Convergence of a distributed asynchronous learning vector quantization algorithm.

ENS ULM, NOVEMBER 2010

Outline.

1. Introduction.
2. Vector quantization, convergence of the CLVQ.
3. General distributed asynchronous algorithm.
4. Distributed Asynchronous Learning Vector Quantization (DALVQ).
5. Bibliography

Distributed computing.

- Distributed algorithms arise in a wide range of applications: including telecommunications, scientific computing...
- Parallelization: most promising way to allow more computing resources. Building faster serial computers: increasingly expensive + strikes physical limits (transmission speed, miniaturization).
- Distributed large scale algorithms encounter problems: communication delays (latency, bandwidth), the lack of efficient shared memory.

Distributed computing.

- Distributed algorithms arise in a wide range of applications: including telecommunications, scientific computing...
- Parallelization: most promising way to allow more computing resources. Building faster serial computers: increasingly expensive + strikes physical limits (transmission speed, miniaturization).
- Distributed large scale algorithms encounter problems: communication delays (latency, bandwidth), the lack of efficient shared memory.

Distributed computing.

- Distributed algorithms arise in a wide range of applications: including telecommunications, scientific computing...

- Parallelization: most promising way to allow more computing resources. Building faster serial computers: increasingly expensive + strikes physical limits (transmission speed, miniaturization).

- Distributed large scale algorithms encounter problems: communication delays (latency, bandwidth), the lack of efficient shared memory.

Figure: Chicago data center for Microsoft Windows Azure (Paas).

Clustering algorithms.

- Outstanding role in datamining: scientific data exploration, information retrieval, marketing, text mining, computational biology...
- Clustering: division of data into groups of similar objects.
- Representing data by clusters: loses certain fine details but achieves simplification.
- Probabilistic POV: find a simplified representation of the underlying distribution of the data.

Clustering algorithms.

- Outstanding role in datamining: scientific data exploration, information retrieval, marketing, text mining, computational biology...
- Clustering: division of data into groups of similar objects.
- Representing data by clusters: loses certain fine details but achieves simplification.
- Probabilistic POV: find a simplified representation of the underlying distribution of the data.

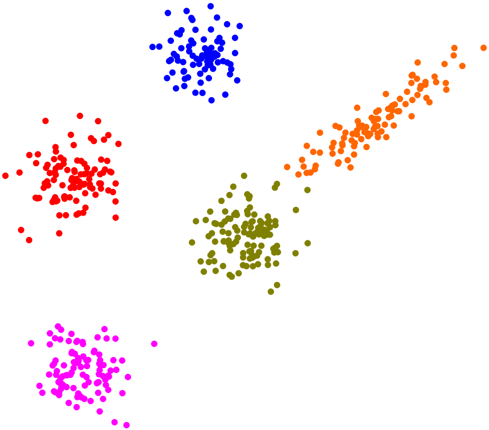


Figure: Division of data into similar (colored) groups: clustering.

Distortion.

- Data has a distribution μ: Borel probability measure on $\mathbb{R}^d$ (with a second order moment).
- Model this distribution by κ vectors of $\mathbb{R}^d$: the number of prototypes (centroids), $w \in \left(\mathbb{R}^d\right)^\kappa$.

Objective: minimization of the distortion C, find w° s.t.

$$w^\circ \in \operatorname*{argmin}_{w \in \left(\mathbb{R}^d\right)^\kappa} C(w),$$

where, for a quantization scheme $w = (w_1, \ldots, w_\kappa) \in \left(\mathbb{R}^d\right)^\kappa$,

$$C(w) \triangleq \frac{1}{2} \int_{\mathcal{G}} \min_{1 \le \ell \le \kappa} \left\| \mathbf{z} - w_\ell \right\|^2 d\mu(\mathbf{z}).$$

$\mathcal{G}$: closed convex hull of $\operatorname{supp}(\mu)$.

Distortion.

- Data has a distribution μ: Borel probability measure on $\mathbb{R}^d$ (with a second order moment).
- Model this distribution by κ vectors of $\mathbb{R}^d$: the number of prototypes (centroids), $w \in \left(\mathbb{R}^d\right)^\kappa$.

Objective: minimization of the distortion C, find w° s.t.

$$w^\circ \in \underset{w \in \left(\mathbb{R}^d\right)^\kappa}{\operatorname{argmin}}\, C(w),$$

where, for a quantization scheme $w = (w_1, \ldots, w_\kappa) \in \left(\mathbb{R}^d\right)^\kappa$,

$$C(w) \triangleq \frac{1}{2} \int_{\mathcal{G}} \min_{1 \le \ell \le \kappa} \left\| \mathbf{z} - w_\ell \right\|^2 d\mu(\mathbf{z}).$$

$\mathcal{G}$: closed convex hull of $\operatorname{supp}(\mu)$.

Distortion.

- Data has a distribution μ: Borel probability measure on $\mathbb{R}^d$ (with a second order moment).
- Model this distribution by κ vectors of $\mathbb{R}^d$: the number of prototypes (centroids), $w \in \left(\mathbb{R}^d\right)^\kappa$.

Objective: minimization of the distortion C, find w° s.t.

$$w^\circ \in \underset{w \in \left(\mathbb{R}^d\right)^\kappa}{\operatorname{argmin}} C(w),$$

where, for a quantization scheme $w = (w_1, \ldots, w_\kappa) \in \left(\mathbb{R}^d\right)^\kappa$,

$$C(w) \triangleq \frac{1}{2} \int_{\mathcal{G}} \min_{1 \le \ell \le \kappa} \|\mathbf{z} - w_\ell\|^2 \, d\mu(\mathbf{z}).$$

$\mathcal{G}$: closed convex hull of $\operatorname{supp}(\mu)$.

μ is only known through n independent random variables $\mathbf{z}_1, \ldots, \mathbf{z}_n$.

Much attention has been devoted to the consistency of the quantization scheme provided by the empirical minimizers

$$w_n^\circ = \underset{w \in \left(\mathbb{R}^d\right)^\kappa}{\operatorname{argmin}} \, C_n(w)$$

where

$$C_n(w) = \frac{1}{2} \int_{\mathcal{G}} \min_{1 \leq \ell \leq \kappa} \|\mathbf{z} - w_\ell\|^2 \, d\mu_n(\mathbf{z}) = \frac{1}{2n} \sum_{i=1}^{n} \min_{1 \leq \ell \leq \kappa} \|\mathbf{z}_i - w_\ell\|^2,$$

where

$$\mu_n \triangleq \frac{1}{n} \sum_{i=1}^{n} \delta_{\mathbf{z}_i}.$$

μ is only known through n independent random variables $\mathbf{z}_1, \ldots, \mathbf{z}_n$.

