Expectation maximization tutorial

Octavian Ganea

November 18, 2016

Today

- Expectation - maximization algorithm
- Topic modelling

ML & MAP

- Observed data: $X = \{x_1, x_2 \dots x_N\}$

ML & MAP

- Observed data: $X = \{x_1, x_2 \dots x_N\}$
- Probabilistic model of the data: $p(X|\theta) = \prod_{i=1}^{n} p(x_i|\theta)$

ML & MAP

- Observed data: $X = \{x_1, x_2 \dots x_N\}$
- Probabilistic model of the data: $p(X|\theta) = \prod_{i=1}^{n} p(x_i|\theta)$
- Estimate parameters:

ML & MAP

- Observed data: $X = \{x_1, x_2 \dots x_N\}$
- Probabilistic model of the data: $p(X|\theta) = \prod_{i=1}^{n} p(x_i|\theta)$
- Estimate parameters:
  - Maximum likelihood: $\hat{\theta}_{ML} = \arg\max_\theta p(X|\theta)$

ML & MAP

- Observed data: $X = \{x_1, x_2 \dots x_N\}$
- Probabilistic model of the data: $p(X|\theta) = \prod_{i=1}^{n} p(x_i|\theta)$
- Estimate parameters:
  - Maximum likelihood: $\hat{\theta}_{ML} = \arg\max_\theta p(X|\theta)$
  - Maximum a-posteriori:
    $\hat{\theta}_{MAP} = \arg\max_\theta p(\theta|X) = \arg\max_\theta [p(\theta) + p(X|\theta)]$

Maximizing the log-likelihood

- Observed data: $X = \{x_1, x_2 \ldots x_N\}$

Maximizing the log-likelihood

- Observed data: $X = \{x_1, x_2 \ldots x_N\}$
- Log-likelihood: $l(\theta) = \log p(X|\theta) = \sum_{i=1}^{N} \log p(x_i|\theta)$

Maximizing the log-likelihood

- Observed data: $X = \{x_1, x_2 \dots x_N\}$
- Log-likelihood: $l(\theta) = \log p(X|\theta) = \sum_{i=1}^{N} \log p(x_i|\theta)$
- Latent variables: $\log p(X|\theta) = \log \sum_Z p(X, Z|\theta)$

Maximizing the log-likelihood

- Observed data: $X = \{x_1, x_2 \dots x_N\}$
- Log-likelihood: $l(\theta) = \log p(X|\theta) = \sum_{i=1}^{N} \log p(x_i|\theta)$
- Latent variables: $\log p(X|\theta) = \log \sum_Z p(X, Z|\theta)$
- Hard to maximize $l(\theta)$ directly (no closed form solution in most of the interesting cases).

Maximizing the log-likelihood

- Observed data: $X = \{x_1, x_2 \dots x_N\}$
- Log-likelihood: $l(\theta) = \log p(X|\theta) = \sum_{i=1}^{N} \log p(x_i|\theta)$
- Latent variables: $\log p(X|\theta) = \log \sum_Z p(X, Z|\theta)$
- Hard to maximize $l(\theta)$ directly (no closed form solution in most of the interesting cases).
- One solution:

Maximizing the log-likelihood

- Observed data: $X = \{x_1, x_2 \dots x_N\}$
- Log-likelihood: $l(\theta) = \log p(X|\theta) = \sum_{i=1}^{N} \log p(x_i|\theta)$
- Latent variables: $\log p(X|\theta) = \log \sum_Z p(X, Z|\theta)$
- Hard to maximize $l(\theta)$ directly (no closed form solution in most of the interesting cases).
- One solution:
  - use a gradient method (e.g. gradient ascent, Newton)

Maximizing the log-likelihood

- Observed data: $X = \{x_1, x_2 \dots x_N\}$
- Log-likelihood: $l(\theta) = \log p(X|\theta) = \sum_{i=1}^{N} \log p(x_i|\theta)$
- Latent variables: $\log p(X|\theta) = \log \sum_Z p(X, Z|\theta)$
- Hard to maximize $l(\theta)$ directly (no closed form solution in most of the interesting cases).
- One solution:
  - use a gradient method (e.g. gradient ascent, Newton)
  - sometimes the gradient is hard to compute, hard to implement, or we do not want a black-box optimization routine with no guarantees

Expectation - maximization algorithm

- Used in models with latent variables.
- Iterative algorithm that guarantees convergence to stationary point of $l(\theta)$ (i.e. point with gradient zero, either local optimum or saddle point).
- No global optima guarantees. EM reaches either a local maximum or a saddle point
- Convergence speed might be slow.

