Introduction to Probabilistic Graphical Models

Christoph Lampert

IST Austria (Institute of Science and Technology Austria)

Laplace Smoothing

Maximum likelihood estimation assigns 0 probability to any outcome it has not seen.
This can have unfortunate consequences:

Laplace Smoothing

Maximum likelihood estimation assigns 0 probability to any outcome it has not seen.
This can have unfortunate consequences:

- simplest probabilistic text model: $p(D) = \prod_i p(w_i)$ *"bag of words"*
- how to estimate p ?
- take an English text: $D = (w_1, w_2, \ldots, w_n)$ where each w_i is a word
- estimate the probability, $\hat{p}_{ML}(w)$, of each English word w using maximum likelihood
- take another English text: $D' = (w'_1, w'_2, \ldots, w'_{n'})$. What is $\hat{p}_{ML}(D')$

Laplace Smoothing

Maximum likelihood estimation assigns 0 probability to any outcome it has not seen.
This can have unfortunate consequences:

- simplest probabilistic text model: $p(D) = \prod_i p(w_i)$ *"bag of words"*
- how to estimate p ?
- take an English text: $D = (w_1, w_2, \ldots, w_n)$ where each w_i is a word
- estimate the probability, $\hat{p}_{ML}(w)$, of each English word w using maximum likelihood
- take another English text: $D' = (w'_1, w'_2, \ldots, w'_{n'})$. What is $\hat{p}_{ML}(D')$
- most likely 0, namely whenever D' contains a word w not present in D, so $\hat{p}_{ML}(w) = 0$

Laplace Smoothing

Maximum likelihood estimation assigns 0 probability to any outcome it has not seen.
This can have unfortunate consequences:

- simplest probabilistic text model: $p(D) = \prod_i p(w_i)$ *"bag of words"*
- how to estimate p ?
- take an English text: $D = (w_1, w_2, \ldots, w_n)$ where each w_i is a word
- estimate the probability, $\hat{p}_{ML}(w)$, of each English word w using maximum likelihood
- take another English text: $D' = (w'_1, w'_2, \ldots, w'_{n'})$. What is $\hat{p}_{ML}(D')$
- most likely 0, namely whenever D' contains a word w not present in D, so $\hat{p}_{ML}(w) = 0$

How to overcome?

Laplace Smoothing

Maximum likelihood estimation assigns 0 probability to any outcome it has not seen.
This can have unfortunate consequences:

- simplest probabilistic text model: $p(D) = \prod_i p(w_i)$ *"bag of words"*
- how to estimate p ?
- take an English text: $D = (w_1, w_2, \ldots, w_n)$ where each w_i is a word
- estimate the probability, $\hat{p}_{ML}(w)$, of each English word w using maximum likelihood
- take another English text: $D' = (w'_1, w'_2, \ldots, w'_{n'})$. What is $\hat{p}_{ML}(D')$
- most likely 0, namely whenever D' contains a word w not present in D, so $\hat{p}_{ML}(w) = 0$

How to overcome? $\hat{p}_{ML}(x) = \dfrac{n_x}{n} \quad \rightarrow \quad \hat{p}_{\alpha}(x) = \dfrac{n_x + \alpha}{n + L\alpha}$ *"Laplace smoothing"*

- where n_x is the number of counts of any $x \in \mathcal{X}$,
- $L = |\mathcal{X}|$ is the number of states,
- α is a small value, e.g. 1, or $\frac{1}{2}$, or $\frac{1}{L}$. also: "pseudo-count"

Maximum A-Posteriori Parameter Estimation

Role of the prior

Imagine a game:

- a roll a die five times: 1, 5, 2, 1, 3, 5 $\quad\rightarrow\quad \hat{p}_{ML}(x) = (\frac{1}{3}, \frac{1}{6}, \frac{1}{6}, 0, \frac{1}{3}, 0)$
- Now I offer you a bet:
  - I roll the die once more: if I roll a 6, you pay me 100 Euros, otherwise, I pay you 10 Euros.
  - Do you accept?

Maximum A-Posteriori Parameter Estimation

Role of the prior

Imagine a game:

- a roll a die five times: 1, 5, 2, 1, 3, 5 $\rightarrow$ $\hat{p}_{ML}(x) = (\frac{1}{3}, \frac{1}{6}, \frac{1}{6}, 0, \frac{1}{3}, 0)$
- Now I offer you a bet:
  - I roll the die once more: if I roll a 6, you pay me 100 Euros, otherwise, I pay you 10 Euros.
  - Do you accept?

Possibly not, even though maximum likelihood says yes:

$$\hat{p}_{ML}(6) = 0 \quad \rightarrow \quad \mathbb{E}_{x \sim \hat{p}_{ML}}[\text{outcome}] = 0 \cdot (-100) + 1 \cdot 10 = 10$$

What about Laplace-smoothing? For $\alpha = 1$: $\hat{p}_1(x) = (\frac{1}{4}, \frac{1}{6}, \frac{1}{6}, \frac{1}{12}, \frac{1}{4}, \frac{1}{12})$

$$\hat{p}_{\alpha=1}(6) = \frac{1}{12} \quad \rightarrow \quad \mathbb{E}_{x \sim \hat{p}_1}[\text{outcome}] = \frac{1}{12}(-100) + \frac{11}{12}10 = \frac{5}{6} > 0$$

Maximum A-Posteriori Parameter Estimation

Role of the prior

Imagine a game:

- a roll a die five times: 1, 5, 2, 1, 3, 5 $\rightarrow$ $\hat{p}_{ML}(x) = (\frac{1}{3}, \frac{1}{6}, \frac{1}{6}, 0, \frac{1}{3}, 0)$
- Now I offer you a bet:
  - I roll the die once more: if I roll a 6, you pay me 100 Euros, otherwise, I pay you 10 Euros.
  - Do you accept?