Much attention has been devoted to the consistency of the quantization scheme provided by the empirical minimizers

$$w_n^\circ = \underset{w \in \left(\mathbb{R}^d\right)^\kappa}{\operatorname{argmin}}\, C_n(w)$$

where

$$\begin{aligned} C_n(w) &= \frac{1}{2} \int_{\mathcal{G}} \min_{1 \leq \ell \leq \kappa} \left\| \mathbf{z} - w_\ell \right\|^2 d\mu_n(\mathbf{z}) \\ &= \frac{1}{2n} \sum_{i=1}^{n} \min_{1 \leq \ell \leq \kappa} \left\| \mathbf{z}_i - w_\ell \right\|^2, \end{aligned}$$

where

$$\mu_n \triangleq \frac{1}{n} \sum_{i=1}^{n} \delta_{\mathbf{z}_i}.$$

Pollard [1, 2] , Abaya et al. [3]:

$$C(w_n^\circ) \xrightarrow[n\to\infty]{a.s.} \min_{w\in\left(\mathbb{R}^d\right)^\kappa} C(w).$$

Rates of convergence, non asymptotic performance bounds: Pollard [4], Chou [5], Linder et al. [6], Bartlett et al [7], etc...

Inaba et al. [8] minimization of the empirical distortion is a computationally hard problem: complexity exponential in κ and d.

Untractable for most of the practical applications.

Here, investigate effective methods that produce accurate quantizations with data samples.

Pollard [1, 2] , Abaya et al. [3]:

$$C(w_n^\circ) \xrightarrow[n\to\infty]{a.s.} \min_{w\in(\mathbb{R}^d)^\kappa} C(w).$$

Rates of convergence, non asymptotic performance bounds: Pollard [4], Chou [5], Linder et al. [6], Bartlett et al [7], etc...

Inaba et al. [8] minimization of the empirical distortion is a computationally hard problem: complexity exponential in κ and d.

Untractable for most of the practical applications.

Here, investigate effective methods that produce accurate quantizations with data samples.

Pollard [1, 2] , Abaya et al. [3]:

$$C(w_n^\circ) \xrightarrow[n\to\infty]{a.s.} \min_{w\in\left(\mathbb{R}^d\right)^\kappa} C(w).$$

Rates of convergence, non asymptotic performance bounds: Pollard

[4], Chou [5], Linder et al. [6], Bartlett et al [7], etc...

Inaba et al. [8] minimization of the empirical distortion is a computationally hard problem: complexity exponential in κ and d.

Untractable for most of the practical applications.

Here, investigate effective methods that produce accurate quantizations with data samples.

Pollard [1, 2] , Abaya et al. [3]:

$$C(w_n^\circ) \xrightarrow[n\to\infty]{a.s.} \min_{w\in\left(\mathbb{R}^d\right)^\kappa} C(w).$$

Rates of convergence, non asymptotic performance bounds: Pollard

[4], Chou [5], Linder et al. [6], Bartlett et al [7], etc...

Inaba et al. [8] minimization of the empirical distortion is a computationally hard problem: complexity exponential in κ and d.

Untractable for most of the practical applications.

Here, investigate effective methods that produce accurate quantizations with data samples.

Assumption on the distribution.

We will make the following assumption.

Assumption (Compact supported density)

μ has a bounded density (w.r.t. Lebesgue measure) whose support is the compact convex set $\mathcal{G}$.

This assumption is similar to the peak power constraint (see Chou [5] and Linder [9]).

Voronoï tesselations.

Notation:

- The set of all κ-tuples of $\mathcal{G}$ is denoted $\mathcal{G}^\kappa$.
- $\mathcal{D}_*^\kappa = \left\{ w \in \left(\mathbb{R}^d\right)^\kappa | w_\ell \neq w_k \text{ if and only if } \ell \neq k \right\}$.

$\forall w \in \mathcal{D}_*^\kappa$,

$$C(w) = \frac{1}{2} \sum_{\ell=1}^{\kappa} \int_{W_\ell(w)} \|\mathbf{z} - w_\ell\|^2 \, d\mu(\mathbf{z}).$$

Voronoï tesselations.

Notation:

- The set of all κ-tuples of $\mathcal{G}$ is denoted $\mathcal{G}^\kappa$.
- $\mathcal{D}_*^\kappa = \left\{ w \in \left(\mathbb{R}^d\right)^\kappa | w_\ell \neq w_k \text{ if and only if } \ell \neq k \right\}$.

$\forall w \in \mathcal{D}_*^\kappa$,

$$C(w) = \frac{1}{2} \sum_{\ell=1}^{\kappa} \int_{W_\ell(w)} \|\mathbf{z} - w_\ell\|^2 \, d\mu(\mathbf{z}).$$

Definition

Let $w \in \left(\mathbb{R}^d\right)^\kappa$, the Voronoï tessellation of $\mathcal{G}$ related to w is the family of open sets $\{W_\ell(w)\}_{1\leq\ell\leq\kappa}$ defined as follows:

- If $w \in \mathcal{D}_*^\kappa$, for all $1 \leq \ell \leq \kappa$,

$$W_\ell(w) = \left\{ v \in \mathcal{G} \;\middle|\; \|w_\ell - v\| < \min_{k\neq\ell} \|w_k - v\| \right\}.$$

- If $w \in \left(\mathbb{R}^d\right)^\kappa \setminus \mathcal{D}_*^\kappa$, for all $1 \leq \ell \leq \kappa$,
  - if $\ell = \min\{k | w_k = w_\ell\}$,

$$W_\ell(w) = \left\{ v \in \mathcal{G} \;\middle|\; \|w_\ell - v\| < \min_{w_k\neq w_\ell} \|w_k - v\| \right\},$$

  - otherwise, $W_\ell(w) = \emptyset$.

Definition

Let $w \in \left(\mathbb{R}^d\right)^\kappa$, the Voronoï tessellation of $\mathcal{G}$ related to w is the family of open sets $\{W_\ell(w)\}_{1 \le \ell \le \kappa}$ defined as follows:

- If $w \in \mathcal{D}_*^\kappa$, for all $1 \le \ell \le \kappa$,

$$W_\ell(w) = \left\{ v \in \mathcal{G} \;\middle|\; \|w_\ell - v\| < \min_{k \neq \ell} \|w_k - v\| \right\}.$$

- If $w \in \left(\mathbb{R}^d\right)^\kappa \setminus \mathcal{D}_*^\kappa$, for all $1 \le \ell \le \kappa$,
  - if $\ell = \min\{k | w_k = w_\ell\}$,

$$W_\ell(w) = \left\{ v \in \mathcal{G} \;\middle|\; \|w_\ell - v\| < \min_{w_k \neq w_\ell} \|w_k - v\| \right\},$$

  - otherwise, $W_\ell(w) = \emptyset$.