Idea:

- Builds sequence: $l(\theta^{(0)}) \leq l(\theta^{(1)}) \leq \ldots \leq l(\theta^{(t)}) \leq \ldots$

Expectation - maximization algorithm

- Used in models with latent variables.
- Iterative algorithm that guarantees convergence to stationary point of $l(\theta)$ (i.e. point with gradient zero, either local optimum or saddle point).
- No global optima guarantees. EM reaches either a local maximum or a saddle point
- Convergence speed might be slow.

Idea:

- Builds sequence: $l(\theta^{(0)}) \leq l(\theta^{(1)}) \leq \ldots \leq l(\theta^{(t)}) \leq \ldots$
- At each step, using Jensen's inequality, finds a lower bound g s.t.

$$l(\theta^{(t)}) \leq g(\theta^{(t+1)}, q) \leq l(\theta^{(t+1)})$$

Expectation - maximization algorithm

For any probability distribution $q(Z)$ (s.t. $\sum_Z q(Z) = 1$), Jensen inequality gives a lower bound $F(q, \theta)$ on the true likelihood:

$$l(\theta) = \log\left(\sum_Z p(X, Z|\theta)\right) = \log\left(\sum_Z q(Z) \cdot \frac{p(X, Z|\theta)}{q(Z)}\right) \geq$$

$$\geq \sum_Z q(Z) \log\left(\cdot \frac{p(X, Z|\theta)}{q(Z)}\right) := F(q, \theta)$$

- Reason: $log(\cdot)$ is concave.
- Equality case: $q(Z) = p(Z|X, \theta)$.

Expectation - maximization algorithm

Update rule:

$$\theta^{(t+1)} = \arg\max_{\theta} g_t(\theta)$$

where

$$g_t(\theta) := F(p(Z|X,\theta^{(t)}),\theta) = \sum_Z p(Z|X,\theta^{(t)}) \log\left(\frac{p(X,Z|\theta)}{p(Z|X,\theta^{(t)})}\right)$$

- From above, $g_t(\theta) \leq l(\theta), \forall\theta$
  - in particular: $g_t(\theta^{(t+1)}) \leq l(\theta^{(t+1)})$
- Equality in Jensen: $g_t(\theta^{(t)}) = l(\theta^{(t)})$

So: $$l(\theta^{(t)}) = g_t(\theta^{(t)}) \leq g_t(\theta^{(t+1)}) \leq l(\theta^{(t+1)})$$

Expectation - maximization algorithm

EM algorithm:

- **E-step:** $q^{(t+1)} = \arg\max_q F(q, \theta^{(t)})$
  (i.e. $q^{(t+1)} = p(Z|X, \theta^{(t)})$)
- **M-step:** $\theta^{(t+1)} = \arg\max_\theta F(q^{(t+1)}, \theta)$

EM algorithm - convergence

- We proved so far that:

$$l(\theta^{(0)}) \leq l(\theta^{(1)}) \leq \ldots \leq l(\theta^{(t)}) \leq \ldots$$

- But why does it converge to a stationary point ? (Who guarantees no early stopping ?)

Proof:

- Let θ^* be the limit of the sequence defined by the EM algorithm.
- Then: $\theta^* = \arg\max_{\theta^*} g_*(\theta)$, where $g_*(\theta) = F(p(Z|X, \theta^*), \theta)$. This implies: $\boxed{\nabla_\theta g_*(\theta^*) = 0}$.
- Let $h_*(\theta) := l(\theta) - g_*(\theta) = -\sum_Z p(Z|X, \theta^*) \log\left(\frac{p(Z|X,\theta)}{p(Z|X,\theta^*)}\right)$
  - Then, $h_*(\theta) \geq 0, \forall\theta$ (since g_* is a lower bound of l)
  - and $h_*(\theta^*) = 0$ (Jensen equality case)
  - So, $\theta^* = \arg\min_\theta h_*(\theta) \quad \Rightarrow \boxed{\nabla_\theta h_*(\theta^*) = 0}$
- So, $\nabla_\theta l(\theta^*) = \nabla_\theta h_*(\theta^*) + \nabla_\theta g_*(\theta^*) = 0$, q.e.d.

