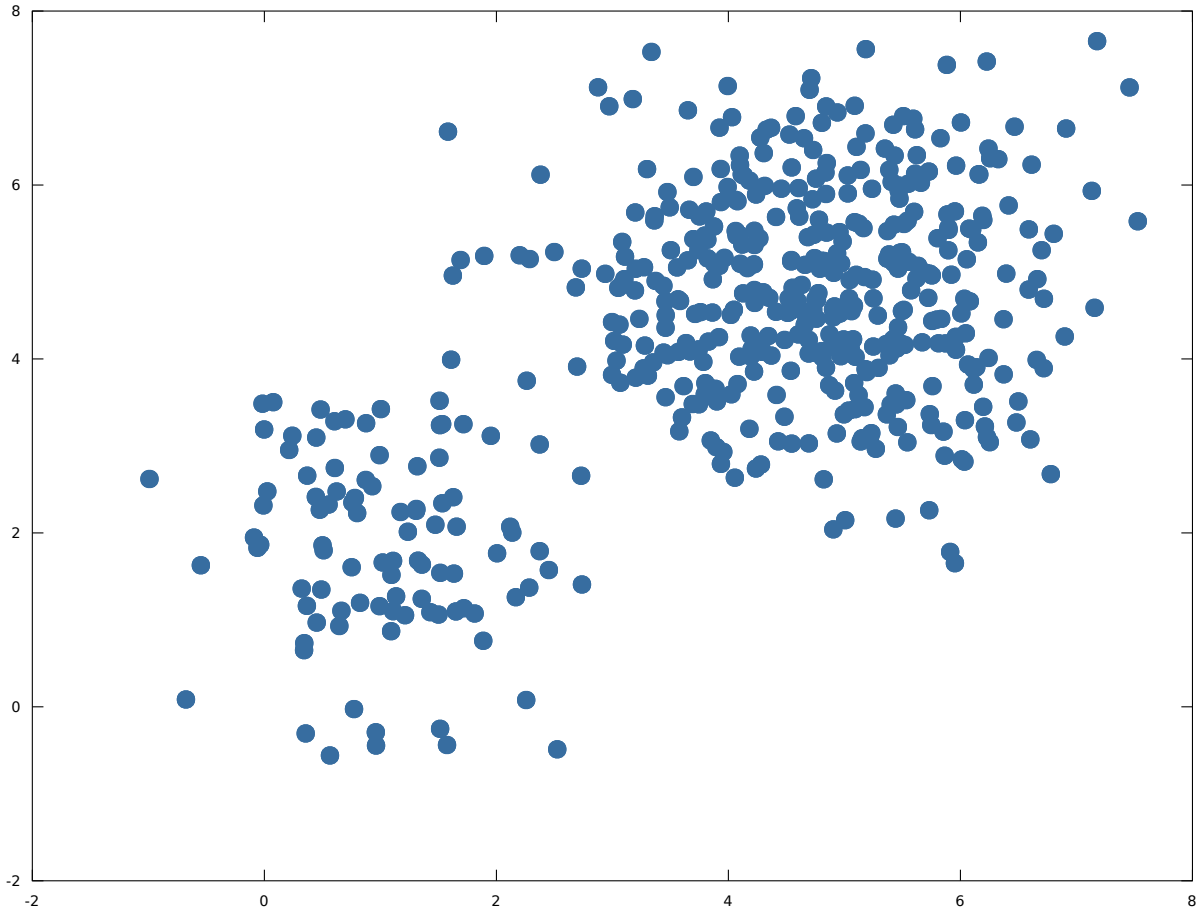
Problem: Unsupervised learning

- Regularized EM for gaussians?
- Regularized EM for gaussians with some help? Information about the class priors?

Semi-supervised learning

Intuition

Can we guess the two classes given the priors $p(c_1) = 0.19$ and $p(c_2) = 0.81$?



Semi-supervised learning

Unsupervised learning for generative models:

$$\arg\max_{\theta} \log p_\theta(x) = \arg\max_{\theta} \log \sum_y p_\theta(x|y)\ p(y)$$

Some preliminarly papers [Ravi 2011]

For discriminative models: **?**

Semi-supervised learning

Formal statement

Consider a labeled sample $(\mathbf{X}, \mathbf{Z})$ and a unlabeled sample $(\tilde{\mathbf{X}})$ (IID). Let us assume that we know the labels of $(\tilde{\mathbf{X}})$ and we are using a *generative* model.

From expression (71):

$$\theta_{n+1} = \arg\max_{\theta} \left\{ \sum_{\mathbf{z}} \mathcal{P}(\mathbf{z}|\mathbf{X}, \theta_n) \ln \mathcal{P}(\mathbf{X}, \mathbf{z}|\theta) \right\}$$

If we assume that the unknown labels $\tilde{\mathbf{Z}}$ are given for $\tilde{\mathbf{X}}$, then the "complete-data" likelihood is:

$$\mathcal{P}(\mathbf{X}, \mathbf{Z}, \tilde{\mathbf{X}}, \tilde{\mathbf{Z}}) = \mathcal{P}(\mathbf{X}, \mathbf{Z})\, \mathcal{P}(\tilde{\mathbf{X}}, \tilde{\mathbf{Z}}) \tag{100}$$