Voronoï tesselations 2D.

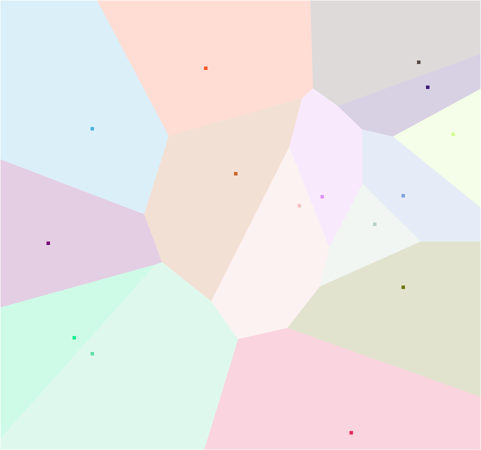


Figure: Voronoï tesselations of a vector of $\left(\mathbb{R}^2\right)^{15}$.

CLVQ

Competitive Learning Vector Quantization (CLVQ).

- Data arrive over time while the execution of the algorithm and their characteristics are unknown until their arrival times.
- On-line algorithm: uses each item of the training sequence at each update.

Data stream $\mathbf{z}_1, \mathbf{z}_2, \ldots$.

Initialization with κ-prototypes $w(0) = (w_1(0), \ldots, w_\kappa(0))$.
For each $t = 0, \ldots$
ℓ_0 s.t. $w_{\ell_0}(t)$ nearest prototype of $\mathbf{z}_{t+1}$ among $(w_1(t), \ldots, w_\kappa(t))$

$$w_{\ell_0}(t+1) = w_{\ell_0}(t) + \varepsilon_{t+1}(\mathbf{z}_{t+1} - w_{\ell_0}(t)),$$

$\varepsilon_t \in (0, 1)$.

CLVQ

Competitive Learning Vector Quantization (CLVQ).

- Data arrive over time while the execution of the algorithm and their characteristics are unknown until their arrival times.
- On-line algorithm: uses each item of the training sequence at each update.

Data stream $\mathbf{z}_1, \mathbf{z}_2, \ldots$.

Initialization with κ-prototypes $w(0) = (w_1(0), \ldots, w_\kappa(0))$.
For each $t = 0, \ldots$
ℓ_0 s.t. $w_{\ell_0}(t)$ nearest prototype of $\mathbf{z}_{t+1}$ among $(w_1(t), \ldots, w_\kappa(t))$

$$w_{\ell_0}(t+1) = w_{\ell_0}(t) + \varepsilon_{t+1}(\mathbf{z}_{t+1} - w_{\ell_0}(t)),$$

$\varepsilon_t \in (0, 1)$.

Video (short).

Video (long).

Regularity of the distortion.

Theorem (Pagès [1].)

C is continuously differentiable at every $w = (w_1, \dots, w_\kappa) \in \mathcal{D}_*^\kappa$.
$\forall 1 \le \ell \le \kappa$,

$$\nabla_\ell C(w) = \int_{W_\ell(w)} (w_\ell - \mathbf{z})\, d\mu(\mathbf{z}).$$

Local observation of the gradient.

Definition

For any $\mathbf{z} \in \mathbb{R}^d$ and $w \in \mathcal{D}_*^\kappa$, define function H by its ℓ-th component,

$$H_\ell(\mathbf{z}, w) = \begin{cases} \mathbf{z} - w_\ell & \text{if } \mathbf{z} \in W_\ell(w) \\ 0 & \text{otherwise.} \end{cases}$$

If random variable $\mathbf{z} \sim \mu$, the next equality holds for all $w \in \mathcal{D}_*^\kappa$,

$$\mathbb{E}\left\{H(\mathbf{z}, w)\right\} = \nabla C(w).$$

Thus, we extend the definition, for all $w \in \left(\mathbb{R}^d\right)^\kappa$,

$$h(w) \triangleq \mathbb{E}\left\{H(\mathbf{z}, w)\right\}.$$

Local observation of the gradient.

Definition

For any $\mathbf{z} \in \mathbb{R}^d$ and $w \in \mathcal{D}_*^\kappa$, define function H by its ℓ-th component,

$$H_\ell(\mathbf{z}, w) = \begin{cases} \mathbf{z} - w_\ell & \text{if } \mathbf{z} \in W_\ell(w) \\ 0 & \text{otherwise.} \end{cases}$$

If random variable $\mathbf{z} \sim \mu$, the next equality holds for all $w \in \mathcal{D}_*^\kappa$,

$$\mathbb{E}\left\{H(\mathbf{z}, w)\right\} = \nabla C(w).$$

Thus, we extend the definition, for all $w \in \left(\mathbb{R}^d\right)^\kappa$,

$$h(w) \triangleq \mathbb{E}\left\{H(\mathbf{z}, w)\right\}.$$

Local observation of the gradient.

Definition

For any $\mathbf{z} \in \mathbb{R}^d$ and $w \in \mathcal{D}_*^\kappa$, define function H by its ℓ-th component,

$$H_\ell(\mathbf{z}, w) = \begin{cases} \mathbf{z} - w_\ell & \text{if } \mathbf{z} \in W_\ell(w) \\ 0 & \text{otherwise.} \end{cases}$$

If random variable $\mathbf{z} \sim \mu$, the next equality holds for all $w \in \mathcal{D}_*^\kappa$,

$$\mathbb{E}\left\{H(\mathbf{z}, w)\right\} = \nabla C(w).$$

Thus, we extend the definition, for all $w \in \left(\mathbb{R}^d\right)^\kappa$,

$$h(w) \triangleq \mathbb{E}\left\{H(\mathbf{z}, w)\right\}.$$

Stochastic gradient optimization.

Minimize C: gradient descent procedure $w := w - \varepsilon \nabla C(w)$.

$\nabla C(w)$ is unknown, use $H(\mathbf{z}, w)$ instead.

$$w(t+1) = w(t) - \varepsilon_{t+1} H\left(\mathbf{z}_{t+1}, w(t)\right) \quad \text{(CLVQ)},$$

$w(0) \in \overset{\circ}{\mathcal{G}^{\kappa}} \cap \mathcal{D}_*^{\kappa}$ and $\mathbf{z}_1, \mathbf{z}_2 \dots$ are independent observations distributed according to the probability measure μ.

