

Alternating Maximization: An Unified View of EM and Blahut-Arimoto Algorithms

Jie Wang
Fourth Year Undergraduate Student
Pure Mathematics Major
Network Coding Lab

September 16, 2019

Table of Content

- 1 Introduction to Alternating Optimization
- 2 Introduction to EM Algorithm
- 3 Incremental Variants of EM (Alternating Maximization)
- 4 Introduction to Channel Capacity
- 5 Blahut-Arimoto Algorithm (Alternating Maximization)
- 6 Variants of Blahut-Arimoto Algorithm (Proximal Point Algorithm)

Table of Contents

- 1 Introduction to Alternating Optimization
- 2 Introduction to EM Algorithm
- 3 Incremental Variants of EM (Alternating Maximization)
- 4 Introduction to Channel Capacity
- 5 Blahut-Arimoto Algorithm (Alternating Maximization)
- 6 Variants of Blahut-Arimoto Algorithm (Proximal Point Algorithm)

Motivation

- Why Alternating Optimization?

$\max_x f(x, y)$ is easy for fixed y

$\max_y f(x, y)$ is easy for fixed x

- Alternating Optimization Framework:

$$x_+ = \arg \max_x f(x, y)$$

$$y_+ = \arg \max_y f(x_+, y)$$

- Comment:

- ▶ “big” global steps, and much faster than “local” gradient descent
- ▶ No step size to be tuned / chosen

Motivation

- Why Alternating Optimization?

$\max_x f(x, y)$ is easy for fixed y

$\max_y f(x, y)$ is easy for fixed x

- Alternating Optimization Framework:

$$x_+ = \arg \max_x f(x, y)$$

$$y_+ = \arg \max_y f(x_+, y)$$

- Comment:

- ▶ “big” global steps, and much faster than “local” gradient descent
- ▶ No step size to be tuned / chosen

Motivation

- Why Alternating Optimization?

$\max_x f(x, y)$ is easy for fixed y

$\max_y f(x, y)$ is easy for fixed x

- Alternating Optimization Framework:

$$x_+ = \arg \max_x f(x, y)$$

$$y_+ = \arg \max_y f(x_+, y)$$

- Comment:

- ▶ “big” global steps, and much faster than “local” gradient descent
- ▶ No step size to be tuned / chosen

Problem Setting

- Consider optimization for groups of variables:

$$\max_{x \in \mathcal{X}} \max_{y \in \mathcal{Y}} f(x, y)$$

where \mathcal{X}, \mathcal{Y} are *convex*, and the objective function f is such that:

- ▶ $f : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$ is bounded above;
- ▶ f is continuous and has continuous partial derivatives on $\mathcal{X} \times \mathcal{Y}$;
- ▶ For any fixed $y \in \mathcal{Y}$, there exists a unique $c_1(y) \in \mathcal{X}$ such that

$$f(c_1(y), y) = \max_{x \in \mathcal{X}} f(x, y)$$

For any fixed $x \in \mathcal{X}$, there exists a unique $c_2(x) \in \mathcal{Y}$ such that

$$f(x, c_2(x)) = \max_{y \in \mathcal{Y}} f(x, y)$$

Problem Setting

- Consider optimization for groups of variables:

$$\max_{x \in \mathcal{X}} \max_{y \in \mathcal{Y}} f(x, y)$$

where \mathcal{X}, \mathcal{Y} are *convex*, and the objective function f is such that:

- ▶ $f : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$ is bounded above;
- ▶ f is continuous and has continuous partial derivatives on $\mathcal{X} \times \mathcal{Y}$;
- ▶ For any fixed $y \in \mathcal{Y}$, there exists a unique $c_1(y) \in \mathcal{X}$ such that

$$f(c_1(y), y) = \max_{x \in \mathcal{X}} f(x, y)$$

For any fixed $x \in \mathcal{X}$, there exists a unique $c_2(x) \in \mathcal{Y}$ such that

$$f(x, c_2(x)) = \max_{y \in \mathcal{Y}} f(x, y)$$

Problem Setting

- Consider optimization for groups of variables:

$$\max_{x \in \mathcal{X}} \max_{y \in \mathcal{Y}} f(x, y)$$

where \mathcal{X}, \mathcal{Y} are *convex*, and the objective function f is such that:

- ▶ $f : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$ is bounded above;
- ▶ f is continuous and has continuous partial derivatives on $\mathcal{X} \times \mathcal{Y}$;
- ▶ For any fixed $y \in \mathcal{Y}$, there exists a unique $c_1(y) \in \mathcal{X}$ such that

$$f(c_1(y), y) = \max_{x \in \mathcal{X}} f(x, y)$$

For any fixed $x \in \mathcal{X}$, there exists a unique $c_2(x) \in \mathcal{Y}$ such that

$$f(x, c_2(x)) = \max_{y \in \mathcal{Y}} f(x, y)$$

Alternating Optimization (AO) and its Variants

Algorithm 1 Alternating Optimization

Require: Objective function $f : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$;

Ensure: A near-optimal point $(\hat{x}, \hat{y}) \in \mathcal{X} \times \mathcal{Y}$;

- 1: $(x^1, y^1) \leftarrow \text{INITIALIZE}()$;
 - 2: **for** $t = 1, 2, \dots, T$ **do**
 - 3: $x^{t+1} \leftarrow \arg \max_{x \in \mathcal{X}} f(x, y^t)$;
 - 4: $y^{t+1} \leftarrow \arg \max_{y \in \mathcal{Y}} f(x^{t+1}, y)$;
 - 5: **end for**
 - 6: **return** (x^T, y^T)
-

Remarks for Alternating Optimization

- Alternating Optimization Algorithm converges to *Bistable Point*.

Definition (Bistable Point)

A point $(x^*, y^*) \in \mathcal{X} \times \mathcal{Y}$ is said to be a *bistable point* if

$$y^* = \arg \max_{y \in \mathcal{Y}} f(x^*, y), \quad \text{and} \quad x^* = \arg \max_{x \in \mathcal{X}} f(x, y^*)$$

- All bistable points are *globally optimal* for convex problems.
For marginally-convex problems, bi-stable points are equivalent to FOSP.
- Alternating Optimization has sublinear convergence rate.

Remarks for Alternating Optimization

- Alternating Optimization Algorithm converges to *Bistable Point*.

Definition (Bistable Point)

A point $(x^*, y^*) \in \mathcal{X} \times \mathcal{Y}$ is said to be a *bistable point* if

$$y^* = \arg \max_{y \in \mathcal{Y}} f(x^*, y), \quad \text{and} \quad x^* = \arg \max_{x \in \mathcal{X}} f(x, y^*)$$

- All bistable points are *globally optimal* for convex problems.
For marginally-convex problems, bi-stable points are equivalent to FOSP.
- Alternating Optimization has sublinear convergence rate.