Objective function

$$\mathcal{P}(\mathbf{X}, \mathbf{Z}, \tilde{\mathbf{X}}) = \sum_{\tilde{\mathbf{z}}} \mathcal{P}(\mathbf{X}, \mathbf{Z}, \tilde{\mathbf{X}}, \tilde{\mathbf{z}}) \tag{101}$$

Semi-supervised learning

$$
\begin{aligned}
\mathcal{Q}(\theta \mid \theta_n) &= \mathbb{E}_{\tilde{\mathbf{Z}}|\mathbf{X},\mathbf{Z},\tilde{\mathbf{X}},\theta_n} \ln \mathcal{P}(\mathbf{X},\mathbf{Z},\tilde{\mathbf{X}},\tilde{\mathbf{Z}}) \\
&= \sum_{\tilde{\mathbf{z}}} \mathcal{P}(\tilde{\mathbf{z}} \mid \mathbf{X},\mathbf{Z},\tilde{\mathbf{X}},\theta_n) \left(\ln \mathcal{P}(\mathbf{X},\mathbf{Z} \mid \theta) + \ln \mathcal{P}(\tilde{\mathbf{X}},\tilde{\mathbf{Z}} \mid \theta) \right) \quad \left(\ln \mathcal{P}(\mathbf{X},\mathbf{Z} \mid \theta) = \ln \prod_{i=1}^{n} \mathcal{P}(\mathbf{x}^{(i)},\mathbf{z}^{(i)} \mid \theta) \right) \\
&= \sum_{i=1}^{l} \ln \mathcal{P}(\mathbf{x}^{(i)},\mathbf{z}^{(i)} \mid \theta) \overbrace{\sum_{\tilde{\mathbf{z}}} \mathcal{P}(\tilde{\mathbf{z}} \mid \mathbf{X},\mathbf{Z},\tilde{\mathbf{X}},\theta_n)}^{=1} \\
&\quad + \sum_{j=1}^{m} \sum_{\tilde{\mathbf{z}}} \mathcal{P}(\tilde{\mathbf{z}} \mid \mathbf{X},\mathbf{Z},\tilde{\mathbf{X}},\theta_n) \ln \mathcal{P}(\tilde{\mathbf{x}}^{(j)},\tilde{\mathbf{z}}^{(j)} \mid \theta) \\
&= \sum_{i=1}^{l} \ln \mathcal{P}(\mathbf{x}^{(i)},\mathbf{z}^{(i)} \mid \theta) + \sum_{j=1}^{m} \sum_{\tilde{\mathbf{z}}} \mathcal{P}(\tilde{\mathbf{z}} \mid \tilde{\mathbf{X}},\theta_n) \ln \mathcal{P}(\tilde{\mathbf{x}}^{(j)},\tilde{\mathbf{z}}^{(j)} \mid \theta) \\
&= \sum_{i=1}^{l} \ln \mathcal{P}(\mathbf{x}^{(i)},\mathbf{z}^{(i)} \mid \theta) + \sum_{j=1}^{m} \sum_{\tilde{\mathbf{z}}} \left(\prod_{j'=1}^{m} \mathcal{P}(\tilde{\mathbf{z}} \mid \tilde{\mathbf{x}}^{(j')},\theta_n) \right) \ln \mathcal{P}(\tilde{\mathbf{x}}^{(j)},\tilde{\mathbf{z}}^{(j)} \mid \theta) \\
&= \sum_{i=1}^{l} \ln \mathcal{P}(\mathbf{x}^{(i)},\mathbf{z}^{(i)} \mid \theta) + \sum_{j=1}^{m} \sum_{\tilde{\mathbf{z}}^{(j)}} \mathcal{P}(\tilde{\mathbf{z}}^{(j)} \mid \tilde{\mathbf{x}}^{(j)},\theta_n) \ln \mathcal{P}(\tilde{\mathbf{x}}^{(j)},\tilde{\mathbf{z}}^{(j)} \mid \theta) \quad \text{(See all maths here)} \quad (102)
\end{aligned}
$$

Semi-supervised learning

Note that:

$$\mathcal{P}(\tilde{\mathbf{z}}^{(j)} \mid \tilde{\mathbf{x}}^{(j)}, \theta_n) \tag{103}$$

are the current estimates for the probabilities of each of the labels of the unlabeled examples. These can be computed as:

$$\mathcal{P}(\tilde{\mathbf{z}}^{(j)} \mid \tilde{\mathbf{x}}^{(j)}, \theta_n) = \frac{\mathcal{P}(\tilde{\mathbf{z}}^{(j)}, \tilde{\mathbf{x}}^{(j)} \mid \theta_n)}{\sum_{\tilde{\mathbf{z}}} \mathcal{P}(\tilde{\mathbf{z}}, \tilde{\mathbf{x}} \mid \theta_n)} \tag{104}$$

Note that:

$$\mathcal{P}(\tilde{\mathbf{z}}^{(j)}, \tilde{\mathbf{x}}^{(j)} \mid \theta_n) = \mathcal{P}(\tilde{\mathbf{z}}^{(j)} \mid \theta_n)\, \mathcal{P}(\tilde{\mathbf{x}}^{(j)} \mid \tilde{\mathbf{z}}^{(j)}, \theta_n)$$