Usual constraints on the decreasing speed of the sequence of steps $\{\varepsilon_t\}_{t=0}^{\infty} \in (0,1)$,

1. $\sum_{t=0}^{\infty} \varepsilon_t = \infty$.
2. $\sum_{t=0}^{\infty} \varepsilon_t^2 < \infty$.

Stochastic gradient optimization.

Minimize C: gradient descent procedure $w := w - \varepsilon \nabla C(w)$.

$\nabla C(w)$ is unknown, use $H(\mathbf{z}, w)$ instead.

$$w(t+1) = w(t) - \varepsilon_{t+1} H\left(\mathbf{z}_{t+1}, w(t)\right) \quad \text{(CLVQ)},$$

$w(0) \in \overset{\circ}{\mathcal{G}^{\kappa}} \cap \mathcal{D}_*^{\kappa}$ and $\mathbf{z}_1, \mathbf{z}_2 \ldots$ are independent observations distributed according to the probability measure μ.

Usual constraints on the decreasing speed of the sequence of steps $\{\varepsilon_t\}_{t=0}^{\infty} \in (0,1)$,

1. $\sum_{t=0}^{\infty} \varepsilon_t = \infty$.
2. $\sum_{t=0}^{\infty} \varepsilon_t^2 < \infty$.

Stochastic gradient optimization.

Minimize C: gradient descent procedure $w := w - \varepsilon \nabla C(w)$.

$\nabla C(w)$ is unknown, use $H(\mathbf{z}, w)$ instead.

$$w(t+1) = w(t) - \varepsilon_{t+1} H\left(\mathbf{z}_{t+1}, w(t)\right) \quad \text{(CLVQ)},$$

$w(0) \in \overset{\circ}{\mathcal{G}^{\kappa}} \cap \mathcal{D}_{*}^{\kappa}$ and $\mathbf{z}_1, \mathbf{z}_2 \ldots$ are independent observations distributed according to the probability measure μ.

Usual constraints on the decreasing speed of the sequence of steps $\{\varepsilon_t\}_{t=0}^{\infty} \in (0,1)$,

1. $\sum_{t=0}^{\infty} \varepsilon_t = \infty$.
2. $\sum_{t=0}^{\infty} \varepsilon_t^2 < \infty$.

Stochastic gradient optimization.

Minimize C: gradient descent procedure $w := w - \varepsilon \nabla C(w)$.

$\nabla C(w)$ is unknown, use $H(\mathbf{z}, w)$ instead.

$$w(t+1) = w(t) - \varepsilon_{t+1} H\left(\mathbf{z}_{t+1}, w(t)\right) \quad \text{(CLVQ)},$$

$w(0) \in \overset{\circ}{\mathcal{G}}{}^{\kappa} \cap \mathcal{D}_{*}^{\kappa}$ and $\mathbf{z}_1, \mathbf{z}_2 \ldots$ are independent observations distributed according to the probability measure μ.

Usual constraints on the decreasing speed of the sequence of steps $\{\varepsilon_t\}_{t=0}^{\infty} \in (0,1)$,

1. $\sum_{t=0}^{\infty} \varepsilon_t = \infty$.
2. $\sum_{t=0}^{\infty} \varepsilon_t^2 < \infty$.

Troubles.

On the distortion:

- C is not a convex function.
- $\|C(w)\| \nrightarrow \infty$ as $\|w\| \to \infty$.

On its gradient:

- h is singular at $\complement \mathcal{D}_*^\kappa$.
- h is zero on wide zone outside $\mathcal{G}^\kappa$.

Troubles.

On the distortion:

- C is not a convex function.
- $\|C(w)\| \nrightarrow \infty$ as $\|w\| \to \infty$.

On its gradient:

- h is singular at $\complement \mathcal{D}_*^\kappa$.
- h is zero on wide zone outside $\mathcal{G}^\kappa$.

What can be expected?

$$w(t) \nrightarrow w^\circ = \underset{w \in \mathcal{G}^\kappa}{\operatorname{argmin}}\, C(w), \quad \text{almost surely (a.s.).}$$

Proposition (Pagès [1].)

$$\underset{w \in (\mathbb{R}^d)^\kappa}{\operatorname{argmin}}\, C(w) \subset \underset{w \in \mathcal{G}^\kappa}{\operatorname{argminloc}}\, C(w) \subset \overset{\circ}{\mathcal{G}^\kappa} \cap \{\nabla C = 0\} \cap \mathcal{D}_*^\kappa.$$

$$w(t) \xrightarrow[t \to \infty]{a.s.} \overset{\circ}{\mathcal{G}^\kappa} \cap \{\nabla C = 0\} \cap \mathcal{D}_*^\kappa.$$

What can be expected?

$$w(t) \nrightarrow w^\circ = \underset{w \in \mathcal{G}^\kappa}{\operatorname{argmin}}\, C(w), \quad \text{almost surely (a.s.).}$$

Proposition (Pagès [1].)

$$\underset{w \in (\mathbb{R}^d)^\kappa}{\operatorname{argmin}}\, C(w) \subset \underset{w \in \mathcal{G}^\kappa}{\operatorname{argminloc}}\, C(w) \subset \overset{\circ}{\mathcal{G}}{}^\kappa \cap \{\nabla C = 0\} \cap \mathcal{D}_*^\kappa.$$

$$w(t) \xrightarrow[t \to \infty]{a.s.} \overset{\circ}{\mathcal{G}}{}^\kappa \cap \{\nabla C = 0\} \cap \mathcal{D}_*^\kappa.$$

What can be expected?

$$w(t) \nrightarrow w^{\circ} = \operatorname*{argmin}_{w \in \mathcal{G}^{\kappa}} C(w), \quad \text{almost surely (a.s.).}$$

Proposition (Pagès [1].)

$$\operatorname*{argmin}_{w \in (\mathbb{R}^d)^{\kappa}} C(w) \subset \operatorname*{argminloc}_{w \in \mathcal{G}^{\kappa}} C(w) \subset \overset{\circ}{\mathcal{G}^{\kappa}} \cap \{\nabla C = 0\} \cap \mathcal{D}_*^{\kappa}.$$

$$w(t) \xrightarrow[t \to \infty]{a.s.} \overset{\circ}{\mathcal{G}^{\kappa}} \cap \{\nabla C = 0\} \cap \mathcal{D}_*^{\kappa}.$$

Theorem (G-Lemma, Fort and Pagès [2].)