Remarks for Alternating Optimization

- Alternating Optimization Algorithm converges to *Bistable Point*.

Definition (Bistable Point)

A point $(x^*, y^*) \in \mathcal{X} \times \mathcal{Y}$ is said to be a *bistable point* if

$$y^* = \arg \max_{y \in \mathcal{Y}} f(x^*, y), \quad \text{and} \quad x^* = \arg \max_{x \in \mathcal{X}} f(x, y^*)$$

- All bistable points are *globally optimal* for convex problems.
For marginally-convex problems, bi-stable points are equivalent to FOSP.
- Alternating Optimization has sublinear convergence rate.

Concluding Remarks

- 1 Successful for a collection of machine learning problems:
 - ▶ Phase retrieval
 - ▶ Matrix sensing
 - ▶ Robust PCA
 - ▶ Matrix Completion
 - ▶ Mixed linear regression
- 2 Special initialization is designed, such as *spectral initialization*
- 3 Empirically: memory and fast
- 4 **Open Problem:** general theory about convergence ...

Table of Contents

- 1 Introduction to Alternating Optimization
- 2 Introduction to EM Algorithm**
- 3 Incremental Variants of EM (Alternating Maximization)
- 4 Introduction to Channel Capacity
- 5 Blahut-Arimoto Algorithm (Alternating Maximization)
- 6 Variants of Blahut-Arimoto Algorithm (Proximal Point Algorithm)

Problem Setting

- An statistical model \mathcal{F} is generating $X \in \mathcal{X}, Z \in \mathcal{Z}$:

$$\mathcal{F} = \{f_{\theta} = p(\cdot, \cdot | \theta) : \theta \in \Theta\}$$

- f_{θ^*} generates sample pair $(x_i, z_i)_{i=1}^n$, but only $\{x_i\}_{i=1}^n$ is observed.
- The goal is to recover θ^* , by using only samples $\{x_i\}_{i=1}^n$.

Likelihood maximization:

$$\mathcal{L}(\theta; x_1, \dots, x_n) \triangleq p(x_1, \dots, x_n | \theta) = \prod_{i=1}^n \sum_{z_i \in \mathcal{Z}} p(x_i, z_i | \theta)$$

$$\hat{\theta}_{\text{MLE}} = \arg \max_{\theta \in \Theta} \log \mathcal{L}(\theta; x_1, \dots, x_n) \triangleq \sum_{i=1}^n \log \mathcal{L}(\theta; x_i)$$

- Intractability.

Problem Setting

- An statistical model \mathcal{F} is generating $X \in \mathcal{X}, Z \in \mathcal{Z}$:

$$\mathcal{F} = \{f_{\theta} = p(\cdot, \cdot | \theta) : \theta \in \Theta\}$$

- f_{θ^*} generates sample pair $(x_i, z_i)_{i=1}^n$, but only $\{x_i\}_{i=1}^n$ is observed.
- The goal is to recover θ^* , by using only samples $\{x_i\}_{i=1}^n$.

Likelihood maximization:

$$\mathcal{L}(\theta; x_1, \dots, x_n) \triangleq p(x_1, \dots, x_n | \theta) = \prod_{i=1}^n \sum_{z_i \in \mathcal{Z}} p(x_i, z_i | \theta)$$

$$\hat{\theta}_{\text{MLE}} = \arg \max_{\theta \in \Theta} \log \mathcal{L}(\theta; x_1, \dots, x_n) \triangleq \sum_{i=1}^n \log \mathcal{L}(\theta; x_i)$$

- Intractability.

Problem Setting

- An statistical model \mathcal{F} is generating $X \in \mathcal{X}, Z \in \mathcal{Z}$:

$$\mathcal{F} = \{f_{\theta} = p(\cdot, \cdot | \theta) : \theta \in \Theta\}$$

- f_{θ^*} generates sample pair $(x_i, z_i)_{i=1}^n$, but only $\{x_i\}_{i=1}^n$ is observed.
- The goal is to recover θ^* , by using only samples $\{x_i\}_{i=1}^n$.

Likelihood maximization:

$$\mathcal{L}(\theta; x_1, \dots, x_n) \triangleq p(x_1, \dots, x_n | \theta) = \prod_{i=1}^n \sum_{z_i \in \mathcal{Z}} p(x_i, z_i | \theta)$$

$$\hat{\theta}_{\text{MLE}} = \arg \max_{\theta \in \Theta} \log \mathcal{L}(\theta; x_1, \dots, x_n) \triangleq \sum_{i=1}^n \log \mathcal{L}(\theta; x_i)$$

- Intractability.

Problem Setting

- An statistical model \mathcal{F} is generating $X \in \mathcal{X}, Z \in \mathcal{Z}$:

$$\mathcal{F} = \{f_{\theta} = p(\cdot, \cdot | \theta) : \theta \in \Theta\}$$

- f_{θ^*} generates sample pair $(x_i, z_i)_{i=1}^n$, but only $\{x_i\}_{i=1}^n$ is observed.
- The goal is to recover θ^* , by using only samples $\{x_i\}_{i=1}^n$.

Likelihood maximization:

$$\mathcal{L}(\theta; x_1, \dots, x_n) \triangleq p(x_1, \dots, x_n | \theta) = \prod_{i=1}^n \sum_{z_i \in \mathcal{Z}} p(x_i, z_i | \theta)$$

$$\hat{\theta}_{\text{MLE}} = \arg \max_{\theta \in \Theta} \log \mathcal{L}(\theta; x_1, \dots, x_n) \triangleq \sum_{i=1}^n \log \mathcal{L}(\theta; x_i)$$

- Intractability.

Problem Setting

- An statistical model \mathcal{F} is generating $X \in \mathcal{X}, Z \in \mathcal{Z}$:

$$\mathcal{F} = \{f_\theta = p(\cdot, \cdot \mid \theta) : \theta \in \Theta\}$$

- f_{θ^*} generates sample pair $(x_i, z_i)_{i=1}^n$, but only $\{x_i\}_{i=1}^n$ is observed.
- The goal is to recover θ^* , by using only samples $\{x_i\}_{i=1}^n$.

Likelihood maximization:

$$\mathcal{L}(\theta; x_1, \dots, x_n) \triangleq p(x_1, \dots, x_n \mid \theta) = \prod_{i=1}^n \sum_{z_i \in \mathcal{Z}} p(x_i, z_i \mid \theta)$$

$$\hat{\theta}_{\text{MLE}} = \arg \max_{\theta \in \Theta} \log \mathcal{L}(\theta; x_1, \dots, x_n) \triangleq \sum_{i=1}^n \log \mathcal{L}(\theta; x_i)$$

- Intractability.