For HTR, a language model can be used and only the “best” decoding can be used:

$$\hat{\mathbf{z}} = \arg\max_{z} \mathcal{P}_{\mathbf{LM}}(\mathbf{z})\, \mathcal{P}(\tilde{\mathbf{x}}^{(j)} \mid \mathbf{z})$$

Semi-supervised learning

Main idea in semi-supervised learning

1. Obtain initial models trained in a supervised way, with a small amount of annotated data
2. Use the current models for obtaining new “labeled” data from unannotated data
3. Use expression (104) to perform supervised model training
4. Repeat steps 2 and 3 until converge

Semi-supervised learning

Main idea in semi-supervised learning

1. Obtain initial models trained in a supervised way, with a small amount of annotated data

2. Use the current models with a **very good prior** for obtaining new data from unannotated data together with confidence measures

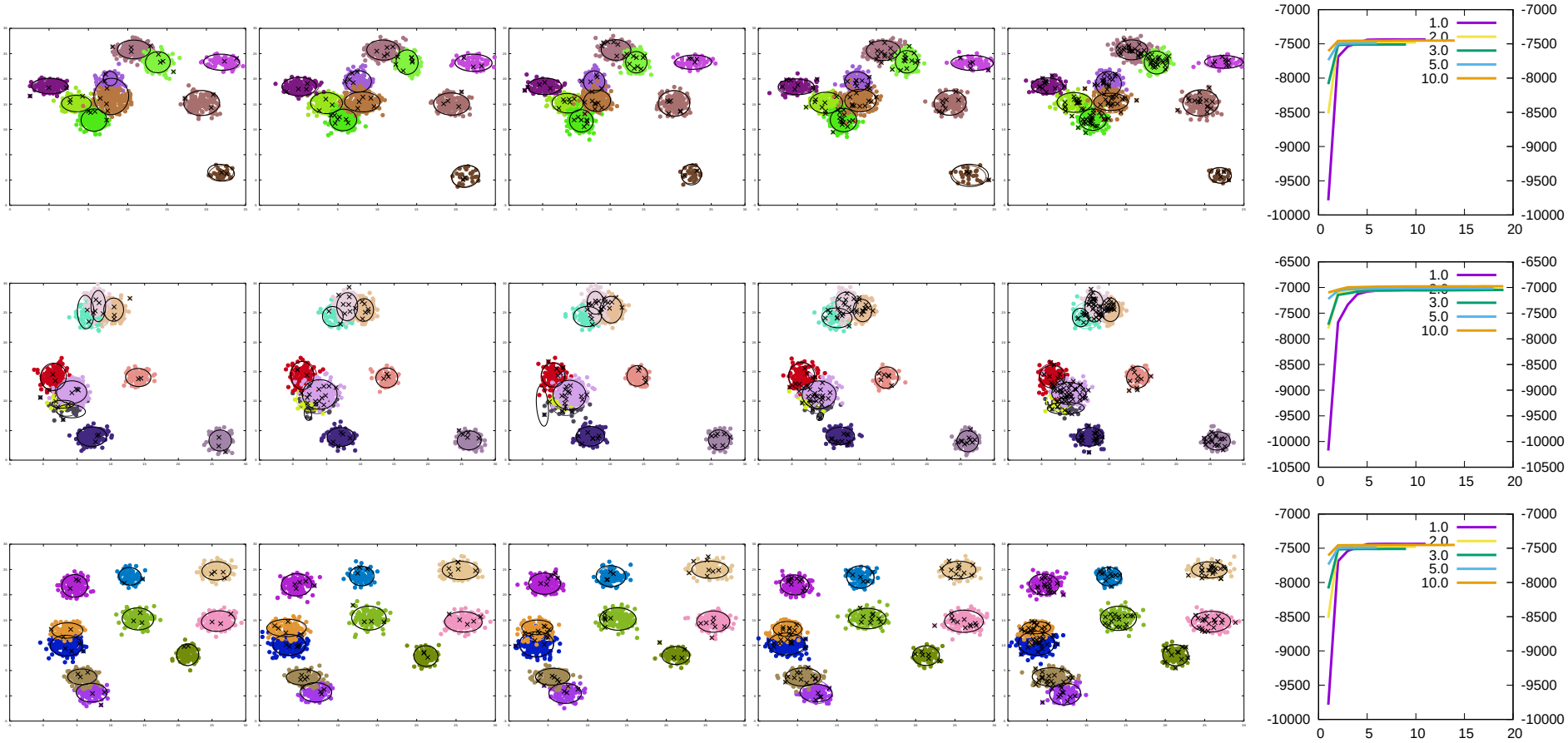
3. Use the more confident data (and the corresponding labels) from step 2 (and supervised samples from step 1) to perform supervised model training

4. Repeat steps 2 and 3 until converge

Semi-supervised learning

2D gaussians. 10 clusters. Confidence $> 50\%$

$\%= 1.0$ $\%= 2.0$ $\%= 3.0$ $\%= 5.0$ $\%= 10.0$



Semi-supervised learning

Exercise (***): Choose a data set and perform a classification experiment in a similar way to the previous slide. Plot the error rate as a function of the number of samples initially annotated.