Assume that:

1. $\{w(t)\}_{t=0}^{\infty}$ *and* $\{h(w(t))\}_{t=0}^{\infty}$ *are bounded with probability 1.*
2. *The series* $\sum_{t=0}^{\infty} \varepsilon_{t+1} \left(H(\mathbf{z}_{t+1}, w(t)) - h(w(t))\right)$ *converge a.s. in* $\left(\mathbb{R}^d\right)^{\kappa}$.
3. *There exists a l.s.c. nonnegative function* $G : \left(\mathbb{R}^d\right)^{\kappa} \to \mathbb{R}_+$ *s.t.*

$$\sum_{s=0}^{\infty} \varepsilon_{s+1} G(w(s)) < \infty \quad a.s..$$

Then there exists a connected component Ξ *of* $\{G = 0\}$ *s.t.*

$$\lim_{t \to \infty} \operatorname{dist}(w(t), \Xi) = 0 \quad a.s..$$

A suitable G:

For every $w \in \mathcal{G}^\kappa$,

$$\widehat{G}(w) \triangleq \liminf_{v \in \mathcal{G}^\kappa \cap \mathcal{D}_*^\kappa, v \to w} \|\nabla C(v)\|^2 .$$

$\widehat{G}$ is a nonnegative l.s.c. function on $\mathcal{G}^\kappa$.

A suitable G:

For every $w \in \mathcal{G}^\kappa$,

$$\widehat{G}(w) \triangleq \liminf_{v \in \mathcal{G}^\kappa \cap \mathcal{D}_*^\kappa, v \to w} \|\nabla C(v)\|^2 .$$

$\widehat{G}$ is a nonnegative l.s.c. function on $\mathcal{G}^\kappa$.

Theorem (Pagès [1].)

Under assumption [Compact supported density], on the event

$$\left\{ \liminf_{t\to\infty} \mathrm{dist}\left(w(t), \complement\mathcal{D}_*^{\kappa} \right) > 0 \right\},$$

$$\mathrm{dist}\left(w(t), \Xi_\infty \right) = 0 \quad \textit{a.s. as } t \to \infty,$$

where Ξ_∞ is some connected component of $\{\nabla C = 0\}$.

Remarks:

- Asymptotically parted component.
- No satisfactory convergence result is provided without this assumption.
- However, some studies have been carried out by Pagès [1].

Theorem (Pagès [1].)

Under assumption [Compact supported density], on the event

$$\left\{ \liminf_{t\to\infty} \mathrm{dist}\left(w(t), \complement\mathcal{D}_*^{\kappa}\right) > 0 \right\},$$

$$\mathrm{dist}\left(w(t), \Xi_\infty\right) = 0 \quad \textit{a.s. as } t \to \infty,$$

where Ξ_∞ *is some connected component of* $\{\nabla C = 0\}$.

Remarks:

- Asymptotically parted component.
- No satisfactory convergence result is provided without this assumption.
- However, some studies have been carried out by Pagès [1].

Parallelization.

Why?

- On line algorithm have impressive convergence properties.
- Such algorithms are entirely sequential in their nature.
- Thus, CLVQ algorithm is too slow on large data sets or with high dimension data.

Parallelization.

Why?

- On line algorithm have impressive convergence properties.
- Such algorithms are entirely sequential in their nature.
- Thus, CLVQ algorithm is too slow on large data sets or with high dimension data.

Parallelization.

Why?

- On line algorithm have impressive convergence properties.
- Such algorithms are entirely sequential in their nature.
- Thus, CLVQ algorithm is too slow on large data sets or with high dimension data.

Parallelization.

- We introduce a model that brings together the CLVQ and the comprehensive theory of asynchronous parallel linear algorithms (Tsitsiklis [3]).
- Resulting model will be called Distributed Asynchronous Learning Vector Quantization (DALVQ).
- DALVQ parallelizes several executions of CLVQ concurrently at different processors while the results of theses latter algorithms are broadcasted through the distributed framework in efficient way.
- Our parallel DALVQ algorithm is able to process, for a given time span, much more data than a (single processor) execution of the CLVQ procedure.

Parallelization.

- We introduce a model that brings together the CLVQ and the comprehensive theory of asynchronous parallel linear algorithms (Tsitsiklis [3]).
- Resulting model will be called Distributed Asynchronous Learning Vector Quantization (DALVQ).
- DALVQ parallelizes several executions of CLVQ concurrently at different processors while the results of theses latter algorithms are broadcasted through the distributed framework in efficient way.
- Our parallel DALVQ algorithm is able to process, for a given time span, much more data than a (single processor) execution of the CLVQ procedure.

Parallelization.

- We introduce a model that brings together the CLVQ and the comprehensive theory of asynchronous parallel linear algorithms (Tsitsiklis [3]).
- Resulting model will be called Distributed Asynchronous Learning Vector Quantization (DALVQ).
- DALVQ parallelizes several executions of CLVQ concurrently at different processors while the results of theses latter algorithms are broadcasted through the distributed framework in efficient way.
- Our parallel DALVQ algorithm is able to process, for a given time span, much more data than a (single processor) execution of the CLVQ procedure.

Parallelization.

- We introduce a model that brings together the CLVQ and the comprehensive theory of asynchronous parallel linear algorithms (Tsitsiklis [3]).
- Resulting model will be called Distributed Asynchronous Learning Vector Quantization (DALVQ).
- DALVQ parallelizes several executions of CLVQ concurrently at different processors while the results of theses latter algorithms are broadcasted through the distributed framework in efficient way.
- Our parallel DALVQ algorithm is able to process, for a given time span, much more data than a (single processor) execution of the CLVQ procedure.

Distributed framework.

- We dispose of a distributed architecture with M computing entities called processors/workers.
- Each processor is labeled by a natural number $i \in \{1, \ldots, M\}$.
- Each processor i has a buffer (local memory) where the current version of the iteration is kept: $\left\{w^i(t)\right\}_{t=0}^{\infty}$, $\left(\mathbb{R}^d\right)^{\kappa}$-valued sequence.