Possibly not, even though maximum likelihood says yes:

$$\hat{p}_{ML}(6) = 0 \quad \rightarrow \quad \mathbb{E}_{x \sim \hat{p}_{ML}}[\text{outcome}] = 0 \cdot (-100) + 1 \cdot 10 = 10$$

What about Laplace-smoothing? For $\alpha = 1$: $\hat{p}_1(x) = (\frac{1}{4}, \frac{1}{6}, \frac{1}{6}, \frac{1}{12}, \frac{1}{4}, \frac{1}{12})$

$$\hat{p}_{\alpha=1}(6) = \frac{1}{12} \quad \rightarrow \quad \mathbb{E}_{x \sim \hat{p}_1}[\text{outcome}] = \frac{1}{12}(-100) + \frac{11}{12}10 = \frac{5}{6} > 0$$

So why not?

Maximum A-Posteriori Parameter Estimation

Role of the prior

Imagine a game:

- a roll a die five times: 1, 5, 2, 1, 3, 5 $\quad\rightarrow\quad \hat{p}_{ML}(x) = (\frac{1}{3}, \frac{1}{6}, \frac{1}{6}, 0, \frac{1}{3}, 0)$
- Now I offer you a bet:
  - I roll the die once more: if I roll a 6, you pay me 100 Euros, otherwise, I pay you 10 Euros.
  - Do you accept?

Possibly not, even though maximum likelihood says yes:

$$\hat{p}_{ML}(6) = 0 \quad \rightarrow \quad \mathbb{E}_{x\sim\hat{p}_{ML}}[\text{outcome}] = 0 \cdot (-100) + 1 \cdot 10 = 10$$

What about Laplace-smoothing? For $\alpha = 1$: $\quad \hat{p}_1(x) = (\frac{1}{4}, \frac{1}{6}, \frac{1}{6}, \frac{1}{12}, \frac{1}{4}, \frac{1}{12})$

$$\hat{p}_{\alpha=1}(6) = \frac{1}{12} \quad \rightarrow \quad \mathbb{E}_{x\sim\hat{p}_1}[\text{outcome}] = \frac{1}{12}(-100) + \frac{11}{12}10 = \frac{5}{6} > 0$$

So why not? Most likely, you have a prior belief about what probabilities to expect!

Maximum A-Posteriori Parameter Estimation

- We treated θ as a random variable instead of unknown fixed value.
- for any fixed θ, we have a distribution over x: $p(x;\theta) \rightarrow p(x|\theta)$
- for data $x_1, \ldots, x_n$, we interested in $p(\theta|x_1, \ldots, x_n)$

$$p(\theta|x_1, \ldots, x_n) \overset{\text{Bayes rule}}{=} \frac{p(x_1, \ldots, x_n|\theta)p(\theta)}{p(x_1, \ldots, x_n)}$$

Maximum A-Posteriori Parameter Estimation

- We treated θ as a random variable instead of unknown fixed value.
- for any fixed θ, we have a distribution over x: $p(x;\theta) \to p(x|\theta)$
- for data $x_1, \dots, x_n$, we interested in $p(\theta|x_1, \dots, x_n)$

$$p(\theta|x_1, \dots, x_n) \overset{\text{Bayes rule}}{=} \frac{p(x_1, \dots, x_n|\theta)p(\theta)}{p(x_1, \dots, x_n)}$$

- what's the most likely value for θ? maximum a-posteriori (MAP) estimate

$$\begin{aligned}\hat{\theta}_{MAP} &= \underset{\theta}{\operatorname{argmax}}\, p(\theta|x_1, \dots, x_n) = \underset{\theta}{\operatorname{argmax}}\, p(x_1, \dots, x_n|\theta)p(\theta) \\ &= \underset{\theta}{\operatorname{argmax}}\, \underbrace{p(\theta)}_{\text{Prior}} \underbrace{\prod_{i=1}^{n} p(x_i)|\theta)}_{\text{data likelihood}} = \underset{\theta}{\operatorname{argmax}} \left[\underbrace{\log p(\theta)}_{\text{log-prior}} + \underbrace{\sum_{i=1}^{n} \log p(x_i)|\theta)}_{\text{data log-likelihood}} \right]\end{aligned}$$

Maximum A-Posteriori Parameter Estimation

Maximum likelihood estimator for coin toss

We need a prior! How likely are different parameter values (without having seen data)?

- $p(\theta) = 1$ for all $\theta \in [0, 1]$

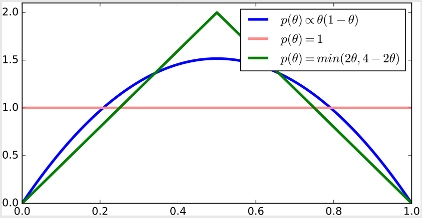
$$\hat{\theta}_{MAP} = \frac{n_{\text{head}}}{n} = \hat{\theta}$$

- $p(\theta) \propto \theta(1-\theta)$ (more mass at $\theta = \frac{1}{2}$)

$$\hat{\theta}_{MAP} = \frac{n_{\text{head}} + 1}{n + 2}$$

- $p(\theta) = 2\min(\theta, 2-\theta)$ (also more mass at $\theta = \frac{1}{2}$)

no simple expression for $\hat{\theta}_{MAP}$



Maximum A-posteriori estimation for coin toss

A prior should reflect our belief, but not destroy tractability of computations.