Description of EM Algorithm

- The likelihood maximization is to solve

$$\hat{\theta}_{\text{MLE}} = \arg \max_{\theta \in \Theta} \log \mathcal{L}(\theta; x_1, \dots, x_n) \triangleq \sum_{i=1}^n \log \underbrace{\sum_{z_i \in \mathcal{Z}} p(x_i, z_i \mid \theta)}_{\log \mathcal{L}(\theta, x_i)}$$

- The EM Algorithm is separated into two steps:

- 1 E-step: Construct the Q -function $Q(\theta \mid \theta^t)$;
(A lower bound of the objective function for current iteration)
- 2 M-step: Maximize the Q -function such that

$$\theta^{t+1} \leftarrow \arg \max Q(\theta \mid \theta^t).$$

Description of EM Algorithm

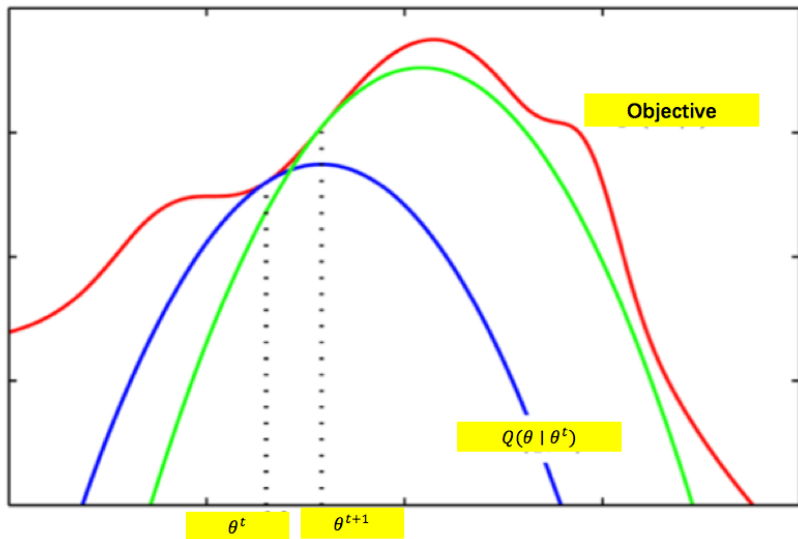
- The likelihood maximization is to solve

$$\hat{\theta}_{\text{MLE}} = \arg \max_{\theta \in \Theta} \log \mathcal{L}(\theta; x_1, \dots, x_n) \triangleq \sum_{i=1}^n \underbrace{\log \sum_{z_i \in \mathcal{Z}} p(x_i, z_i \mid \theta)}_{\log \mathcal{L}(\theta, x_i)}$$

- The EM Algorithm is separated into two steps:
 - 1 E-step: Construct the Q -function $Q(\theta \mid \theta^t)$;
(A lower bound of the objective function for current iteration)
 - 2 M-step: Maximize the Q -function such that

$$\theta^{t+1} \leftarrow \arg \max Q(\theta \mid \theta^t).$$

EM is a successive approximation method



EM Algorithm in Detail

Key inequality:

$$\log \mathcal{L}(\theta; x_i) \triangleq \log \sum_{z_i \in \mathcal{Z}} p(x_i, z_i | \theta) \geq Q_{x_i}(\theta | \theta^t) \triangleq \mathbb{E}_{z \sim p(\cdot | x_i, \theta^t)} [\log p(x_i, z | \theta)]$$

$$\text{Objective} \triangleq \sum_{i=1}^n \log \mathcal{L}(\theta; x_i) \geq Q(\theta | \theta^t) \triangleq \sum_{i=1}^n Q_{x_i}(\theta | \theta^t)$$

EM Algorithm Description

- E-step: Compute the distribution $p(Z | x_i, \theta^t)$.
(Then the Q -function can be constructed)
- M-Step: Choose θ^{t+1} maximizing the objective function above.

EM Algorithm in Detail

Key inequality:

$$\log \mathcal{L}(\theta; x_i) \triangleq \log \sum_{z_i \in \mathcal{Z}} p(x_i, z_i | \theta) \geq Q_{x_i}(\theta | \theta^t) \triangleq \mathbb{E}_{z \sim p(\cdot | x_i, \theta^t)} [\log p(x_i, z | \theta)]$$

$$\text{Objective} \triangleq \sum_{i=1}^n \log \mathcal{L}(\theta; x_i) \geq Q(\theta | \theta^t) \triangleq \sum_{i=1}^n Q_{x_i}(\theta | \theta^t)$$

EM Algorithm Description

- E-step: Compute the distribution $p(Z | x_i, \theta^t)$.
(Then the Q -function can be constructed)
- M-Step: Choose θ^{t+1} maximizing the objective function above.

EM as maximization-maximization

Consider the following function:

$$F(\theta, q) = -D(q||q_\theta) + \mathcal{L}(\theta; x_{1:n}), \quad q_\theta(z) = p(z | x_{1:n}, \theta)$$

Or equivalently,

$$F(\theta, q) \equiv \mathbb{E}_q[\log p(z, x_{1:n} | \theta)] + H(q), \quad q \text{ is some distribution of } z.$$

Theorem (Alternative Maximization of F reduces to EM)

- 1 For fixed θ , the unique maximizer of $F(\theta, q)$ is given by

$$q = q_\theta \triangleq p(z | x_{1:n}, \theta).$$

- 2 For fixed $q = q_\theta$, the M-step is equivalent to maximizing $F(\theta, q_\theta)$.

EM as maximization-maximization

Consider the following function:

$$F(\theta, q) = -D(q||q_\theta) + \mathcal{L}(\theta; x_{1:n}), \quad q_\theta(z) = p(z | x_{1:n}, \theta)$$

Or equivalently,

$$F(\theta, q) \equiv \mathbb{E}_q[\log p(z, x_{1:n} | \theta)] + H(q), \quad q \text{ is some distribution of } z.$$

Theorem (Alternative Maximization of F reduces to EM)

- 1 For fixed θ , the unique maximizer of $F(\theta, q)$ is given by

$$q = q_\theta \triangleq p(z | x_{1:n}, \theta).$$

- 2 For fixed $q = q_\theta$, the M-step is equivalent to maximizing $F(\theta, q_\theta)$.