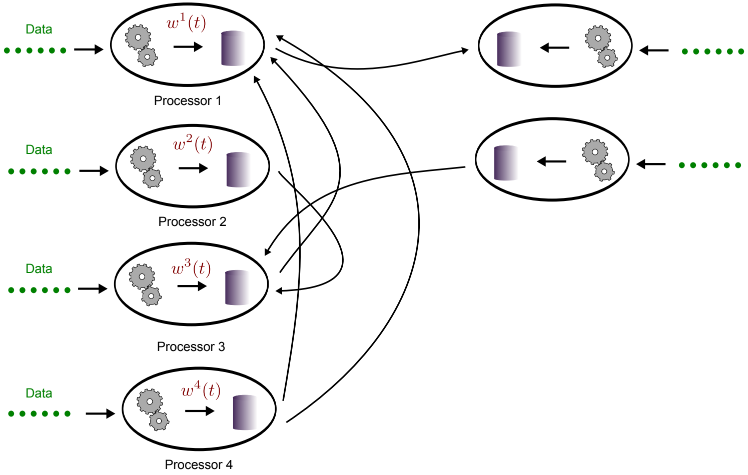
Distributed framework.

- We dispose of a distributed architecture with M computing entities called processors/workers.
- Each processor is labeled by a natural number $i \in \{1, \ldots, M\}$.
- Each processor i has a buffer (local memory) where the current version of the iteration is kept: $\{w^i(t)\}_{t=0}^{\infty}$, $(\mathbb{R}^d)^{\kappa}$-valued sequence.

Distributed framework.

- We dispose of a distributed architecture with M computing entities called processors/workers.
- Each processor is labeled by a natural number $i \in \{1, \dots, M\}$.
- Each processor i has a buffer (local memory) where the current version of the iteration is kept: $\left\{w^i(t)\right\}_{t=0}^{\infty}$, $\left(\mathbb{R}^d\right)^{\kappa}$-valued sequence.

Distributed framework.



Independent.

Independent

A generic descent term:

$$w(t+1) = w(t) + \underbrace{-\varepsilon_t H\left(\mathbf{z}_{t+1}, w(t)\right)}_{\triangleq s(t)}.$$

Parallelization.

Basic parallelization.

For all $1 \leq i \leq M$, where M is the number of processors.

$$w^i(t+1) = \sum_{j=1}^{M} a^{i,j}(t) w^j(t) + s^i(t).$$

Where the $\left\{a^{i,j}(t)\right\}_{j=1}^{M}$ are some weights (convex combination).

For many $t \geq 0$,

$$a^{i,j}(t) = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise.} \end{cases}$$

For such values: local iterations

$$w^i(t+1) = w^i(t) + s^i(t)$$

Synchronization effects:

- Synchronizations required in this model.
- We should take into account communication delays and design an asynchronous algorithm.
- Local algorithms do not have to wait at preset points for some messages to become available.
- Processors compute faster and execute more iterations than others. Communication delays are allowed to be substantial and unpredictable.
- Messages can be deliver out of order (a different order than the one in which they were transmitted).

Advantages

- Reduction of the synchronization penalty: speed advantage over a synchronous execution.
- For a potential industrialization, asynchronism has a greater implementation flexibility.

Advantages

- Reduction of the synchronization penalty: speed advantage over a synchronous execution.
- For a potential industrialization, asynchronism has a greater implementation flexibility.

The Tsitsikils's asynchronous model.

General Distributed Asynchronous System (GDAS), Tsitsklis [3, 4]:

$$w^i(t+1) = \sum_{j=1}^{M} a^{i,j}(t) w^j(\tau^{i,j}(t)) + s^i(t).$$

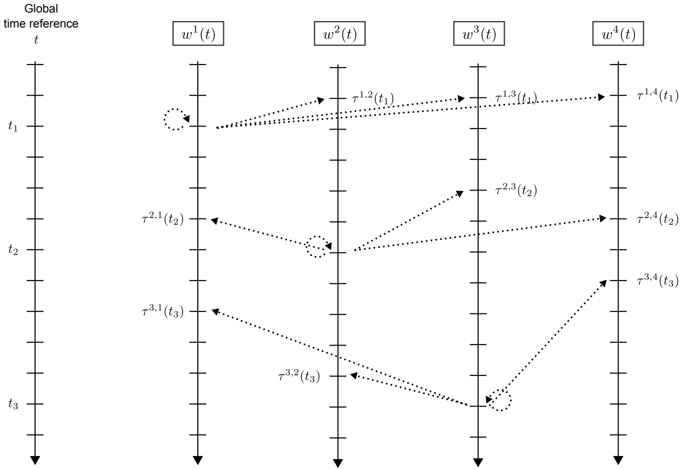
- $0 \leq \tau^{i,j}(t) \leq t$: deterministic (but unknown) time instant.
- $t - \tau^{i,j}(t)$: communication delays.
- $\tau^{i,i}(t) = t$.

The Tsitsikils's asynchronous model.

General Distributed Asynchronous System (GDAS), Tsitsklis [3, 4]:

$$w^i(t+1) = \sum_{j=1}^{M} a^{i,j}(t) w^j(\tau^{i,j}(t)) + s^i(t).$$

- $0 \leq \tau^{i,j}(t) \leq t$: deterministic (but unknown) time instant.
- $t - \tau^{i,j}(t)$: communication delays.
- $\tau^{i,i}(t) = t$.


Global
time reference
t
$w^1(t)$
$w^2(t)$
$w^3(t)$
$w^4(t)$
t_1
t_2
t_3
$\tau^{1,2}(t_1)$
$\tau^{1,3}(t_1)$
$\tau^{1,4}(t_1)$
$\tau^{2,3}(t_2)$
$\tau^{2,1}(t_2)$
$\tau^{2,4}(t_2)$
$\tau^{3,4}(t_3)$
$\tau^{3,1}(t_3)$
$\tau^{3,2}(t_3)$

Model agreement.

Agreement algorithm.

$$x^i(t+1) = \sum_{j=1}^{M} a^{i,j}(t) x^j(\tau^{i,j}(t)),$$

$x^i(0) \in \left(\mathbb{R}^d\right)^\kappa$, for all i.

Remark:
Similar to (GDAS) but with $s^i(t) = 0$ for all t, i.

Is there (or at least what are the conditions to ensure) an asymptotical consensus between the processors/workers?

Model agreement.

Agreement algorithm.

$$x^i(t+1) = \sum_{j=1}^{M} a^{i,j}(t) x^j(\tau^{i,j}(t)),$$

$x^i(0) \in \left(\mathbb{R}^d\right)^\kappa$, for all i.

Remark:
Similar to (GDAS) but with $s^i(t) = 0$ for all t, i.

Is there (or at least what are the conditions to ensure) an asymptotical consensus between the processors/workers?

Model agreement.

Agreement algorithm.

$$x^i(t+1) = \sum_{j=1}^{M} a^{i,j}(t) x^j(\tau^{i,j}(t)),$$

$x^i(0) \in \left(\mathbb{R}^d\right)^\kappa$, for all i.