- a prior such that $p(\theta|x)$ has same parametric form as $p(\theta)$ is called conjugate.
- Coin example: $p(x_1, \ldots, x_n|\theta) = \theta^{n_{\text{head}}}(1-\theta)^{n-n_{\text{head}}}$
- Conjugate prior for θ: $p(\theta) \propto \theta^{a-1}(1-\theta)^{b-1}$ "beta distribution" Beta(a, b)
- Posterior distribution:
  $p(\theta|x_1, \ldots, x_n) \propto p(x_1, \ldots, x_n|\theta)p(\theta) = \theta^{a-1+n_{\text{head}}}(1-\theta)^{b-1+n-n_{\text{head}}}$
- MAP estimate: $\hat{\theta}_{\text{MAP}} = \dfrac{a-1+n_{\text{head}}}{n+a+b-2}$
- special cases:
  - $a=1, b=1$: $p(\theta) = 1$
  - $a=2, b=2$: $p(\theta) \propto \theta(1-\theta)$

  in both cases, we were still able to compute $\hat{\theta}_{MAP}$

A Fully Bayesian Treatment

- $\hat{\theta}_{ML}$ and $\hat{\theta}_{MAP}$ are just **point estimates** for θ

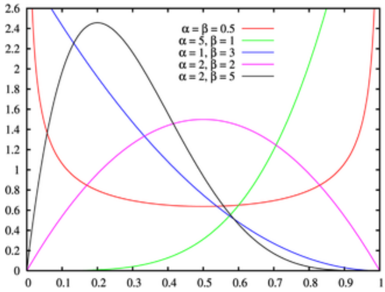
A Fully Bayesian Treatment

- $\hat{\theta}_{ML}$ and $\hat{\theta}_{MAP}$ are just **point estimates** for θ
- Maybe the full posterior distribution contains more information?

$$p(\theta|x_1,\ldots,x_n) \propto \theta^{a-1+n_{\text{head}}}(1-\theta)^{b-1+n-n_{\text{head}}}$$

- $p(\theta|x_1,\ldots,x_n)$ is a beta-distribution

$$\text{Beta}(t \mid \alpha,\beta) = \frac{1}{B(\alpha,\beta)} t^{\alpha-1}(1-t)^{\beta-1}$$



Examples of Beta distributions

A Fully Bayesian Treatment

- $\hat{\theta}_{ML}$ and $\hat{\theta}_{MAP}$ are just **point estimates** for θ
- Maybe the full posterior distribution contains more information?

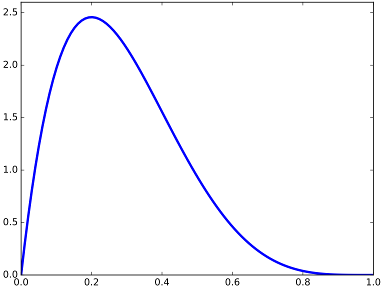
$$p(\theta|x_1, \ldots, x_n) \propto \theta^{a-1+n_{\text{head}}}(1-\theta)^{b-1+n-n_{\text{head}}}$$

- $p(\theta|x_1, \ldots, x_n)$ is a beta-distribution

$$\text{Beta}(t \mid \alpha, \beta) = \frac{1}{B(\alpha, \beta)} t^{\alpha-1}(1-t)^{\beta-1}$$

For example, at $\alpha = 2, \beta = 5$:

- asymmetric/skewed



$\text{Beta}(2, 5)$

A Fully Bayesian Treatment

- $\hat{\theta}_{ML}$ and $\hat{\theta}_{MAP}$ are just **point estimates** for θ
- Maybe the full posterior distribution contains more information?

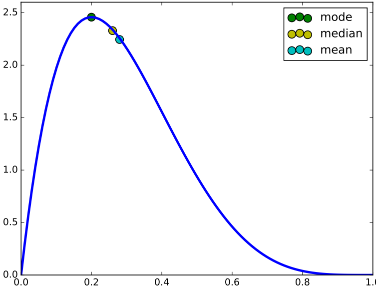
$$p(\theta|x_1, \ldots, x_n) \propto \theta^{a-1+n_{\text{head}}}(1-\theta)^{b-1+n-n_{\text{head}}}$$

- $p(\theta|x_1, \ldots, x_n)$ is a beta-distribution

$$\text{Beta}(t \mid \alpha, \beta) = \frac{1}{B(\alpha, \beta)} t^{\alpha-1}(1-t)^{\beta-1}$$

For example, at $\alpha = 2, \beta = 5$:

- asymmetric/skewed
  - maximum at $t = \frac{\alpha-1}{\alpha+\beta-2}$. Here $t = 0.2$
  - median at $t \approx \frac{\alpha-\frac{1}{3}}{\alpha+\beta-\frac{2}{3}}$. Here: $t \approx 0.26$:
  - mean at $t = \frac{\alpha}{\alpha+\beta}$. Here $t \approx 0.28$



Beta$(2, 5)$

A Fully Bayesian Treatment

- $\hat{\theta}_{ML}$ and $\hat{\theta}_{MAP}$ are just **point estimates** for θ
- Maybe the full posterior distribution contains more information?

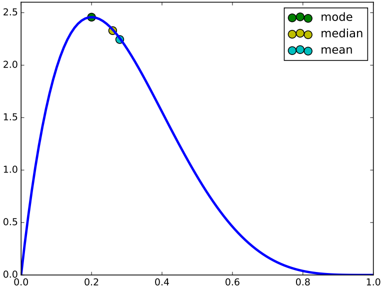
$$p(\theta|x_1,\ldots,x_n) \propto \theta^{a-1+n_{\text{head}}}(1-\theta)^{b-1+n-n_{\text{head}}}$$

- $p(\theta|x_1,\ldots,x_n)$ is a beta-distribution

$$\text{Beta}(t \mid \alpha,\beta) = \frac{1}{B(\alpha,\beta)} t^{\alpha-1}(1-t)^{\beta-1}$$

For example, at $\alpha = 2, \beta = 5$:

- asymmetric/skewed
  - maximum at $t = \frac{\alpha-1}{\alpha+\beta-2}$. Here $t = 0.2$
  - median at $t \approx \frac{\alpha-\frac{1}{3}}{\alpha+\beta-\frac{2}{3}}$. Here: $t \approx 0.26$:
  - mean at $t = \frac{\alpha}{\alpha+\beta}$. Here $t \approx 0.28$



Beta(2, 5)

Common choice for "Bayesians": posterior mean $\hat{\theta}_{PM} = \mathbb{E}_{\theta\sim p(\theta|\mathcal{D})}[\theta]$

Maximum A-Posteriori vs. Maximum Likelihood vs. Bayesian

Maximum A-Posteriori vs. Maximum Likelihood vs. Bayesian

Maximum likelihood

+ usually the easiest to use
+ consistent estimator, if model distribution is correct
– hard to include prior knowledge, *e.g.* reasonable ranges
– overconfident if little data is available, e.g. probability is 0 for never-seen values