EM Applications

- Tired of too much math ? :)
- Let's look at some cool applications of EM

Application 1 : Coin Flipping

- There are two coins A and B with θ_A and θ_B being the probability landing on Head when tossed.
- Do 5 rounds.
- In each round, select one coin uniformly at random and toss it 10 times then record the results.
- The observed data consists of 50 coin tosses.
- However, we don't know which coin was selected for a particular round.
- Estimate θ_A and θ_B.

Application 1 : Coin Flipping

Let's start simple:

- One coin A with $P(Y = H) = \theta_A$
- 10 tosses: $\#H = x \in \{0, \ldots, 10\}$, $\#T = 10 - x$
- How to estimate θ_A ? Maximize what we see !
- Mathematically, maximize data (log-)likelihood:
  - $\theta_A^* = \arg\max_{\theta_A} l(\theta_A)$, where $l(\theta_A) := \log P(X = x|\theta_A)$
  - $P(X = x|\theta_A) = \theta_A^x \cdot (1 - \theta_A)^{10-x}$ (note: fixed order of tosses)
  - $l(\theta_A) = x \log(\theta_A) + (10 - x) \log(1 - \theta_A)$
  - Set derivative to 0: $\frac{\partial l}{\partial \theta_A}(\theta_A^*) = 0 \iff \theta_A^* = \frac{x}{10}$
  - Best ML distribution is the empirical distribution.

Application 1 : Coin Flipping

Back to our original problem.

- Parameters $\theta = \{\theta_A, \theta_B\}$

Application 1 : Coin Flipping

Back to our original problem.

- Parameters $\theta = \{\theta_A, \theta_B\}$
- Latent r.v. Z_r - the coin selected in round $r \in \{1, \ldots, 5\}$: $p(Z_r = A) = p(Z_r = B) = 0.5$

Application 1 : Coin Flipping

Back to our original problem.

- Parameters $\theta = \{\theta_A, \theta_B\}$
- Latent r.v. Z_r - the coin selected in round $r \in \{1, \ldots, 5\}$: $p(Z_r = A) = p(Z_r = B) = 0.5$
- In each round r, the number of heads is x_r. Associated r.v. X_r.

Application 1 : Coin Flipping

Back to our original problem.

- Parameters $\theta = \{\theta_A, \theta_B\}$
- Latent r.v. Z_r - the coin selected in round $r \in \{1, \ldots, 5\}$: $p(Z_r = A) = p(Z_r = B) = 0.5$
- In each round r, the number of heads is x_r. Associated r.v. X_r.
- $p(X_r = x_r | Z_r = A; \theta) = \theta_A^{x_r}(1 - \theta_A)^{10 - x_r}$

Application 1 : Coin Flipping

Back to our original problem.

- Parameters $\theta = \{\theta_A, \theta_B\}$
- Latent r.v. Z_r - the coin selected in round $r \in \{1, \ldots, 5\}$: $p(Z_r = A) = p(Z_r = B) = 0.5$
- In each round r, the number of heads is x_r. Associated r.v. X_r.
- $p(X_r = x_r | Z_r = A; \theta) = \theta_A^{x_r}(1 - \theta_A)^{10-x_r}$
- Bayes rule: $p(Z_r = A | x_r; \theta) = \frac{\theta_A^{x_r}(1-\theta_A)^{10-x_r}}{\theta_A^{x_r}(1-\theta_A)^{10-x_r} + \theta_B^{x_r}(1-\theta_B)^{10-x_r}}$

Application 1 : Coin Flipping

- Data likelihood (per one round):

$$\begin{aligned} p(x_r;\theta) &= p(x_r|Z_r = A;\theta)p(Z_r = A) + p(x_r|Z_r = B;\theta)p(Z_r = B) \\ &= 0.5\left(\theta_A^{x_r}(1-\theta_A)^{10-x_r} + \theta_B^{x_r}(1-\theta_B)^{10-x_r}\right) \end{aligned}$$