Semi-supervised learning

Problem: Semi-supervised learning for Handwritten Text Recognition

Approach

$$\hat{\mathbf{w}} = \arg\max_{\mathbf{w}} P(\mathbf{w} \mid \mathbf{x}) = \arg\max_{\mathbf{w}} P(\mathbf{x} \mid \mathbf{w}) P(\mathbf{w})$$

$P(\mathbf{w})$: language model $\longleftarrow$ n-grams

$P(\mathbf{x} \mid \mathbf{w})$: optical model $\longleftarrow$ CRNN-HMM

Training

- $P(\mathbf{w})$: n-grams training from plain text $\longleftarrow$ **easy and cheap**
- $P(\mathbf{x} \mid \mathbf{w})$: CRNN-HMM training from pairs
  (line image, line transcript) $\longleftarrow$ **PROBLEM**

Semi-supervised learning

Training of $P(\mathbf{x} \mid \mathbf{w})$

Obtaining pairs (line image, line transcript):

- Supervised: **GT preparation is expensive**
- Unsupervised: **TO BE RESEARCHED**
- Semi-supervised: **Requires a lot of computation**

  V. Frinken, M. Baumgartner, A. Fischer, and H. Bunke, “Semi-supervised learning for cursive handwriting recognition using keyword spotting”, ICFHR, 2012.

Semi-supervised learning

Main idea

1. Obtain initial models trained in a supervised way, with a small amount of transcribed line images
2. Use the current HMM and a **very good LM** for obtaining transcripts automatically from a large amount of untranscribed lines together with confidence measures (at character, word and/or sentence level)
   (Remark: note that transcripts obtained automatically may have errors)
3. Use the more confident transcripts (and the corresponding images) from step 2 (and supervised samples from step 1) to perform model training with forced alignment
4. Repeat steps 2 and 3 until converge

Semi-supervised learning

Experiments with Benthan dataset [Sánchez, 2019]

- Training: 400 pages, 10,613 lines, 99K words
- Test: 33 pages, 860 lines, 7.8K words

Optical models: Semi-supervised learning of CRNN trained with Laia

Character-based 7-gram language model trained with:

- Scenario 1. Supervised. ICFHR 2014 training (Baseline)
- Scenario 2. Semi-supervised.
  - Scenario 2.1. Task dependent data: ICFHR 2014 training (99K words)
  - Scenario 2.2. Task independent data: Eighteenth Century Collections Online (ECCO)[2] database (2.8M words)

[2] http://www.textcreationpartnership.org/tcp-ecco/

Semi-supervised learning

Scenario 2. Semi-supervised.

- Start with 150 lines ($\approx$ 5 pages) for supervised training and 30 lines ($\approx$ 1 page) for development randomly selected $\longrightarrow$ **Labeled data**
- Iterate until convergence:
  1. Train the optical models from scratch (CRNN training) with labeled data
  2. Transcribe the whole untranscribed training dataset
  3. Substitute characters with a confidence below 99% by a garbage character
  4. Retain sentences which transcript has a confidence above 95% and with less than 10% of incorrect characters for the next iteration. Keep these sentences of all iterations
  5. Sort and select the sentences according to the confidence score from this and previous iterations $\longrightarrow$ **New labeled data maybe with errors**
  6. 10% is used for development (priorize sentences without errors)