Landscape of Alternating Maximization

Alternating Maximization & EM Algorithm

Suppose that the function $F(\theta, q)$ has a local maxima (θ^*, q^*) :

$$\begin{aligned}(\theta^*, q^*) &= \arg \max F(\theta, q) \\ &:= \arg \max \mathbb{E}_q[\log p(z, x_{1:n} \mid \theta)] + H(q) \\ &:= \arg \max -D(q \parallel q_\theta) + \mathcal{L}(\theta; x_{1:n}).\end{aligned}$$

Then the point θ^* is a local maxima of objective function $\mathcal{L}(\theta; x_{1:n})$:

$$\theta^* = \arg \max \mathcal{L}(\theta; x_1, \dots, x_n) \triangleq p(x_1, \dots, x_n \mid \theta).$$

Proof.

$$\text{local maxima } (\theta^*, q^*) \implies q^* = q_\theta \implies \theta^* = \arg \max \mathcal{L}(\theta; x_{1:n})$$



Landscape of Alternating Maximization

Alternating Maximization & EM Algorithm

Suppose that the function $F(\theta, q)$ has a local maxima (θ^*, q^*) :

$$\begin{aligned}(\theta^*, q^*) &= \arg \max F(\theta, q) \\ &:= \arg \max \mathbb{E}_q[\log p(z, x_{1:n} \mid \theta)] + H(q) \\ &:= \arg \max -D(q \parallel q_\theta) + \mathcal{L}(\theta; x_{1:n}).\end{aligned}$$

Then the point θ^* is a local maxima of objective function $\mathcal{L}(\theta; x_{1:n})$:

$$\theta^* = \arg \max \mathcal{L}(\theta; x_1, \dots, x_n) \triangleq p(x_1, \dots, x_n \mid \theta).$$

Proof.

$$\text{local maxima } (\theta^*, q^*) \implies q^* = q_\theta \implies \theta^* = \arg \max \mathcal{L}(\theta; x_{1:n})$$



EM Algorithm as Alternating Maximization

Algorithm 2 Expectation Maximization as Alternating Maximization

- 1: **Input:** Solvers of $\max_{\theta} F(\theta, q)$ for fixed q and $\max_q F(\theta, q)$ for fixed θ ;
 - 2: **Output:** A good parameter $\hat{\theta} \in \Theta$;
 - 3: $\theta^1 \leftarrow \text{INITIALIZE}()$;
 - 4: **for** $t = 1, 2, \dots$ **do**
 - 5: $q^t \leftarrow \arg \max_q F(\theta^{t-1}, q)$;
 - 6: $\theta^t \leftarrow \arg \max_{\theta} F(\theta, q^t)$;
 - 7: **end for**
-

Comment:

- ① The Alternating Maximization blindly searches for FOSP.
- ② Pay attention to Initialization, unless for unique bi-stable point;

EM Algorithm as Alternating Maximization

Algorithm 3 Expectation Maximization as Alternating Maximization

```

1: Input: Solvers of  $\max_{\theta} F(\theta, q)$  for fixed  $q$  and  $\max_q F(\theta, q)$  for fixed  $\theta$ ;
2: Output: A good parameter  $\hat{\theta} \in \Theta$ ;
3:  $\theta^1 \leftarrow \text{INITIALIZE}()$ ;
4: for  $t = 1, 2, \dots$  do
5:    $q^t \leftarrow \arg \max_q F(\theta^{t-1}, q)$ ;
6:    $\theta^t \leftarrow \arg \max_{\theta} F(\theta, q^t)$ ;
7: end for
  
```

Comment:

- ① The Alternating Maximization blindly searches for FOSP.
- ② Pay attention to Initialization, unless for unique bi-stable point;

Table of Contents

- 1 Introduction to Alternating Optimization
- 2 Introduction to EM Algorithm
- 3 Incremental Variants of EM (Alternating Maximization)**
- 4 Introduction to Channel Capacity
- 5 Blahut-Arimoto Algorithm (Alternating Maximization)
- 6 Variants of Blahut-Arimoto Algorithm (Proximal Point Algorithm)

Incremental Variants of EM Algorithm (IVEM)

Key Observation:

$$\begin{aligned}
 F(\theta, q) &= \mathbb{E}_q[\log p(z, x_{1:n} \mid \theta)] + H(q) \\
 &= \sum_{i=1}^n \mathbb{E}_q[\log p(z_i, x_i \mid \theta)] + H(q_i) \\
 &\triangleq \sum_{i=1}^n F_i(\theta, q_i),
 \end{aligned}$$

Incremental Variants of EM Algorithm (IVEM)

$$F(\theta, q) = \sum_{i=1}^n F_i(\theta, q_i)$$

Perform the maximization of $q := (q_i)_{i=1}^n$ *lazily*:

Algorithm 4 An Incremental Variant of EM Algorithm

- 1: **Input:** Solvers of $\max F_i(\theta, q)$ for fixed θ ;
 - 2: **Output:** A good parameter $\hat{\theta} \in \Theta$;
 - 3: $\theta^1 \leftarrow \text{INITIALIZE}()$;
 - 4: **for** $t = 1, 2, \dots$ **do**
 - 5: Pick Index i_t from $\{1, \dots, n\}$;
 - 6: $q_{i_t}^t \leftarrow \arg \max F_{i_t}(\theta^{t-1}, q_{i_t})$;
 - 7: $q_j^t \leftarrow q_j^{t-1}$ for any $j \neq i_t$;
 - 8: $\theta^t \leftarrow \arg \max F(\theta, q^t) = \sum_{i=1}^n F_i(\theta, q_i^t)$ (Not Memory Efficient);
 - 9: **end for**
-

Finite-Sum Problem Setting

Consider the optimization for a finite-sum objective:

$$\max_x \sum_{i=1}^n f_i(x)$$

Full GD : $x^{t+1} = x^t + \eta_t \frac{1}{n} \sum_i \nabla f_i(x^t)$ $\mathcal{O}(n)$ computations per iteration

SGD : $x^{t+1} = x^t + \eta_t \nabla f_{i_t}(x^t)$ $\mathcal{O}(1)$ computation per iteration
(but many more iterations)

Stochastic Average Gradient (SAG)

- At each iteration t , update the gradient estimate lazily:
 - Pick Index i_t from $\{1, \dots, n\}$,

$$g_{i_t}^t = \nabla f_{i_t}(x^{t-1})$$

$$g_j^t = g_j^{t-1}, \quad \text{for any } j \neq i_t$$

- The gradient estimate for objective $\sum_{i=1}^n f_i(x)$ is given by

$$\sum_{i=1}^n g_i^t$$

- At each iteration t perform the update

$$x^t = x^{t-1} + \eta_t \sum_{i=1}^n g_i^t$$

Stochastic Average Gradient (SAG)