Maximum A-Posteriori vs. Maximum Likelihood vs. Bayesian

Maximum likelihood

+ usually the easiest to use
+ consistent estimator, if model distribution is correct
– hard to include prior knowledge, *e.g.* reasonable ranges
– overconfident if little data is available, e.g. probability is 0 for never-seen values

Maximum a-posteriori

+ can reflect prior knowledge, *e.g.* known parameter ranges
+ more robust: if n is small, estimate stays close to prior
– not always clear how to chose a prior
– computationally more challenging, especially if no conjugate prior is used

Maximum A-Posteriori vs. Maximum Likelihood vs. Bayesian

Maximum likelihood

+ usually the easiest to use
+ consistent estimator, if model distribution is correct
– hard to include prior knowledge, *e.g.* reasonable ranges
– overconfident if little data is available, e.g. probability is 0 for never-seen values

Maximum a-posteriori

+ can reflect prior knowledge, *e.g.* known parameter ranges
+ more robust: if n is small, estimate stays close to prior
– not always clear how to chose a prior
– computationally more challenging, especially if no conjugate prior is used

Bayesian

+ same advantages of maximum a-posteriori
+ information about uncertainty of estimate
– same disadvantages as maximum a-posteriori, computationally even more challenging

Maximum A-Posteriori vs. Maximum Likelihood vs. Bayesian

Maximum likelihood

+ usually the easiest to use
+ consistent estimator, if model distribution is correct
- hard to include prior knowledge, *e.g.* reasonable ranges
- overconfident if little data is available, e.g. probability is 0 for never-seen values

Maximum a-posteriori

+ can reflect prior knowledge, *e.g.* known parameter ranges
+ more robust: if n is small, estimate stays close to prior
- not always clear how to chose a prior
- computationally more challenging, especially if no conjugate prior is used

Bayesian

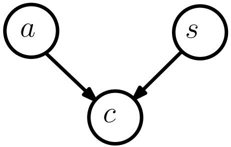
+ same advantages of maximum a-posteriori
+ information about uncertainty of estimate
- same disadvantages as maximum a-posteriori, computationally even more challenging

Note: for $n \to \infty$, data will dominate the prior and all pretty much the same

Maximum Likelihood for Bayesian Networks

Example: Lung Cancer network

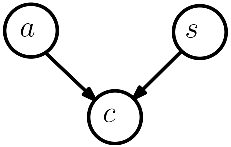
- Patient
  - has lung cancer $c \in \{0, 1\}$
  - was exposed to asbestos $a \in \{0, 1\}$
  - is a smoker $s \in \{0, 1\}$



Example: Lung Cancer network

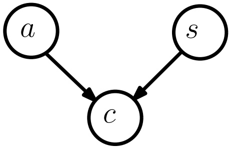
- Patient
  - has lung cancer $c \in \{0, 1\}$
  - was exposed to asbestos $a \in \{0, 1\}$
  - is a smoker $s \in \{0, 1\}$
- Given the following relationship

$$p(a, s, c) = p(c \mid a, s)p(a)p(s)$$



Example: Lung Cancer network

- Patient
  - has lung cancer $c \in \{0, 1\}$
  - was exposed to asbestos $a \in \{0, 1\}$
  - is a smoker $s \in \{0, 1\}$
- Given the following relationship

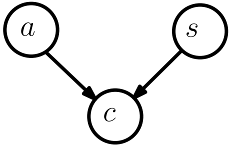


$$p(a, s, c) = p(c \mid a, s)p(a)p(s)$$

- What are the parameters to learn?

Example: Lung Cancer network

- Patient
  - has lung cancer $c \in \{0, 1\}$
  - was exposed to asbestos $a \in \{0, 1\}$
  - is a smoker $s \in \{0, 1\}$
- Given the following relationship



$$p(a, s, c) = p(c \mid a, s)p(a)p(s)$$

- What are the parameters to learn? Conditional probability tables (CPT)

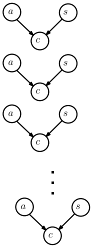
$$\theta^a = p(a = 1) \in \mathbb{R}, \qquad \theta^s = p(s = 1) \in \mathbb{R},$$
$$\theta^c = \left(\theta^c_{a=0,s=0},\ \theta^c_{a=0,s=1},\ \theta^c_{a=1,s=0},\ \theta^c_{a=1,s=1}\right) \in \mathbb{R}^4$$

with $\quad \theta^c_{a=i,s=j} = p(c = 1|a = i, s = j).$

Example: Lung Cancer network

We observe N patients: observations $\mathcal{D} = \{(a_1, s_1, c_1), (a_2, s_2, c_2), \ldots\}$

a	s	c
1	1	1
1	0	0
0	1	1
0	1	0
1	1	1
0	0	0
1	0	1



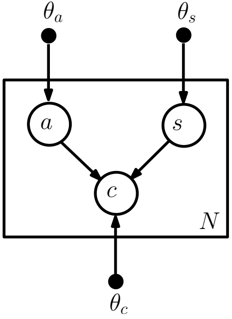


plate notation

Example: Lung Cancer network

$$p(a, s, c) = p(c \mid a, s)p(a)p(s)$$

- Log-likelihood

$$\log \mathcal{L}(\theta; \mathcal{D}) = \sum_i \log p(a_i, s_i, c_i) = \sum_i \log p(a_i; \theta_a) + \sum_i \log p(s_i; \theta_s) + \sum_i \log p(c_i \mid a_i, s_i; \theta_c)$$

Example: Lung Cancer network

$$p(a, s, c) = p(c \mid a, s)p(a)p(s)$$

- Log-likelihood

$$\log \mathcal{L}(\theta; \mathcal{D}) = \sum_i \log p(a_i, s_i, c_i) = \sum_i \log p(a_i; \theta_a) + \sum_i \log p(s_i; \theta_s) + \sum_i \log p(c_i \mid a_i, s_i; \theta_c)$$