Application 1 : Coin Flipping

- Data likelihood (per one round):

$$\begin{aligned} p(x_r;\theta) &= p(x_r|Z_r = A;\theta)p(Z_r = A) + p(x_r|Z_r = B;\theta)p(Z_r = B) \\ &= 0.5\left(\theta_A^{x_r}(1-\theta_A)^{10-x_r} + \theta_B^{x_r}(1-\theta_B)^{10-x_r}\right) \end{aligned}$$

- Data log-likelihood (all rounds):
$l(\theta) = \log p(X;\theta) = \sum_{r=1}^{5} \log p(x_r;\theta)$

Application 1 : Coin Flipping

- Data likelihood (per one round):

$$\begin{aligned} p(x_r;\theta) &= p(x_r|Z_r = A;\theta)p(Z_r = A) + p(x_r|Z_r = B;\theta)p(Z_r = B) \\ &= 0.5\left(\theta_A^{x_r}(1-\theta_A)^{10-x_r} + \theta_B^{x_r}(1-\theta_B)^{10-x_r}\right) \end{aligned}$$

- Data log-likelihood (all rounds):
  $l(\theta) = \log p(X;\theta) = \sum_{r=1}^{5} \log p(x_r;\theta)$

- Cannot maximize log-likelihood directly (i.e. by setting gradient to zero).

Application 1 : Coin Flipping

- Data likelihood (per one round):

$$\begin{aligned} p(x_r;\theta) &= p(x_r|Z_r = A;\theta)p(Z_r = A) + p(x_r|Z_r = B;\theta)p(Z_r = B) \\ &= 0.5\left(\theta_A^{x_r}(1-\theta_A)^{10-x_r} + \theta_B^{x_r}(1-\theta_B)^{10-x_r}\right) \end{aligned}$$

- Data log-likelihood (all rounds):
$l(\theta) = \log p(X;\theta) = \sum_{r=1}^{5} \log p(x_r;\theta)$

- Cannot maximize log-likelihood directly (i.e. by setting gradient to zero).

- Instead, maximize EM lower bound on $l(\theta)$ (formalized last time).

Application 1 : Coin Flipping

- EM lower-bound per round (Jensen inequality):

$$\log p(x_r;\theta) \geq \sum_{c=A,B} q_r(Z_r = c) \log \left(\frac{p(x_r, Z_r = c;\theta)}{q_r(Z_r = c)} \right) := F_r(q_r, \theta)$$

Application 1 : Coin Flipping

- EM lower-bound per round (Jensen inequality):

$$\log p(x_r;\theta) \geq \sum_{c=A,B} q_r(Z_r = c) \log \left(\frac{p(x_r, Z_r = c;\theta)}{q_r(Z_r = c)} \right) := F_r(q_r, \theta)$$

- Expectation step:
$q_r(Z_r = c) = p(Z_r = c|x_r;\theta^{(t)}), \quad \forall r \in \{1,\ldots,5\}$

Application 1 : Coin Flipping

- EM lower-bound per round (Jensen inequality):

$$\log p(x_r;\theta) \geq \sum_{c=A,B} q_r(Z_r = c) \log \left(\frac{p(x_r, Z_r = c;\theta)}{q_r(Z_r = c)} \right) := F_r(q_r, \theta)$$

- Expectation step:
  $q_r(Z_r = c) = p(Z_r = c|x_r;\theta^{(t)}), \quad \forall r \in \{1,\ldots,5\}$
- Maximization step:

$$\theta^{(t+1)} = \arg\max_{\theta} g_t(\theta)$$

$$\text{where} \quad g_t(\theta) = \sum_{r=1}^{5} F_r(p(Z_r = \cdot|x_r, \theta^{(t)}), \theta)$$

Application 1 : Coin Flipping

- Maximization step:

$$\theta^{(t+1)} = \arg\max_{\theta} \sum_{r=1}^{5} \sum_{c=A,B} p(Z_r = c | x_r, \theta^{(t)}) \log \left(p(x_r, Z_r = c; \theta) \right)$$