Remark:
Similar to (GDAS) but with $s^i(t) = 0$ for all t, i.

Is there (or at least what are the conditions to ensure) an asymptotical consensus between the processors/workers?

Assumptions 1.

Assumption (Bounded communication delays)

There exists a positive integer B_1 s.t.

$$t - B_1 < \tau^{i,j}(t) \leq t,$$

for all $(i,j) \in \{1,\dots,M\}^2$ and all $t \geq 0$.

Assumption (Convex combination and threshold)

There exists $\alpha > 0$ s.t. the following three properties hold:

1. $a^{i,i}(t) \geq \alpha, \quad i \in \{1,\dots,M\}$ *and* $t \geq 0$,
2. $a^{i,j}(t) \in \{0\} \cup [\alpha, 1], \quad (i,j) \in \{1,\dots,M\}^2$ *and* $t \geq 0$,
3. $\sum_{j=1}^{M} a^{i,j}(t) = 1, \quad i \in \{1,\dots,M\}$ *and* $t \geq 0$.

Assumptions 1.

Assumption (Bounded communication delays)

There exists a positive integer B_1 s.t.

$$t - B_1 < \tau^{i,j}(t) \le t,$$

for all $(i,j) \in \{1,\ldots,M\}^2$ and all $t \ge 0$.

Assumption (Convex combination and threshold)

There exists $\alpha > 0$ s.t. the following three properties hold:

1. $a^{i,i}(t) \ge \alpha, \quad i \in \{1,\ldots,M\}$ *and* $t \ge 0$,
2. $a^{i,j}(t) \in \{0\} \cup [\alpha, 1], \quad (i,j) \in \{1,\ldots,M\}^2$ *and* $t \ge 0$,
3. $\sum_{j=1}^{M} a^{i,j}(t) = 1, \quad i \in \{1,\ldots,M\}$ *and* $t \ge 0$.

Assumption 2.

Definition (Communication graph)

Let us fix $t \geq 0$, the communication graph $(\mathcal{V}, E(t))$ is defined by

- the set of vertices $\mathcal{V}$ is formed by the set of processors, $\mathcal{V} = \{1, \ldots, M\}$,
- the set of edges $E(t)$ is defined via the relationship

$$(j, i) \in E(t) \text{ if and only if } a^{i,j}(t) > 0.$$

Assumption (Graph connectivity)

The graph $(\mathcal{V}, \cup_{s \geq t} E(s))$ *is strongly connected for all* $t \geq 0$.

Assumption 2.

Definition (Communication graph)

Let us fix $t \geq 0$, the communication graph $(\mathcal{V}, E(t))$ is defined by

- the set of vertices $\mathcal{V}$ is formed by the set of processors, $\mathcal{V} = \{1, \ldots, M\}$,
- the set of edges $E(t)$ is defined via the relationship

$$(j, i) \in E(t) \text{ if and only if } a^{i,j}(t) > 0.$$

Assumption (Graph connectivity)

The graph $(\mathcal{V}, \cup_{s \geq t} E(s))$ *is strongly connected for all* $t \geq 0$.

Assumption 3 and Assumption 4.

Assumption (Bounded communication intervals)

If i communicates with j an infinite number of times, then there is a positive integer B_2 such that, for all $t \geq 0$,
$(i, j) \in E(t) \cup E(t+1) \cup \ldots \cup E(t+B_2-1)$.

Assumption (Symmetry)

There exists some $B_3 > 0$ such that, whenever $(i, j) \in E(t)$, there exists some τ that satisfies $|t-\tau| < B_3$ and $(j, i) \in E(\tau)$.

Assumption 3 and Assumption 4.

Assumption (Bounded communication intervals)

If i communicates with j an infinite number of times, then there is a positive integer B_2 such that, for all $t \geq 0$, $(i, j) \in E(t) \cup E(t+1) \cup \ldots \cup E(t + B_2 - 1)$.

Assumption (Symmetry)

There exists some $B_3 > 0$ such that, whenever $(i, j) \in E(t)$, there exists some τ that satisfies $|t - \tau| < B_3$ and $(j, i) \in E(\tau)$.

Until the end of the presentation either $(AsY)_1$ or $(AsY)_2$ holds

$$(AsY)_1 \equiv \begin{cases} \text{Assumption [Bounded communication delays]} \\ \text{Assumption [Convex combination and threshold]} \\ \text{Assumption [Graph connectivity]} \\ \text{Assumption [Bounded communication intervals]} \end{cases}$$

$$(AsY)_2 \equiv \begin{cases} \text{Assumption [Bounded communication delays]} \\ \text{Assumption [Convex combination and threshold]} \\ \text{Assumption [Graph connectivity]} \\ \text{Assumption [Symmetry]} \end{cases}$$

Agreement theorem.

Agreement algorithm.

$$x^i(t+1) = \sum_{j=1}^{M} a^{i,j}(t) x^j(\tau^{i,j}(t)),$$

Theorem (Blondel et al. [5])

Under assumptions $(AsY)_1$ or $(AsY)_2$ there is a vector $c^\star \in \left(\mathbb{R}^d\right)^\kappa$ (independent of i) s.t.,

$$\lim_{t\to\infty} \left\| x^i(t) - c^\star \right\| = 0.$$

Even more, there exists $\rho \in [0,1)$ and $L > 0$, s.t.,

$$\left\| x^i(t) - x^i(\tau) \right\| \leq L\rho^{t-\tau},$$

Agreement vector.

The previous theorem is useful for the study of (GDAS):

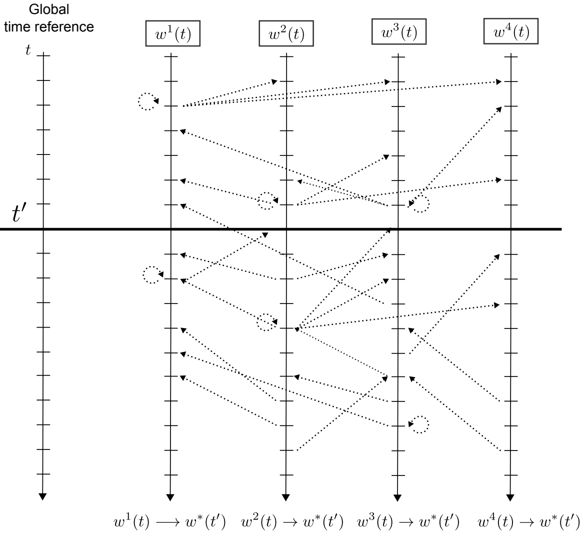
$$w^i(t+1) = \sum_{j=1}^{M} a^{i,j}(t) w^j(\tau^{i,j}(t)) + s^i(t).$$

For any $t' \geq 0$, if computations with descent terms have stopped after t', i.e, $s^i(t) = 0$ for all $t \geq t'$ and all i.

$$w^i(t) \xrightarrow[t\to\infty]{} w^\star(t') \quad \text{for all } i \in \{1, \dots, M\}.$$

Averaging and computation with descent terms (general distributed asynchronous algorithm).