Now we count:

- Denote $n_{a=0,s=0,c=0} = \sum_i [\![a_i = 0 \wedge s_i = 0 \wedge c_i = 0]\!]$ (count number of cases)
- Analogously $n_{a=0,s=0,c=1}, \ldots, n_{a=1,s=1,c=1}$

Example: Lung Cancer network

$$p(a, s, c) = p(c \mid a, s)p(a)p(s)$$

- Log-likelihood

$$\log \mathcal{L}(\theta; \mathcal{D}) = \sum_i \log p(a_i, s_i, c_i) = \sum_i \log p(a_i; \theta_a) + \sum_i \log p(s_i; \theta_s) + \sum_i \log p(c_i \mid a_i, s_i; \theta_c)$$

Now we count:

- Denote $n_{a=0,s=0,c=0} = \sum_i [\![a_i = 0 \wedge s_i = 0 \wedge c_i = 0]\!]$ (count number of cases)
- Analogously $n_{a=0,s=0,c=1}, \ldots, n_{a=1,s=1,c=1}$

Collapse terms in log-likelihood according to value combinations:

$$\begin{aligned}\log \mathcal{L}(\theta; \mathcal{D}) = {} & n_{a=0} \log p(a = 0) + n_{a=1} \log p(a = 1) + n_{s=0} \log p(s = 0) + n_{s=1} \log p(s = 1) \\ & + n_{a=0,s=0,c=0} \log p(c = 0 | a = 0, s = 0) + \ldots \\ & + n_{a=1,s=1,c=1} \log p(c = 1 | a = 1, s = 1)\end{aligned}$$

Example: Lung Cancer network

Express in terms of parameters:

$$\log \mathcal{L}(\theta) = n_{a=0} \log(1-\theta^a) + n_{a=1}\theta^a + n_{s=0} \log(1-\theta^s) + n_{s=1}\theta^s \\ + n_{a=0,s=0,c=0} \log(1-\theta^c_{a=0,s=0}) + \cdots + n_{a=1,s=1,c=1}\theta^c_{a=1,s=0}$$

with conditional probability tables as parameters

- $\theta^a = p(a=1)$
- $\theta^s = p(s=1)$
- $\theta^c_{a=0,s=0} = p(c=1|a=0,s=0)$
- $\theta^c_{a=0,s=1} = p(c=1|a=0,s=1)$
- $\theta^c_{a=1,s=0} = p(c=1|a=1,s=0)$
- $\theta^c_{a=1,s=1} = p(c=1|a=1,s=1)$

Note: no interaction between parameters. We can optimize for each of them separately.

Example: Lung Cancer network

- For example, $\theta^c_{a=1,s=0}$

$$\log \mathcal{L}(\theta) = n_{a=1,s=0,c=1} \log \theta^c_{a=1,s=0} + n_{a=1,s=0,c=0} \log(1 - \theta^c_{a=1,s=0}) \quad + \text{const.}$$

Example: Lung Cancer network

- For example, $\theta^c_{a=1,s=0}$

$$\log \mathcal{L}(\theta) = n_{a=1,s=0,c=1} \log \theta^c_{a=1,s=0} + n_{a=1,s=0,c=0} \log(1 - \theta^c_{a=1,s=0}) \quad + \text{const.}$$

- Setting the derivative to 0

$$\frac{n_{a=1,s=0,c=1}}{\hat{\theta}^c_{a=1,s=0}} - \frac{n_{a=1,s=0,c=0}}{(1 - \hat{\theta}^c_{a=1,s=0})} = 0$$

Example: Lung Cancer network

- For example, $\theta^{c}_{a=1,s=0}$

$$\log \mathcal{L}(\theta) = n_{a=1,s=0,c=1} \log \theta^{c}_{a=1,s=0} + n_{a=1,s=0,c=0} \log(1 - \theta^{c}_{a=1,s=0}) \quad + \text{const.}$$

- Setting the derivative to 0

$$\frac{n_{a=1,s=0,c=1}}{\hat{\theta}^{c}_{a=1,s=0}} - \frac{n_{a=1,s=0,c=0}}{(1 - \hat{\theta}^{c}_{a=1,s=0})} = 0$$

- Therefore

$$\hat{\theta}^{c}_{a=1,s=0} = \frac{n_{a=1,s=0,c=1}}{n_{a=1,s=0,c=0} + n_{a=1,s=0,c=1}}$$

Example: Lung Cancer network

- For example, $\theta^c_{a=1,s=0}$

$$\log \mathcal{L}(\theta) = n_{a=1,s=0,c=1} \log \theta^c_{a=1,s=0} + n_{a=1,s=0,c=0} \log(1 - \theta^c_{a=1,s=0}) \quad + \text{const.}$$

- Setting the derivative to 0

$$\frac{n_{a=1,s=0,c=1}}{\hat{\theta}^c_{a=1,s=0}} - \frac{n_{a=1,s=0,c=0}}{(1 - \hat{\theta}^c_{a=1,s=0})} = 0$$

- Therefore

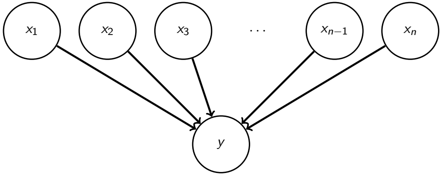
$$\hat{\theta}^c_{a=1,s=0} = \frac{n_{a=1,s=0,c=1}}{n_{a=1,s=0,c=0} + n_{a=1,s=0,c=1}}$$

Maximum Likelihood solution corresponds to empirical counts, just like in coin example!