Semi-supervised learning

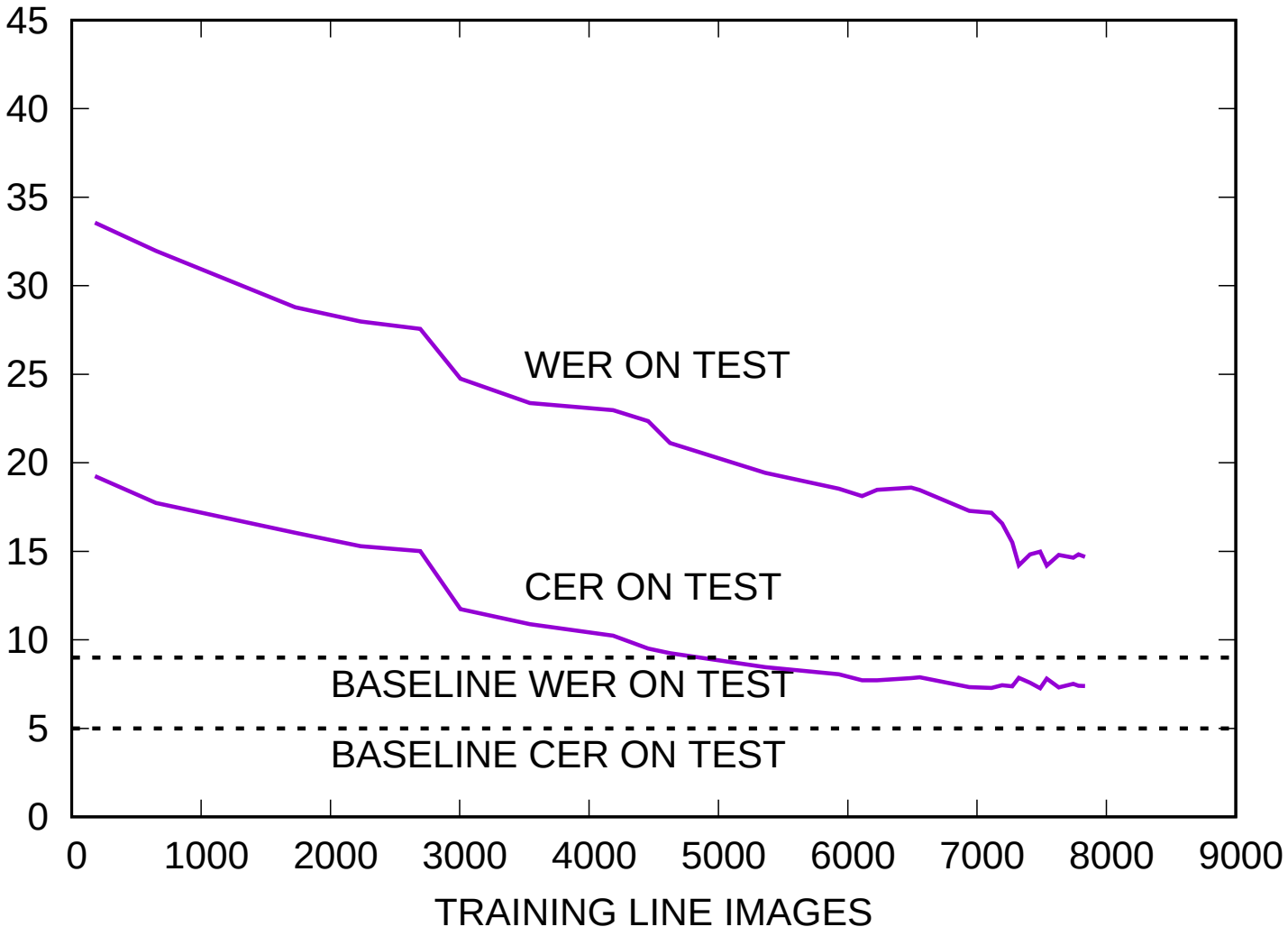
Scenario 2. Semi-supervised. Examples.

```
0.97 1    ap<@>lication of the profit of the measure towards the
0.98 0    application of the profit of the measure towards the
0.99 0    application of the profit of the measure towards the
1.00 0    application of the profit of the measure towards the
1.00 0    Application of the profit of the measure towards the
1.00 1    ap<@>lication of the profit of the measure towards the
1.00 1    ap<@>lication of the profit of the measure towards the
```

```
0.92 1    vation<@> by ether with the consideration of any
0.93 1    uation together with the couricl<@>ration of any
0.97 0    vation by other with the consideration of any
1.00 0    ration together with the consideration of any
```