- At each iteration t , update the gradient estimate lazily:
 - Pick Index i_t from $\{1, \dots, n\}$,

$$g_{i_t}^t = \nabla f_{i_t}(x^{t-1})$$

$$g_j^t = g_j^{t-1}, \quad \text{for any } j \neq i_t$$

- The gradient estimate for objective $\sum_{i=1}^n f_i(x)$ is given by

$$\sum_{i=1}^n g_i^t$$

- At each iteration t perform the update

$$x^t = x^{t-1} + \eta_t \sum_{i=1}^n g_i^t$$

Memoery Efficient Implementation of SAG

The gradient update of SAG only takes *contant* time, independ of n :

$$\sum_{i=1}^n g_i^t = g_{i_t}^t - g_{i_t}^{t-1} + \sum_{i=1}^n g_i^{t-1}$$

$$a^t = g_{i_t}^t - g_{i_t}^{t-1} + a^{t-1}$$

At each iteration t perform the update

$$x^t = x^{t-1} + \eta_t a^t$$

Memoery Efficient Implementation of SAG

The gradient update of SAG only takes *contant* time, independ of n :

$$\sum_{i=1}^n g_i^t = g_{i_t}^t - g_{i_t}^{t-1} + \sum_{i=1}^n g_i^{t-1}$$

$$a^t = g_{i_t}^t - g_{i_t}^{t-1} + a^{t-1}$$

At each iteration t perform the update

$$x^t = x^{t-1} + \eta_t a^t$$

Memoery Efficient Implementation of IVEM

$$\max \sum_{i=1}^n F_i(\theta, q_i^t) = \max \sum_{i=1}^n F_i(\theta, q_i^{t-1}) - F_{i_t}(\theta, q_{i_t}^{t-1}) + F_{i_t}(\theta, q_{i_t}^t)$$

Algorithm 5 Another Incremental Variant of EM Algorithm

- 1: **Input:** Solvers of $\max F_i(\theta, q)$ for fixed θ ;
 - 2: **Output:** A good parameter $\hat{\theta} \in \Theta$;
 - 3: $\theta^1 \leftarrow \text{INITIALIZE}()$;
 - 4: **for** $t = 1, 2, \dots$ **do**
 - 5: Pick Index i_t from $\{1, \dots, n\}$;
 - 6: $q_{i_t}^t \leftarrow \arg \max F_{i_t}(\theta^{t-1}, q_{i_t})$;
 - 7: $\theta^t \leftarrow \arg \max F(\theta, q^{t-1}) - F_{i_t}(\theta, q_{i_t}^{t-1}) + F_{i_t}(\theta, q_{i_t}^t)$;
 - 8: **end for**
-

Exponentially decaying variants of IVEM

$$\max \sum_{i=1}^n F_i(\theta, q_i^t) \approx \max \gamma \sum_{i=1}^n F_i(\theta, q_i^{t-1}) + F_{i_t}(\theta, q_{i_t}^t)$$

Algorithm 6 Exponentially decaying version of IVEM

- 1: **Input:** Solvers of $\max F_i(\theta, q)$ for fixed θ ;
 - 2: **Output:** A good parameter $\hat{\theta} \in \Theta$;
 - 3: $\theta^1 \leftarrow \text{INITIALIZE}()$;
 - 4: **for** $t = 1, 2, \dots$ **do**
 - 5: Pick Index i_t from $\{1, \dots, n\}$;
 - 6: $q_{i_t}^t \leftarrow \arg \max F_{i_t}(\theta^{t-1}, q_{i_t})$;
 - 7: $\theta^t \leftarrow \arg \max \gamma F(\theta, q^{t-1}) + F_{i_t}(\theta, q_{i_t}^t)$;
 - 8: **end for**
-

Table of Contents

- 1 Introduction to Alternating Optimization
- 2 Introduction to EM Algorithm
- 3 Incremental Variants of EM (Alternating Maximization)
- 4 Introduction to Channel Capacity**
- 5 Blahut-Arimoto Algorithm (Alternating Maximization)
- 6 Variants of Blahut-Arimoto Algorithm (Proximal Point Algorithm)

Preliminary Definitions

- The KL-Divergence between two distributions \mathbf{p} and \mathbf{q} is given by:

$$D(\mathbf{p} \parallel \mathbf{q}) = \sum_{i=1}^n p_i \log \frac{p_i}{q_i}$$

where \mathbf{p} and \mathbf{q} are assumed to have the same support set $\{1, \dots, n\}$

- The mutual information between a pair of random variables (X, Y) is:

$$I(X; Y) = D(p_{X,Y} \parallel p_X \otimes p_Y)$$

- Suppose that the joint distribution of (X, Y) is $p_X \cdot p_{Y|X}$. For fixed conditional distribution $p_{Y|X}$, the channel capacity of (X, Y) is defined as

$$C = \max_{p_X} I(X; Y)$$

Preliminary Definitions

- The KL-Divergence between two distributions p and q is given by:

$$D(p\|q) = \sum_{i=1}^n p_i \log \frac{p_i}{q_i}$$

where p and q are assumed to have the same support set $\{1, \dots, n\}$

- The mutual information between a pair of random variables (X, Y) is:

$$I(X; Y) = D(p_{X,Y} \| p_X \otimes p_Y)$$

- Suppose that the joint distribution of (X, Y) is $p_X \cdot p_{Y|X}$. For fixed conditional distribution $p_{Y|X}$, the channel capacity of (X, Y) is defined as

$$C = \max_{p_X} I(X; Y)$$

Preliminary Definitions

- The KL-Divergence between two distributions \mathbf{p} and \mathbf{q} is given by:

$$D(\mathbf{p} \parallel \mathbf{q}) = \sum_{i=1}^n p_i \log \frac{p_i}{q_i}$$

where \mathbf{p} and \mathbf{q} are assumed to have the same support set $\{1, \dots, n\}$

- The mutual information between a pair of random variables (X, Y) is:

$$I(X; Y) = D(p_{X,Y} \parallel p_X \otimes p_Y)$$

- Suppose that the joint distribution of (X, Y) is $p_X \cdot p_{Y|X}$. For fixed conditional distribution $p_{Y|X}$, the channel capacity of (X, Y) is defined as

$$C = \max_{p_X} I(X; Y)$$

Variantional Characterization of Mutual Information

- Suppose that $X \sim p$ and $Y | X \sim Q$, then

$$I(X; Y) \triangleq I(p, Q) = \sum_{i=1}^{|\mathcal{X}|} \sum_{j=1}^{|\mathcal{Y}|} p_i Q_{i,j} \log \frac{Q_{i,j}}{q_j}$$

where $Y \sim q \triangleq p \cdot Q$.