Maximum Likelihood for CPTs

Unfortunately, sometimes, counting is not practical or possible:

- CPT might be too large



(L^n parameters even for L-state variables)

- not enough data (most counts would be zero)
- continuous variables, $x_1, \ldots, x_d \in \mathbb{R}$
- missing data: *e.g.* hidden Markov model
  "observations" are observed, but "hidden states" are not $\rightarrow$ "latent variable models"

Learning mixture models

Mixture Models

A mixture model is one in which a set of simpler models is combined to produce a richer model:

- We observe and care about a random variable V, that does not have a simple distribution.
- We model it as a generated by a two-stage procedure
  - Sample the state of an auxiliary variable $H \sim p(h)$
  - Given the value h of H, sample the value of v from a h-dependent distribution $p(v|h)$

$$H \longrightarrow V \qquad p(v, h) = p(v|h)p(h) \qquad p(v) = \sum_{h \in \mathcal{H}} p(v|h)p(h)$$

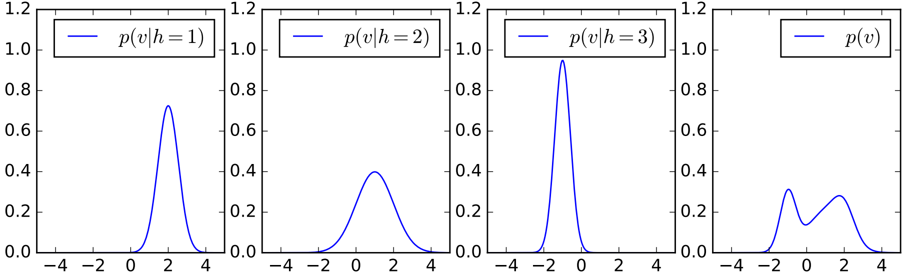
The variable V is visible or observable, while H is hidden or latent.

Note: the effect of the hidden H might be 'real', or just a computational trick.

Mixture Models

Example: Gaussian Mixture Model (GMM)

For $h \in \{1, 2, \ldots, K\}$, each $p(v|h) = \mathcal{N}(x; \mu_h, \Sigma_h)$



If we only see sample $v_1, \ldots, v_n$, can we learn $p(h)$ and $p(v|h)$?

Mixture Models

Example: Gaussian Mixture Model (GMM)

For $h \in \{1, 2, \ldots, K\}$, each $p(v|h) = \mathcal{N}(x; \mu_h, \Sigma_h)$

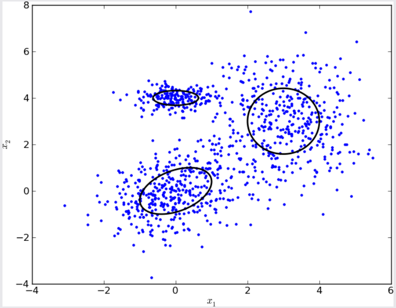


Image: http://pypr.sourceforge.net/

If we only see sample $v_1, \ldots, v_n$, can we learn $p(h)$ and $p(v|h)$?

Maximum Likelihood Estimation for GMMs

- data: $v_1, \dots, v_n$
- parameters:
  - $\pi := \big(p(h=1), \dots, p(h=K)\big) \in \mathbb{R}^K$
  - $\mu_1, \dots, \mu_K$ with $\mu_k \in \mathbb{R}^d$ for $k = 1, \dots, K$
  - $\Sigma_1, \dots, \Sigma_K$ with $\Sigma_k \in \mathbb{R}^{d \times d}$ for $k = 1, \dots, K$
- model:

$$p(v) = \sum_{k=1}^{K} \pi_k \frac{1}{\sqrt{(2\pi)^d |\Sigma_k|}} e^{-\frac{1}{2}(v-\mu_k)^\top \Sigma^{-1}(v-\mu_k)}$$

- data likelihood:

$$p(v_1, \dots, v_n) = \prod_{i=1}^{n} p(v_i) = \prod_{i=1}^{n} \sum_{k=1}^{K} \pi_k \frac{1}{\sqrt{(2\pi)^d |\Sigma_k|}} e^{-\frac{1}{2}(v_i-\mu_k)^\top \Sigma_k^{-1}(v_i-\mu_k)}$$

No closed-form expressions as for single Gaussian maximum likelihood estimation
$\rightarrow$ numeric optimization, e.g. gradient descent

Expectation Maximization (EM) Algorithm for GMMs

Thinking of the generating process:

- for each example: sample a hidden value $h_i \sim p(h)$, then sample $v_i \sim p(v|h_i)$
- if we knew $h_1, \ldots, h_n$,
  - we could split data into groups, $\{v_i : h_i = k\}$, and
  - estimate $p(v|h)$ separately for each value of h
- in practice, we don't know h_i, but if we had $p(v, h)$, we could estimate: $p(h|v_i)$

Expectation Maximization (EM) Algorithm for GMMs

Thinking of the generating process:

- for each example: sample a hidden value $h_i \sim p(h)$, then sample $v_i \sim p(v|h_i)$
- if we knew $h_1, \ldots, h_n$,
    - we could split data into groups, $\{v_i : h_i = k\}$, and
    - estimate $p(v|h)$ separately for each value of h
- in practice, we don't know h_i, but if we had $p(v, h)$, we could estimate: $p(h|v_i)$

Chicken and egg:

- to get a good model $p(v)$, we need $p(h|v)$
- to get $p(h|v)$, we need a good model of $p(v, h)$

Expectation Maximization (EM) Algorithm for GMMs

Thinking of the generating process:

- for each example: sample a hidden value $h_i \sim p(h)$, then sample $v_i \sim p(v|h_i)$
- if we knew $h_1, \ldots, h_n$,
  - we could split data into groups, $\{v_i : h_i = k\}$, and
  - estimate $p(v|h)$ separately for each value of h
- in practice, we don't know h_i, but if we had $p(v, h)$, we could estimate: $p(h|v_i)$

Chicken and egg:

- to get a good model $p(v)$, we need $p(h|v)$
- to get $p(h|v)$, we need a good model of $p(v, h)$

Intuition behind the Expectation Maximization (EM) algorithm:

- alternate between estimating $p(h|v)$, $p(v|h)$ and $p(h)$

EM Algorithm for GMMs [Dempster et al, 1977]

initialize parameters $\Theta = (\pi_1, \ldots, \pi_K, \mu_1, \ldots \mu_K, \Sigma_1, \ldots, \Sigma_K)$

we write $g_k(x) = \mathcal{N}(x; \mu_k, \Sigma_k)$

repeat

E-step

for $i = 1, \ldots, n,\ k = 1 \ldots, K$ **do**

$\gamma_{ik} \leftarrow \frac{\pi_k g_k(v_i)}{\sum_{k=1}^{K} \pi_k g_k(v_i)}$ // "responsibilities" of component k for v_i

end for

M-step

for $k = 1 \ldots, K$ **do**

$n_k \leftarrow \sum_i \gamma_{ik}$ // total weight of components k

$\pi_k \leftarrow \frac{n_k}{n}$ // normalized weight of component k

$\mu_k \leftarrow \frac{1}{n_k} \sum_i \gamma_{ik} v_i$ // mean, weighted by

$\Sigma_k \leftarrow \frac{1}{n_k} \sum_i \gamma_{ik}(v_i - \mu_k)(v_i - \mu_k)^\top$

end for

until convergence

EM Algorithm for GMMs

- $p(h = k) = \pi_k,$
- $p(x|h = k) = g_k(x) = \mathcal{N}(x; \mu_k, \Sigma_k),$
- $p(v) = \sum_h p(v, h) = \sum_{k=1}^{K} p(v|h = k)p(h = k) = \sum_{k=1}^{K} \pi_k g_k(v)$

E-step:

$$p(h = k|v = v_i) = \frac{p(v = v_i, h = k)}{p(v = v_i)} = \frac{\pi_k g_k(v_i)}{\sum_{k=1}^{K} \pi_k g_k(v_i)}$$

EM Algorithm for GMMs

- $p(h = k) = \pi_k$,
- $p(x|h = k) = g_k(x) = \mathcal{N}(x; \mu_k, \Sigma_k)$,
- $p(v) = \sum_h p(v, h) = \sum_{k=1}^{K} p(v|h = k)p(h = k) = \sum_{k=1}^{K} \pi_k g_k(v)$

E-step:

$$p(h = k|v = v_i) = \frac{p(v = v_i, h = k)}{p(v = v_i)} = \frac{\pi_k g_k(v_i)}{\sum_{k=1}^{K} \pi_k g_k(v_i)} \quad \rightarrow \quad \gamma_{ik}$$

EM Algorithm for GMMs

- $p(h = k) = \pi_k$,
- $p(x|h = k) = g_k(x) = \mathcal{N}(x; \mu_k, \Sigma_k)$,
- $p(v) = \sum_h p(v, h) = \sum_{k=1}^K p(v|h = k)p(h) = \sum_{k=1}^K \pi_k g_k(v)$

M-step: for known $h_1, \ldots, h_n$:

$$\log p(v_1, \ldots, v_n, h_1, \ldots h_n) = \log \prod_i p(v_i, h_i) = \log \prod_{i=1}^n g_{h_i}(v_i) = \sum_{k=1}^K \left[\sum_{i=1}^n \delta_{h_i=k} \pi_k \log g_k(v_i) \right]$$

We can do maximum likelihood estimate for each g_k separately, using a subset of the data. If we don't know the h_i? Weigh contribution of each point by how likely it belongs to component k:

$$\min_{\pi,\mu,\sigma} \quad \sum_{k=1}^K \left[\sum_{i=1}^n \gamma_{ik} \pi_k \log g_k(v_i) \right]$$

Derivation of the EM algorithm

We don't really know how to maximize difficult non-convex functions.

Most common is gradient-based optimization (ascent/descent), but it has shortcomings:

- need initialization,
- takes small steps,
- converges to local maximum.

Alternative: turn difficult optimization into sequence of easier ones.

Derivation of the EM algorithm

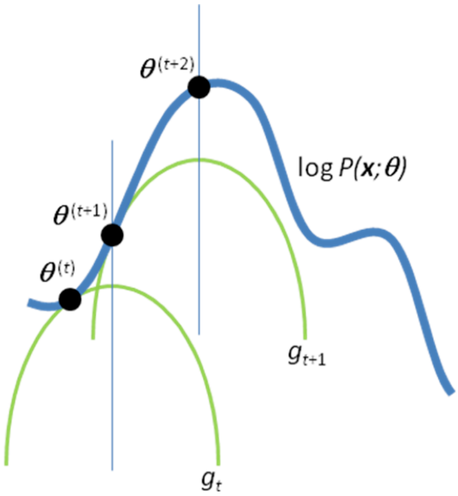


Image: [T. Minka, "Expectation-Maximization as lower bound maximization", TR 1998]

Derivation of the EM algorithm

Change notation from $(v_1, \dots, v_n, h_1, \dots, h_n)$ to (x, z): we want to maximize

$$\mathcal{L}(\theta) = \log p(x;\theta) = \log \sum_z p(x, z; \theta)$$

First observation: it's easy to come up with lower bounds:

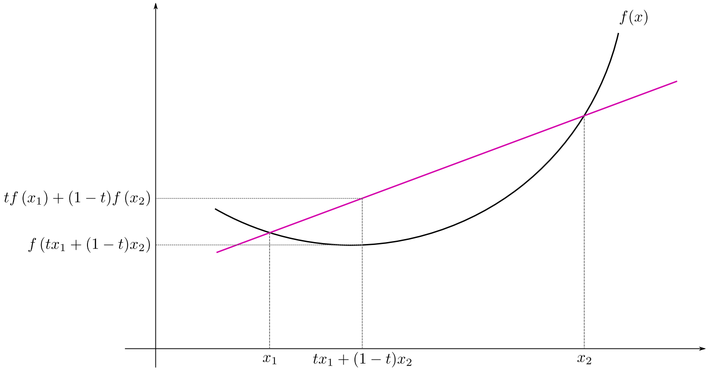
For any function $q(z) \geq 0$ with $\sum_z q(z) = 1$:

$$\begin{aligned}
\log p(x;\theta) &= \log \sum_h p(x, z; \theta) = \log \sum_h q(z) \frac{p(x, z; \theta)}{q(z)} = \log \mathbb{E}_{z \sim q} \left[\frac{p(x, z; \theta)}{q(z)} \right] \\
&\overset{\text{Jensen's ineq.}}{\geq} \mathbb{E}_{z \sim q} \log \left[\frac{p(x, z; \theta)}{q(z)} \right] \\
&= \mathbb{E}_{z \sim q} \log p(x, z; \theta) - \mathbb{E}_{z \sim q} \log q(z) \qquad =: \ G(\theta, q) \text{ "variational lower bound"}
\end{aligned}$$

If $q(z)$ is arbitrary, we didn't lose anything: for $q(z) = p(z|x;\theta)$ the inequality is an equality.