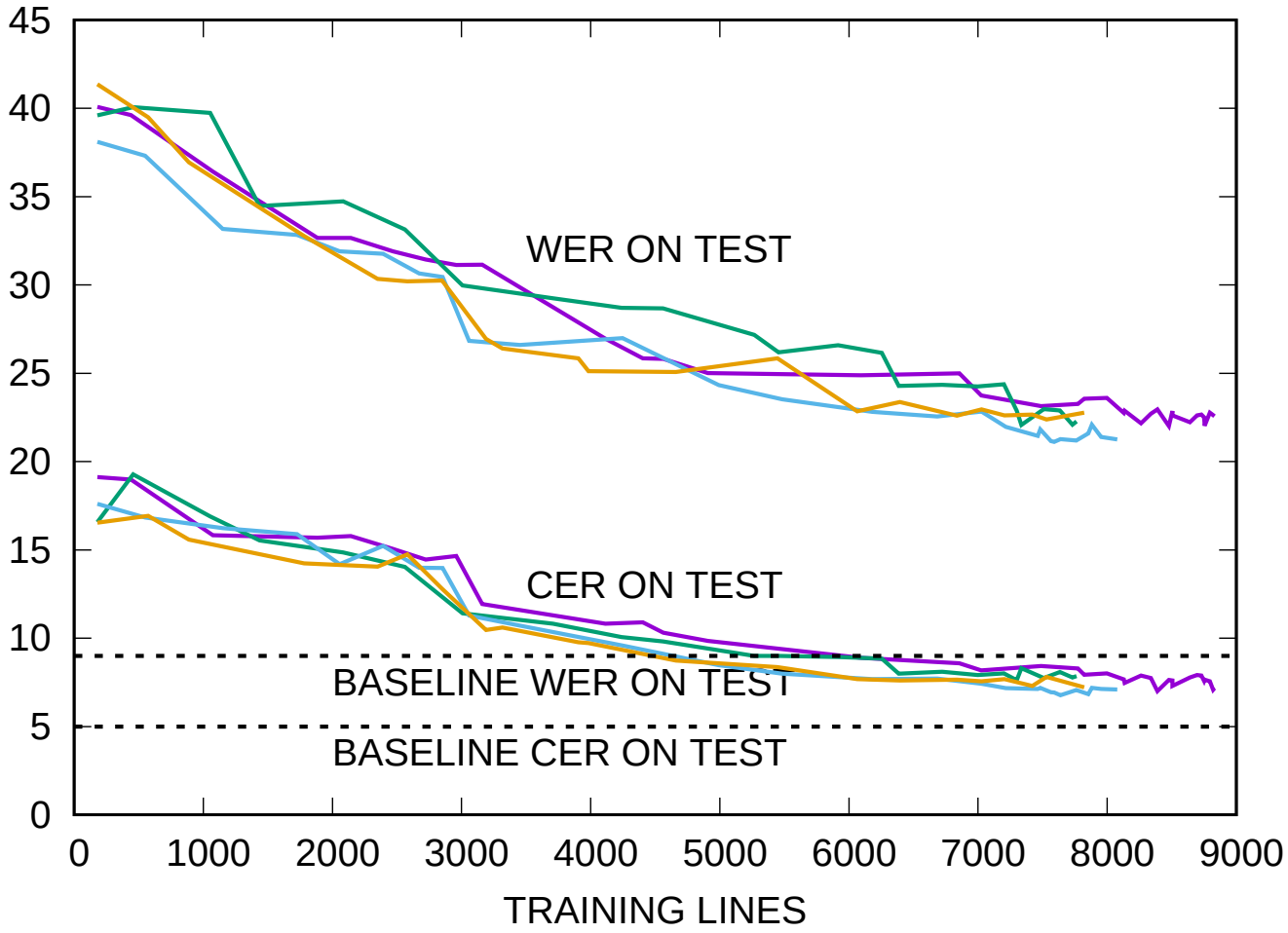
Semi-supervised learning

Scenario 2.1. Results with task dependent data for training the LM. 1 run



Semi-supervised learning

Scenario 2.2. Results with task independent data for training the LM. 4 runs



FINAL CER W/O LM: 8.3 FINAL CER WITH LM: 7.7
FINAL WER W/O LM: 25.5 FINAL WER WITH LM: 23.2

Index

Active learning

Motivation: To select the data to be annotated in a intelligent way for preparing new GT. To reduce the human effort to prepare training data.

Active learning: a *learner* uses an uncertainty function for selection of training data that is subsequently annotated by an *expert* [Settles, 2008]

Uncertainty function for an unannotated training sample $\mathbf{x}$

$$\rho(\mathbf{x}) = -\sum_{\widehat{\mathbf{y}}} P(\widehat{\mathbf{y}}|\mathbf{x}, \Theta) \log P(\widehat{\mathbf{y}}|\mathbf{x}, \Theta)$$

where $\widehat{\mathbf{y}}$ ranges over all possible labels for input $\mathbf{x}$.

Active learning

Main idea

1. Train an initial model Θ with few training annotated samples $\mathcal{L}$
2. Compute informativeness of each sample in a set of unannotated samples $\mathcal{U}$ by means of the function $\rho()$ which is computed with current model Θ
3. A set $X_B^* \subseteq \mathcal{U}$ of B unannotated samples are selected according to the result of the ρ function
4. These samples X_B^* are then annotated by an expert
5. The set $\mathcal{L}$ is updated with the new annotated samples and they are removed from $\mathcal{U}$
6. Train a new model Θ with training annotated samples $\mathcal{L}$
7. Repeat steps 2 to 6 until some convergence criterion is satisfied

Active learning

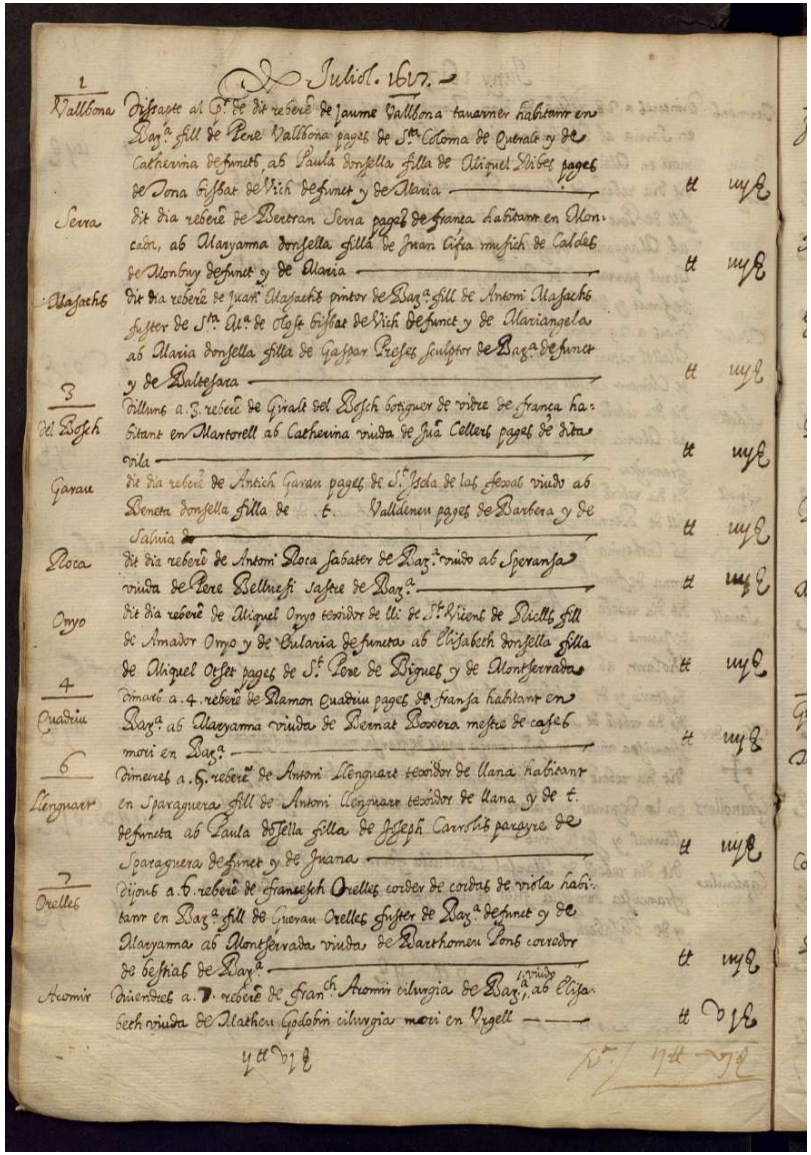
Experiments with the Esposalles dataset [Romero, 2013]