Theorem (Global Tight Lower Bound of Mutual Information)

$$I(p, Q) \geq \tilde{I}(p, Q; \phi) \triangleq \sum_{i=1}^{|\mathcal{X}|} \sum_{j=1}^{|\mathcal{Y}|} p_i Q_{i,j} \log \frac{\phi(i | j)}{p_i}$$

$$I(p, Q) = \max_{\phi \in \Phi} \tilde{I}(p, Q; \phi),$$

where the optimal transition matrix ϕ is given by:

$$\phi^*(i | j) = p_i \frac{Q_{i,j}}{q_j} \triangleq p_i \frac{Q_{i,j}}{\sum_{i=1}^{|\mathcal{X}|} p_i Q_{i,j}}$$

Variantional Characterization of Mutual Information

- Suppose that $X \sim p$ and $Y | X \sim Q$, then

$$I(X; Y) \triangleq I(p, Q) = \sum_{i=1}^{|\mathcal{X}|} \sum_{j=1}^{|\mathcal{Y}|} p_i Q_{i,j} \log \frac{Q_{i,j}}{q_j}$$

where $Y \sim q \triangleq p \cdot Q$.

Theorem (Global Tight Lower Bound of Mutual Information)

$$I(p, Q) \geq \tilde{I}(p, Q; \phi) \triangleq \sum_{i=1}^{|\mathcal{X}|} \sum_{j=1}^{|\mathcal{Y}|} p_i Q_{i,j} \log \frac{\phi(i | j)}{p_i}$$

$$I(p, Q) = \max_{\phi \in \Phi} \tilde{I}(p, Q; \phi),$$

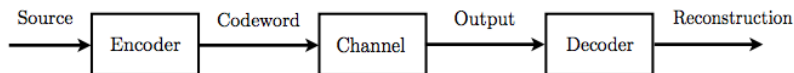
where the optimal transition matrix ϕ is given by:

$$\phi^*(i | j) = p_i \frac{Q_{i,j}}{q_j} \triangleq p_i \frac{Q_{i,j}}{\sum_{i=1}^{|\mathcal{X}|} p_i Q_{i,j}}$$

Variants of Channel Capacity Problem

- Channel Capacity Problem: for fixed transition matrix Q , we aim to solve

$$C(Q) := \max_{p_X} I_Q(X; Y)$$



- Rate Distortion Functions: for fixed distortion D ,

$$R(D) := \min_{Q(y|x)} I(X; Y), \quad \text{subject to } \mathbb{E}d(X; Y) \leq D.$$

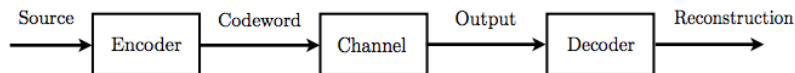
- Information Bottleneck Problem.

Comment: The method to be discussed can be applied to all those problems.

Variants of Channel Capacity Problem

- Channel Capacity Problem: for fixed transition matrix Q , we aim to solve

$$C(Q) := \max_{p_X} I_Q(X; Y)$$



- Rate Distortion Functions: for fixed distortion D ,

$$R(D) := \min_{Q(y|x)} I(X; Y), \quad \text{subject to } \mathbb{E}d(X; Y) \leq D.$$

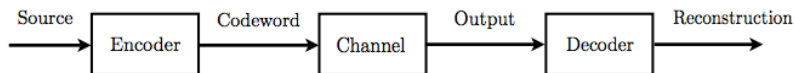
- Information Bottleneck Problem.

Comment: The method to be discussed can be applied to all those problems.

Variants of Channel Capacity Problem

- Channel Capacity Problem: for fixed transition matrix Q , we aim to solve

$$C(Q) := \max_{p_X} I_Q(X; Y)$$



- Rate Distortion Functions: for fixed distortion D ,

$$R(D) := \min_{Q(y|x)} I(X; Y), \quad \text{subject to } \mathbb{E}d(X; Y) \leq D.$$

- Information Bottleneck Problem.

Comment: The method to be discussed can be applied to all those problems.

Table of Contents

- 1 Introduction to Alternating Optimization
- 2 Introduction to EM Algorithm
- 3 Incremental Variants of EM (Alternating Maximization)
- 4 Introduction to Channel Capacity
- 5 Blahut-Arimoto Algorithm (Alternating Maximization)**
- 6 Variants of Blahut-Arimoto Algorithm (Proximal Point Algorithm)

Channel Capacity by Alternating Optimization (1)

The Channel Capacity is by maximizing over two groups of variables:

$$\mathcal{C} = \max_{\mathbf{p}} \max_{\phi \in \Phi} \tilde{I}(\mathbf{p}, \mathbf{Q}; \phi)$$

Key observation

- 1 For fixed \mathbf{p} and \mathbf{Q} , we have

$$\arg \max_{\phi \in \Phi} \tilde{I}(\mathbf{p}, \mathbf{Q}; \phi) = \phi^*, \quad \phi^*(i | j) = p_i \frac{Q_{i,j}}{\sum_{i=1}^{|\mathcal{X}|} p_i Q_{i,j}}$$

- 2 For fixed ϕ and \mathbf{Q} , we have

$$\arg \max_{\mathbf{p}} \tilde{I}(\mathbf{p}, \mathbf{Q}; \phi) = \mathbf{p}^*, \quad p^*(i) = \frac{r_i}{\sum_{i=1}^{|\mathcal{X}|} r_i},$$

$$r_i \triangleq \exp \left(\sum_{j=1}^{|\mathcal{Y}|} Q_{i,j} \log \phi(i | j) \right)$$

Channel Capacity by Alternating Optimization (1)

The Channel Capacity is by maximizing over two groups of variables:

$$\mathcal{C} = \max_{\mathbf{p}} \max_{\phi \in \Phi} \tilde{I}(\mathbf{p}, \mathbf{Q}; \phi)$$

Key observation

- 1 For fixed \mathbf{p} and \mathbf{Q} , we have

$$\arg \max_{\phi \in \Phi} \tilde{I}(\mathbf{p}, \mathbf{Q}; \phi) = \phi^*, \quad \phi^*(i | j) = p_i \frac{Q_{i,j}}{\sum_{i=1}^{|\mathcal{X}|} p_i Q_{i,j}}$$