Jensen's inequality

For a convex function $f : \mathbb{R} \to \mathbb{R}$ and any distribution p: $\quad \mathbb{E}_{t\sim p}[f(t)] \leq f(\mathbb{E}_t t)$



For a concave function $f : \mathbb{R} \to \mathbb{R}$ the inequality holds in the opposite direction.

Figure: By Eli Osherovich - Own work, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=10764763

Derivation of the EM algorithm

$$\text{for any } q: \qquad \log p(x;\theta) \;\leq\; \mathbb{E}_{z\sim q} \log p(x,z;\theta) - \mathbb{E}_{z\sim q} \log q(z) \qquad =: G(\theta, q)$$

Derivation of the EM algorithm

for any q: $\log p(x;\theta) \;\leq\; \mathbb{E}_{z\sim q} \log p(x,z;\theta) - \mathbb{E}_{z\sim q} \log q(z) \quad =: G(\theta, q)$

Coordinate ascent algorithm:

initialize θ^0
for $t = 1, 2, \ldots,$ until convergence **do**
$q^t \leftarrow \text{argmax}_q \;\; G(\theta^{t-1}, q)$ // E-step
$\theta^t \leftarrow \text{argmax}_\theta \;\; G(\theta, q^t)$ // M-step
end for

Observation:

- both steps increase (or at least do not decrease) $G(\theta, q)$
- at convergence, we found a large value for $G(\theta, q)$, so $\log p(x;\theta)$ is also large

Derivation of the EM algorithm

a) $G(\theta, q)$ increases, but does $\mathcal{L}(\theta) = \log(x;\theta)$ also increase?

Derivation of the EM algorithm

a) $G(\theta, q)$ increases, but does $\mathcal{L}(\theta) = \log(x; \theta)$ also increase? **Yes!**

$$\mathcal{L}(\theta^t) \overset{q^t = p(z|x;\theta^t)}{=} G(\theta^t, q^t) \overset{\text{E-step}}{\leq} G(\theta^t, q^{t+1}) \overset{\text{M-step}}{\leq} G(\theta^{t+1}, q^{t+1}) \overset{\text{Jensen's ineq.}}{\leq} \mathcal{L}(\theta^{t+1})$$

Derivation of the EM algorithm

a) $G(\theta, q)$ increases, but does $\mathcal{L}(\theta) = \log(x; \theta)$ also increase? **Yes!**

$$\mathcal{L}(\theta^t) \overset{q^t = p(\underline{z}|x;\theta^t)}{=} G(\theta^t, q^t) \overset{\text{E-step}}{\leq} G(\theta^t, q^{t+1}) \overset{\text{M-step}}{\leq} G(\theta^{t+1}, q^{t+1}) \overset{\text{Jensen's ineq.}}{\leq} \mathcal{L}(\theta^{t+1})$$

b) When we reach a local optimum of $G(\theta, q)$, is this also a local optimum of $\log(x; \theta)$?

to do

Derivation of the EM algorithm for GMMs

Step 1: $q \leftarrow \text{argmax}_q \;\; G(\theta, q)$

- do the maths, or see from bound that $q(z) = p(z|x;\theta)$ is optimal choice

$$q(z) = p(z|x;\theta) = \prod_i p(h|v_i;\theta)$$

$$p(h = k|v = v_i) = \frac{\pi_k g_k(v_i)}{\sum_{k=1}^K \pi_k g_k(v_i)} = \gamma_{ik} \qquad \text{M-step}$$

Step 2: $\theta \leftarrow \text{argmax}_{\theta'} \;\; G(\theta', q)$

$$\begin{aligned}\underset{\theta'}{\text{argmax}}\, G(\theta', q) &= \underset{\theta'}{\text{argmax}}\, \mathbb{E}_{z\sim q} \log p(x, z; \theta) - \mathbb{E}_{z\sim q} \log q(z) \\ &= \underset{\theta'}{\text{argmax}} \sum_i \gamma_{ik} \log \pi_k g_k(v_i;\theta)\end{aligned}$$

Maximize the log-likelihood of Gaussians with γ_{ik}-weighted samples: E-step!

Variational Inference

Lower bound derivation of EM is example of a large class of variational algorithms:

- to handle a difficult distribution p, approximate it by a tractable distribution q (or a sequence of such distributions)
- typically, q is not arbitrary, but taken from a tractable parametric class, e.g.
  - Gaussian distributions
  - distributions that factorize: $q(z) = q(z_1) \dots q(z_n)$
  - ...
- if either step is hard, we don't have to solve it exactly, as long as $G(\theta, z)$ is improved

Variational Inference

Lower bound derivation of EM is example of a large class of variational algorithms:

- to handle a difficult distribution p, approximate it by a tractable distribution q (or a sequence of such distributions)
- typically, q is not arbitrary, but taken from a tractable parametric class, e.g.
  - Gaussian distributions
  - distributions that factorize: $q(z) = q(z_1) \dots q(z_n)$
  - ...
- if either step is hard, we don't have to solve it exactly, as long as $G(\theta, z)$ is improved

Currently very active area in machine learning, in particular for Bayesian handling of graphical models.

Further read: [Martin Wainwright, Michael Jordan. "Graphical Models, Exponential Families, and Variational Inference", Foundations and Trends in Machine Learning 2008]