- Training: 4,620 lines, $\mathcal{L} = 150, \mathcal{U} = 4,470$
- Test: 827 lines

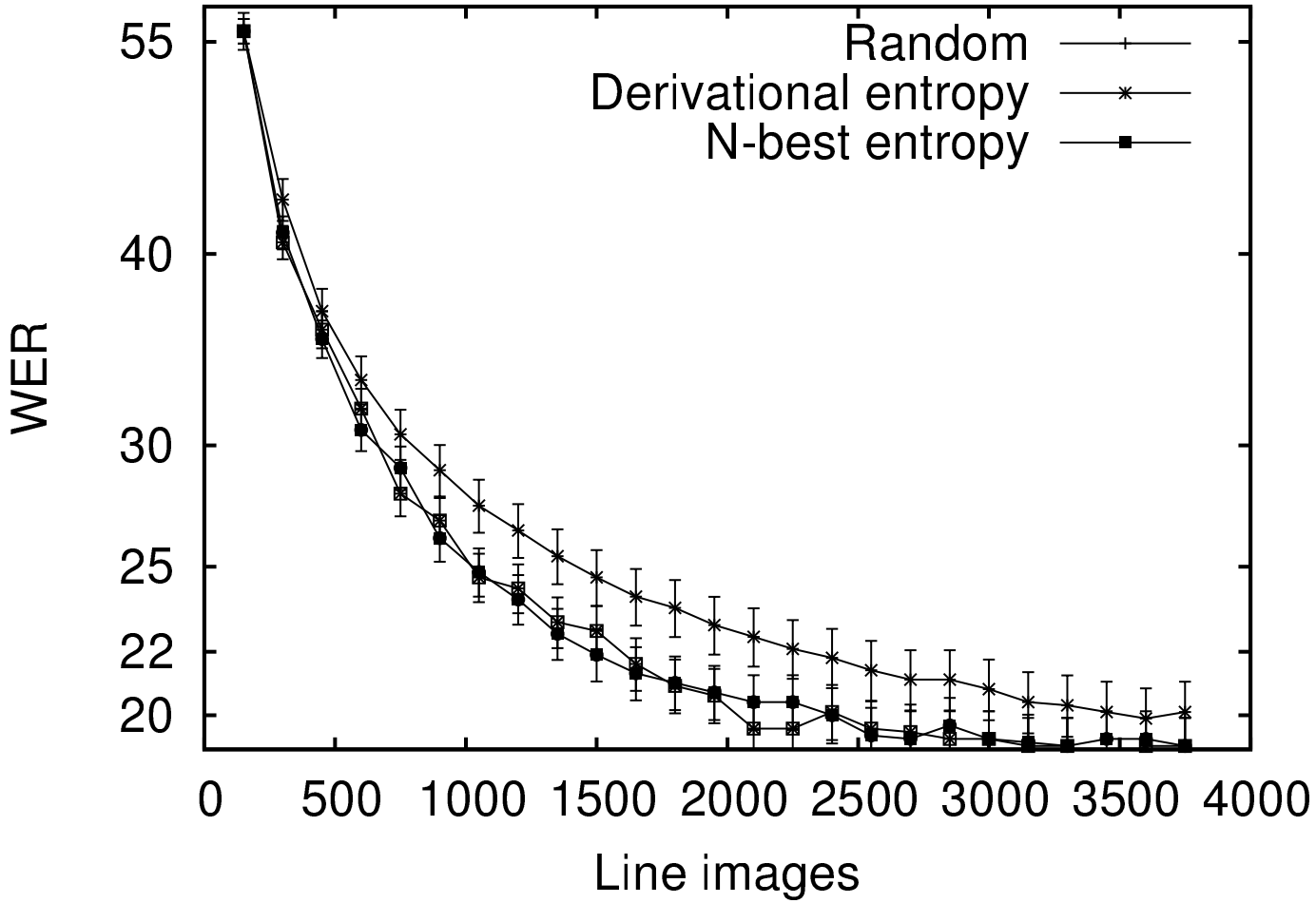
Optical models: Classical HMM

2-gram language model trained with $\mathcal{L}$ in each iteration

In each iteration $B = 150$ (about 5 pages) samples are selected



Active learning



References

- A.L. Berger, V.J. Della Pietra, S. Della Pietra. *A Maximum entropy approach to natural language processing.* Computational Linguistics, 22(1):39–71, 1996.
- C.M. Bishop *Pattern recognition and machine learning.* Springer, 2006.
- T.M. Cover, J.A. Thomas. *Elements of information theory.* John Wiley and Sons, 1991.
- P.S. Gopalakrishnan, D. Kanevsky, A. Nadas, D. Nahamoo, M. A. Picheny, *Decoder selection based on cross-entropies.* ICASSP 1988, pp. 20–23.
- P.S. Gopalakrishnan, D. Kanevsky, A. Nadas, D. Nahamoo. *An inequality for rational functions with applications to some statistical estimation problems.* IEEE Transactions on Information Theory, 37(1):107–113, 1991.
- U. Grenander. *Syntax controlled probabilities. Technical report, Brown University*, Div. of Applied Mathematics, December, 1967.
- D. Hernando, V. Crespi, G. Cybenko. *Efficient computation of the hidden Markov model entropy for a given observation sequence*. IEEE Transactions of Information Theory, 51(7):2681–2685, 2005.
- M. Maca, J.M. Benedí, J.A. Sánchez. *Discriminative learning for PCFG based on the generalized h-criterion*, arXiv:2103.08656 [cs.CL], 2021.
- R. Malouf. *A comparison of algorithms for maximum entropy parameter estimation*, COLING, 1-7, 2002.
- Kevin P. Murphy. *A Probabilistic Machine Learning An Introduction*, MIT Press, 2022.
- D. Povey, P.C. Woodland. *Improved discriminative training techniques for large vocabulary continuous speech recognition.* ICASSP 2001.
- D. Povey, P.C. Woodland. *Mimimum phone error and I-smoothing for improved discriminative training.* ICASSP 2002.
- A. Ratnaparkhi. *Learning to parse natural language with maximum entropy models.* Machine Learning, 34, 151–175, 1999.
- S. Ravi, K. Knight, K. *Bayesian inference for zodiac and other homophonic ciphers.* ACL, 2011.
- V. Romero, A Fornés, N, Serrano, J.A. Sánchez, A.H. Toselli, V. Frinken, E. Vidal, J. Lladós. *The ESPOSALLES database: An ancient marriage license corpus for off-line handwriting recognition*. Pattern Recognition,46 (6), 1658-1669, 2013.