Application 1 : Coin Flipping

- Maximization step:

$$\theta^{(t+1)} = \arg\max_{\theta} \sum_{r=1}^{5} \sum_{c=A,B} p(Z_r = c|x_r, \theta^{(t)}) \log\left(p(x_r, Z_r = c; \theta)\right)$$

- Gradient:

$$\begin{aligned}\frac{\partial g_t(\theta)}{\partial \theta_A} &= \sum_{r=1}^{5} p(Z_r = A|x_r, \theta^{(t)}) \frac{\partial \log\left(p(x_r, Z_r = A; \theta)\right)}{\partial \theta_A} \\ &= \sum_{r=1}^{5} p(Z_r = A|x_r, \theta^{(t)}) \left(\frac{x_r}{\theta_A} + \frac{10 - x_r}{1 - \theta_A}\right)\end{aligned}$$

Application 1 : Coin Flipping

- Maximization step:

$$\theta^{(t+1)} = \arg\max_{\theta} \sum_{r=1}^{5} \sum_{c=A,B} p(Z_r = c|x_r, \theta^{(t)}) \log\left(p(x_r, Z_r = c; \theta)\right)$$

- Gradient:

$$\frac{\partial g_t(\theta)}{\partial \theta_A} = \sum_{r=1}^{5} p(Z_r = A|x_r, \theta^{(t)}) \frac{\partial \log\left(p(x_r, Z_r = A; \theta)\right)}{\partial \theta_A}$$

$$= \sum_{r=1}^{5} p(Z_r = A|x_r, \theta^{(t)}) \left(\frac{x_r}{\theta_A} + \frac{10 - x_r}{1 - \theta_A}\right)$$

- Gradient set to 0 gives: $\theta_A^{(t+1)} = \frac{\alpha_A^{(t)}}{\alpha_A^{(t)} + \beta_A^{(t)}}$ where

$$\alpha_A^{(t)} = \sum_{r=1}^{5} p(Z_r = A|x_r, \theta^{(t)}) x_r; \quad \beta_A^{(t)} = \sum_{r=1}^{5} p(Z_r = A|x_r, \theta^{(t)})(10 - x_r)$$

Application 1 : Coin Flipping

Final algorithm:

- Iteration: $t \leftarrow 0$
- Initialize parameters randomly: $\theta_A^{(0)}, \theta_B^{(0)} \in (0, 1)$
- Do until convergence:
  - $\theta_A^{(t+1)} = \frac{\alpha_A^{(t)}}{\alpha_A^{(t)} + \beta_A^{(t)}}$
  - $\theta_B^{(t+1)} = \frac{\alpha_B^{(t)}}{\alpha_B^{(t)} + \beta_B^{(t)}}$
  - $t \leftarrow t + 1$

Application 2 : Topic Modelling

- Document representations:
  - Used for classification, query retrieval, document similarity, etc.
  - A document can be seen as a multi-set of words $d = \{(w_i \rightarrow tf(w_i; d))\}_{i=1,|V|} \in \mathbb{R}^{|V|}$
  - Issues: high dimensionality, sparsity issues, potentially many infrequent words (with noisy estimated parameters)
  - Alternative (compressed topic representation):
    - topic distributions: $d = \{(t \rightarrow p(t|d))\}_{t=1,K} \in \mathbb{R}^K$
    - K = num of topics
    - $K << |V|$
- How to choose the number of topics K ?
  - Hyper-parameter: the one that gives the best performance on a validation set for the task at hand
  - Minimize perplexity of seen words

Application 2 : Topic Modelling

- Model parameters (to be learned): $\pi_t := p(t|d)$, $a_{nt} := p(w_n|t)$

Application 2 : Topic Modelling

- Model parameters (to be learned): $\pi_t := p(t|d)$, $a_{nt} := p(w_n|t)$
- Log likelihood (one document):

$$l(\pi) = \sum_{n=1}^{N} \log p(w_n|d) = \sum_{n=1}^{N} \log \sum_{t=1}^{T} \pi_t a_{nt}$$

Application 2 : Topic Modelling

- Model parameters (to be learned): $\pi_t := p(t|d)$, $a_{nt} := p(w_n|t)$
- Log likelihood (one document):

$$l(\pi) = \sum_{n=1}^{N} \log p(w_n|d) = \sum_{n=1}^{N} \log \sum_{t=1}^{T} \pi_t a_{nt}$$

- Iterative algorithm: keep a_{nt} fixed, learn π_t ; and reverse.