- 2 For fixed ϕ and \mathbf{Q} , we have

$$\arg \max_{\mathbf{p}} \tilde{I}(\mathbf{p}, \mathbf{Q}; \phi) = \mathbf{p}^*, \quad p^*(i) = \frac{r_i}{\sum_{i=1}^{|\mathcal{X}|} r_i},$$

$$r_i \triangleq \exp \left(\sum_{j=1}^{|\mathcal{Y}|} Q_{i,j} \log \phi(i | j) \right)$$

Channel Capacity by Alternating Optimization (1)

The Channel Capacity is by maximizing over two groups of variables:

$$\mathcal{C} = \max_{\mathbf{p}} \max_{\phi \in \Phi} \tilde{I}(\mathbf{p}, \mathbf{Q}; \phi)$$

Key observation

- 1 For fixed \mathbf{p} and \mathbf{Q} , we have

$$\arg \max_{\phi \in \Phi} \tilde{I}(\mathbf{p}, \mathbf{Q}; \phi) = \phi^*, \quad \phi^*(i | j) = p_i \frac{Q_{i,j}}{\sum_{i=1}^{|\mathcal{X}|} p_i Q_{i,j}}$$

- 2 For fixed ϕ and \mathbf{Q} , we have

$$\arg \max_{\mathbf{p}} \tilde{I}(\mathbf{p}, \mathbf{Q}; \phi) = \mathbf{p}^*, \quad p^*(i) = \frac{r_i}{\sum_{i=1}^{|\mathcal{X}|} r_i},$$

$$r_i \triangleq \exp \left(\sum_{j=1}^{|\mathcal{Y}|} Q_{i,j} \log \phi(i | j) \right)$$

Channel Capacity by Alternating Optimization (2)

The Channel Capacity can be computed using alternating maximization:

$$\phi^{t+1} = \arg \max_{\phi \in \Phi} \tilde{I}(\mathbf{p}^t; \mathbf{Q}; \phi)$$

$$\mathbf{p}^{t+1} = \arg \max_{\mathbf{p}} \tilde{I}(\mathbf{p}; \mathbf{Q}; \phi^{t+1})$$

Key observation

1

$$\phi^{t+1}(i | j) = p_i^t \frac{Q_{i,j}}{\sum_{i=1}^{|\mathcal{X}|} p_i^t Q_{i,j}} \triangleq \frac{p_i^t Q_{i,j}}{q_j^{t+1}}$$

2 Substituting $\phi := \phi^{t+1}$ and for fixed \mathbf{Q} ,

$$p^{t+1}(i) = \frac{r_i^{t+1}}{\sum_{i=1}^{|\mathcal{X}|} r_i^{t+1}}, \quad \text{with } r_i^{t+1} \triangleq p_i^t \exp \left(D(\mathbf{Q}_i \| \mathbf{q}^{t+1}) \right)$$

Blahut-Arimoto Algorithm

Algorithm 7 Blahut-Arimoto Algorithm for Computing Channel Capacity

```

1:  $p^0 \leftarrow \text{INITIALIZE}();$ 
2: for  $t = 1, 2, \dots$  do
3:    $q^{t+1} \leftarrow p^t Q;$ 
4:

```

$$p_i^{t+1} \leftarrow \frac{p_i^t \exp(D(Q_i \| q^{t+1}))}{\sum_{i=1}^{|\mathcal{X}|} p_i^t \exp(D(Q_i \| q^{t+1}))}$$

```

5: end for

```

Table of Contents

- 1 Introduction to Alternating Optimization
- 2 Introduction to EM Algorithm
- 3 Incremental Variants of EM (Alternating Maximization)
- 4 Introduction to Channel Capacity
- 5 Blahut-Arimoto Algorithm (Alternating Maximization)
- 6 Variants of Blahut-Arimoto Algorithm (Proximal Point Algorithm)**

Re-consideration of Blahut-Arimoto Algorithm

- The Blahut-Arimoto Algorithm is essentially the alternating maximization:

$$\phi^{t+1} = \arg \max_{\phi \in \Phi} \tilde{I}(\mathbf{p}^t; \mathbf{Q}; \phi)$$

$$\mathbf{p}^{t+1} = \arg \max_{\mathbf{p}} \tilde{I}(\mathbf{p}; \mathbf{Q}; \phi^{t+1})$$

Theorem (Reformulation for the update of \mathbf{p}^{t+1})

$$\mathbf{p}^{t+1} = \arg \max_{\mathbf{p}} \left(I(\mathbf{p}^t, \mathbf{Q}) + \sum_{i=1}^N (p_i - p_i^t) \cdot D(\mathbf{Q}_i \| \mathbf{q}^t) - D(\mathbf{p} \| \mathbf{p}^t) \right)$$

Re-consideration of Blahut-Arimoto Algorithm

- The Blahut-Arimoto Algorithm is essentially the alternating maximization:

$$\phi^{t+1} = \arg \max_{\phi \in \Phi} \tilde{I}(\mathbf{p}^t; \mathbf{Q}; \phi)$$

$$\mathbf{p}^{t+1} = \arg \max_{\mathbf{p}} \tilde{I}(\mathbf{p}; \mathbf{Q}; \phi^{t+1})$$

Theorem (Reformulation for the update of \mathbf{p}^{t+1})

$$\mathbf{p}^{t+1} = \arg \max_{\mathbf{p}} \left(I(\mathbf{p}^t, \mathbf{Q}) + \sum_{i=1}^N (p_i - p_i^t) \cdot D(\mathbf{Q}_i \| \mathbf{q}^t) - D(\mathbf{p} \| \mathbf{p}^t) \right)$$

Re-consideration of Blahut-Arimoto Algorithm

Theorem (Reformulation for the update of \mathbf{p}^{t+1})

$$\mathbf{p}^{t+1} = \arg \max_{\mathbf{p}} \left(I(\mathbf{p}^t, \mathbf{Q}) + \sum_{i=1}^N (p_i - p_i^t) \cdot D(\mathbf{Q}_i \| \mathbf{q}^t) - D(\mathbf{p} \| \mathbf{p}^t) \right)$$

Proof.