- J.A. Sánchez, U. Pal *Handwritten text recognition for Bengali*, ICFHR, 2016.

References

- J.A. Sánchez, M.A. Rocha, V. Romero, M. Villegas. *On the derivational entropy of left-to-right probabilistic finite-state automata and hidden Markov models.* Computational Linguistics, 44(1), 17-37 2018.
- J.A. Sánchez, V. Romero, A.H. Toselli, M. Villegas, E. Vidal. *A set of benchmarks for Handwritten Text Recognition on historical documents*. Pattern Recognition 94, 122-134, 2019.
- R. Schlüter, W. Macherey, B. Müller, H. Ney. *Comparison of discriminative training criteria and optimization methods for speech recognition*. Speech Communication, 34:287-310, 2001,
- B. Settles, M. Craven, *An analysis of active learning strategies for sequence labelling tasks.* EMNLP, 2008, 1069–1078.
- S. Soule. *Entropies of probabilistic grammars.* Information and Control, 25:57–74, 1974.
- R.A. Thompson *Determination of probabilistic grammars for funtionally specified probability-measure languages.* IEEE Transactions of Computers, c-23(6):603–614, 1974.
- H. Li, K. Zhang, T. Jiang. *The regularized EM algorithm.* 807–812, AAAI, 2005
- X. Yi, C. Caramanis. *Regularized em algorithms: A unified framework and statistical guarantees.*, NIPS, 2015.
- P. Houdouin, E. Ollila, F. Pascal. *Regularized EM algorithm*. ICASSP 2023.

Appendix A. Entropy measures for grammatical models

$$
\begin{aligned}
& p(aba)\ \log\ p(aba) + p(abba)\ \log\ p(abba) + \ldots \\
= \quad & .4\ .3\ .7\ \log\ .4\ .3\ .7 + .4\ .3^2\ .7\ \log\ .4\ .3^2\ .7 + .4\ .3^3\ .7\ \log\ .4\ .3^3\ .7 + \ldots \\
= \quad & .28\ .3\ \log\ .28\ .3 + .28\ .3^2\ \log\ .28\ .3^2 + .28\ .3^3\ \log\ .28\ .3^3 + \ldots \\
= \quad & .28\ .3\ (\log\ .28 + \log\ .3) + .28\ .3^2\ (\log\ .28 + \log\ .3^2) + .28\ .3^3\ (\log\ .28 + \log\ .3^3) + \ldots \\
= \quad & .28\ .3\ \log\ .28 + .28\ .3^2\ \log\ .28 + .28\ .3^3\ \log\ .28 + \ldots + .28\ .3\ \log\ .3 + .28\ .3^2\ \log\ .3^2 + .28\ .3^3\ \log\ .3^3 + \ldots \\
= \quad & .28\ \log\ .28 \sum_{n \geq 1} .3^n + .28\ .3\ \log\ .3 + .28\ .3^2\ 2\ \log\ .3 + .28\ .3^3\ 3\ \log\ .3 + \ldots \\
= \quad & .28\ \log\ .28 \sum_{n \geq 1} .3^n + .28\ \log .3 \sum_{n \geq 1} n\ .3^n \\
= \quad & .28\ \log .28\ \frac{1}{1 - .3} + .28\ \log .3\ \frac{1}{(1 - .3)^2} \\
= \quad & .28\ \log .28\ 1.43 + .28\ \log .3\ 2.04 = -1.78
\end{aligned}
$$