Application 2 : Topic Modelling

- Model parameters (to be learned): $\pi_t := p(t|d)$, $a_{nt} := p(w_n|t)$
- Log likelihood (one document):

$$l(\pi) = \sum_{n=1}^{N} \log p(w_n|d) = \sum_{n=1}^{N} \log \sum_{t=1}^{T} \pi_t a_{nt}$$

- Iterative algorithm: keep a_{nt} fixed, learn π_t ; and reverse.
- We do here just the update of π_t. The update of a_{nt} is similar.

Application 2 : Topic Modelling

- Model parameters (to be learned): $\pi_t := p(t|d)$, $a_{nt} := p(w_n|t)$
- Log likelihood (one document):

$$l(\pi) = \sum_{n=1}^{N} \log p(w_n|d) = \sum_{n=1}^{N} \log \sum_{t=1}^{T} \pi_t a_{nt}$$

- Iterative algorithm: keep a_{nt} fixed, learn π_t ; and reverse.
- We do here just the update of π_t. The update of a_{nt} is similar.
- Log-likelihood with Lagrange multipliers:

$$L(\pi, \lambda) = \sum_{n=1}^{N} \log \sum_{t=1}^{T} \pi_t a_{nt} - \lambda \left(\sum_{t=1}^{T} \pi_t - 1 \right)$$

Application 2 : Topic Modelling

- Iterative update algorithm.

Application 2 : Topic Modelling

- Iterative update algorithm.
- Latent variables Z are now the topics t.

Application 2 : Topic Modelling

- Iterative update algorithm.
- Latent variables Z are now the topics t.
- EM lower bound using Jensen:

$$L(\pi, \lambda) \geq F(q, \pi, \lambda) = \sum_{n=1}^{N} \sum_{t=1}^{T} q_{nt} \left[\log \frac{\pi_t}{q_{nt}} + \log a_{nt} \right] - \lambda \left(\sum_{t=1}^{T} \pi_t - 1 \right)$$

where $\sum_t q_{nt} = 1, \forall n$

Application 2 : Topic Modelling

- Iterative update algorithm.
- Latent variables Z are now the topics t.
- EM lower bound using Jensen:

$$L(\pi, \lambda) \geq F(q, \pi, \lambda) = \sum_{n=1}^{N} \sum_{t=1}^{T} q_{nt} \left[\log \frac{\pi_t}{q_{nt}} + \log a_{nt} \right] - \lambda \left(\sum_{t=1}^{T} \pi_t - 1 \right)$$

  where $\sum_t q_{nt} = 1, \forall n$

- E-step, iteration k: $q_{nt}^{(k)} = \frac{\pi_t^{(k)} a_{nt}}{\sum_{t'} \pi_{t'}^{(k)} a_{nt'}}$

Application 2 : Topic Modelling

- Iterative update algorithm.
- Latent variables Z are now the topics t.
- EM lower bound using Jensen:

$$L(\pi, \lambda) \geq F(q, \pi, \lambda) = \sum_{n=1}^{N} \sum_{t=1}^{T} q_{nt} \left[\log \frac{\pi_t}{q_{nt}} + \log a_{nt} \right] - \lambda \left(\sum_{t=1}^{T} \pi_t - 1 \right)$$

  where $\sum_t q_{nt} = 1, \forall n$

- E-step, iteration k: $q_{nt}^{(k)} = \frac{\pi_t^{(k)} a_{nt}}{\sum_{t'} \pi_{t'}^{(k)} a_{nt'}}$
- M-step, iteration k:

$$\pi_t^{(k+1)} = \frac{\pi_t^{(k)}}{N} \sum_{n=1}^{N} \frac{a_{nt}}{\sum_{t'} \pi_{t'}^{(k)} a_{nt'}}$$

Questions?