By substituting the term ϕ^{t+1} into the update of \mathbf{p}^{t+1} :

$$\begin{aligned} \mathbf{p}^{t+1} &:= \arg \max_{\mathbf{p}} \tilde{I}(\mathbf{p}; \mathbf{Q}; \phi^{t+1}) = \arg \max_{\mathbf{p}} \sum_i \sum_j p_i Q_{i,j} \log \frac{\phi^{t+1}(i | j)}{p_i} \\ &= \arg \max_{\mathbf{p}} \sum_i \sum_j p_i Q_{i,j} \log \frac{p_i^t Q_{i,j}}{p_i q_j^{t+1}} \\ &= \arg \max_{\mathbf{p}} \left(\sum_{i=1}^N p_i \cdot D(\mathbf{Q}_i \| \mathbf{q}^t) - D(\mathbf{p} \| \mathbf{p}^t) \right) \end{aligned}$$



Proximal Point (Mirror Descent) Algorithm

Consider the maximization of a concave function $f(x)$:

$$\max_{x \in \mathcal{X}} f(x)$$

- Let $f_k(x)$ be a certain concave approximation of $f(x)$.
- Let $\{t_k > 0 \mid k = 0, 1, \dots\}$ be a sequence of parameters.
- The Bregman distance is a generalized choice of Euclidean distance:

$$B(y, x) = \Phi(y) - \Phi(x) - \nabla^T \Phi(x)(y - x),$$

where Φ is a smooth and strongly convex function

Algorithm 8 Proximal Point (Mirror Descent) Algorithm

$x^0 \leftarrow \text{INITIALIZE}();$

for $t = 0, 1, 2, \dots$ **do**

$$x^{t+1} \leftarrow \arg \max_{x \in \mathcal{X}} f_k(x) - \frac{1}{\gamma_t} B(x, x^t).$$

end for

Proximal Point (Mirror Descent) Algorithm

Consider the maximization of a concave function $f(x)$:

$$\max_{x \in \mathcal{X}} f(x)$$

- Let $f_k(x)$ be a certain concave approximation of $f(x)$.
- Let $\{t_k > 0 \mid k = 0, 1, \dots\}$ be a sequence of parameters.
- The Bregman distance is a generalized choice of Euclidean distance:

$$B(y, x) = \Phi(y) - \Phi(x) - \nabla^T \Phi(x)(y - x),$$

where Φ is a smooth and strongly convex function

Algorithm 9 Proximal Point (Mirror Descent) Algorithm

$x^0 \leftarrow \text{INITIALIZE}();$

for $t = 0, 1, 2, \dots$ **do**

$$x^{t+1} \leftarrow \arg \max_{x \in \mathcal{X}} f_k(x) - \frac{1}{\gamma_t} B(x, x^t).$$

end for

Proximal Point (Mirror Descent) Algorithm

Consider the maximization of a concave function $f(x)$:

$$\max_{x \in \mathcal{X}} f(x)$$

- Let $f_k(x)$ be a certain concave approximation of $f(x)$.
- Let $\{t_k > 0 \mid k = 0, 1, \dots\}$ be a sequence of parameters.
- The Bregman distance is a generalized choice of Euclidean distance:

$$B(y, x) = \Phi(y) - \Phi(x) - \nabla^T \Phi(x)(y - x),$$

where Φ is a smooth and strongly convex function

Algorithm 10 Proximal Point (Mirror Descent) Algorithm

$x^0 \leftarrow \text{INITIALIZE}();$

for $t = 0, 1, 2, \dots$ **do**

$$x^{t+1} \leftarrow \arg \max_{x \in \mathcal{X}} f_k(x) - \frac{1}{\gamma_t} B(x, x^t).$$

end for

Blahut-Arimoto Algorithm as a Proximal Point Algorithm

- The Blahut-Arimoto Algorithm is essentially the proximal point algorithm:

$$\mathbf{p}^{t+1} = \arg \max_{\mathbf{p}} \left(I_t(\mathbf{p}, \mathbf{Q}) - \frac{1}{\gamma_t} D(\mathbf{p} \| \mathbf{p}^t) \right)$$

where

- $\gamma_t \equiv 1$;
- I_t is the first order Taylor expansion of $I(\mathbf{p}, \mathbf{Q})$ around \mathbf{p}^t .

Theorem

$$\gamma_t \leq \frac{1}{\lambda_{KL}^2(\mathbf{Q})} \triangleq 1 / \left[\sup_{\mathbf{p} \neq \mathbf{p}'} \frac{D(\mathbf{p} \mathbf{Q} \| \mathbf{p}' \mathbf{Q})}{D(\mathbf{p} \| \mathbf{p}')} \right]$$

Blahut-Arimoto Algorithm as a Proximal Point Algorithm

- The Blahut-Arimoto Algorithm is essentially the proximal point algorithm:

$$\mathbf{p}^{t+1} = \arg \max_{\mathbf{p}} \left(I_t(\mathbf{p}, \mathbf{Q}) - \frac{1}{\gamma_t} D(\mathbf{p} \| \mathbf{p}^t) \right)$$

where

- ▶ $\gamma_t \equiv 1$;
- ▶ I_t is the first order Taylor expansion of $I(\mathbf{p}, \mathbf{Q})$ around \mathbf{p}^t .

Theorem

$$\gamma_t \leq \frac{1}{\lambda_{KL}^2(\mathbf{Q})} \triangleq 1 / \left[\sup_{\mathbf{p} \neq \mathbf{p}'} \frac{D(\mathbf{p} \mathbf{Q} \| \mathbf{p}' \mathbf{Q})}{D(\mathbf{p} \| \mathbf{p}')} \right]$$

Concluding Remarks

- EM and Blahut-Arimoto algorithms all belong to *alternating optimization* methods.
- Blahut-Arimoto algorithm can be understood as a generalized EM Algorithm:

$$\phi^{t+1} = \arg \max_{\phi \in \Phi} \tilde{I}(\mathbf{p}^t; \mathbf{Q}; \phi) \quad \text{E-step}$$

$$\mathbf{p}^{t+1} = \arg \max_{\mathbf{p}} \tilde{I}(\mathbf{p}; \mathbf{Q}; \phi^{t+1}) \quad \text{M-Step}$$

$$q^{t+1} = \arg \max -D(q \| q_{\theta^t})$$

$$\theta^{t+1} = \arg \max \mathbb{E}_{q^{t+1}} [\log p(z, x_{1:n} \mid \theta)]$$

Concluding Remarks

- EM and Blahut-Arimoto algorithms all belong to *alternating optimization* methods.
- Blahut-Arimoto algorithm can be understood as a generalized EM Algorithm:

$$\phi^{t+1} = \arg \max_{\phi \in \Phi} \tilde{I}(\mathbf{p}^t; \mathbf{Q}; \phi) \quad \text{E-step}$$

$$\mathbf{p}^{t+1} = \arg \max_{\mathbf{p}} \tilde{I}(\mathbf{p}; \mathbf{Q}; \phi^{t+1}) \quad \text{M-Step}$$

$$q^{t+1} = \arg \max -D(q \| q_{\theta^t})$$

$$\theta^{t+1} = \arg \max \mathbb{E}_{q^{t+1}} [\log p(z, x_{1:n} \mid \theta)]$$