Only averaging (agreement algorithm).



Agreement vector sequence.

Agreement vector sequence: $\{w^\star(t)\}_{t=0}^{\infty}$.
The true definition is more complex.

Remark:
The agreement vector $w^\star$ satisfies, for all $t \geq 0$,

$$w^\star(t+1) = w^\star(t) + \sum_{j=1}^{M} \phi^j(t) s^j(t), \tag{1}$$

$\phi^j(t) \in [0,1]$.

Although the agreement vector sequence is unknown it will be a useful tool for the convergence analysis of (GDAS).

Agreement vector sequence.

Agreement vector sequence: $\{w^\star(t)\}_{t=0}^{\infty}$.
The true definition is more complex.

Remark:
The agreement vector $w^\star$ satisfies, for all $t \geq 0$,

$$w^\star(t+1) = w^\star(t) + \sum_{j=1}^{M} \phi^j(t) s^j(t), \quad (1)$$

$\phi^j(t) \in [0, 1]$.

Although the agreement vector sequence is unknown it will be a useful tool for the convergence analysis of (GDAS).

Agreement vector sequence.

Agreement vector sequence: $\{w^\star(t)\}_{t=0}^{\infty}$.
The true definition is more complex.

Remark:
The agreement vector $w^\star$ satisfies, for all $t \geq 0$,

$$w^\star(t+1) = w^\star(t) + \sum_{j=1}^{M} \phi^j(t) s^j(t), \tag{1}$$

$\phi^j(t) \in [0, 1]$.

Although the agreement vector sequence is unknown it will be a useful tool for the convergence analysis of (GDAS).

Distributed Asynchronous Learning Vector Quantization (DALVQ).

(GDAS) with the descent terms s^i,

$$s^i(t) = \begin{cases} -\varepsilon_t^i H\left(\mathbf{z}_{t+1}^i, w^i(t)\right) & \text{if } t \in T^i \\ 0 & \text{otherwise.} \end{cases}$$

Notation:

- T^i: set of time instant where the version $\left\{w^i(t)\right\}_{t=0}^{\infty}$ is updated with descent terms. T^i deterministic but do not need to be known a priori for the execution.
- $\left\{\mathbf{z}_t^i\right\}_{t=0}^{\infty}$ iid sequences of r.v. of law μ.
- $\mathcal{F}_t \triangleq \sigma\left(\mathbf{z}_s^i, \text{ for all } s \leq t \text{ and } 1 \leq i \leq M\right)$.

Distributed Asynchronous Learning Vector Quantization (DALVQ).

(GDAS) with the descent terms s^i,

$$s^i(t) = \begin{cases} -\varepsilon_t^i H\left(\mathbf{z}_{t+1}^i, w^i(t)\right) & \text{if } t \in T^i \\ 0 & \text{otherwise.} \end{cases}$$

Notation:

- T^i: set of time instant where the version $\left\{w^i(t)\right\}_{t=0}^{\infty}$ is updated with descent terms. T^i deterministic but do not need to be known a priori for the execution.
- $\left\{\mathbf{z}_t^i\right\}_{t=0}^{\infty}$ iid sequences of r.v. of law μ.
- $\mathcal{F}_t \triangleq \sigma\left(\mathbf{z}_s^i, \text{ for all } s \leq t \text{ and } 1 \leq i \leq M\right)$.

DALVQ.

DALVQ

To be continued.

We assume that the sequences ε_t^i satisfy the following conditions:

Assumption (Decreasing steps)

There exist two constants K_1, K_2 s.t., for all i and all $t \geq 0$,

$$\frac{K_1}{t \vee 1} \leq \varepsilon_t^i \leq \frac{K_2}{t \vee 1}.$$

Assumption (Non idle)

$\sum_{j=1}^{M} \mathbb{1}_{t \in T^j} > 0$

To be continued.

We assume that the sequences ε_t^i satisfy the following conditions:

Assumption (Decreasing steps)

There exist two constants K_1, K_2 s.t., for all i and all $t \geq 0$,

$$\frac{K_1}{t \vee 1} \leq \varepsilon_t^i \leq \frac{K_2}{t \vee 1}.$$

Assumption (Non idle)

$\sum_{j=1}^{M} \mathbb{1}_{t \in T^j} > 0$

The recursion formula of agreement vector writes,

$$w^\star(t+1) = w^\star(t) + \sum_{j=1}^{M} \phi^j(t) s^j(t).$$

Using the function h,

$$h(w^\star(t)) = \mathbb{E}\left\{ H\left(\mathbf{z}_{t+1}, w^\star(t)\right) \mid \mathcal{F}_t \right\}$$

and

$$h(w^j(t)) = \mathbb{E}\left\{ H\left(\mathbf{z}_{t+1}, w^j(t)\right) \mid \mathcal{F}_t \right\}, \qquad \text{for all j.}$$

Set,

$$\varepsilon_t^\star \triangleq \frac{1}{2}\sum_{j=1}^{M} \mathbb{1}_{t\in T^j}\phi^j(t)\varepsilon_t^j$$

and

$$\Delta M_t^1 \triangleq \frac{1}{2}\sum_{j=1}^{M} \mathbb{1}_{t\in T^j}\phi^j(t)\varepsilon_t^j \left(h(w^\star(t)) - h(w^j(t))\right),$$

and,

$$\Delta M_t^2 \triangleq \frac{1}{2}\sum_{j=1}^{M} \mathbb{1}_{t\in T^j}\phi^j(t)\varepsilon_t^j \left(h(w^j(t)) - H\left(\mathbf{z}_{t+1}^j, w^j(t)\right)\right).$$

The recursion (1) writes,

$$w^\star(t+1) = w^\star(t) - \varepsilon_t^\star h(w^\star(t)) + \Delta M_t^1 + \Delta M_t^2.$$

Set,

$$\varepsilon_t^\star \triangleq \frac{1}{2}\sum_{j=1}^{M} \mathbb{1}_{t\in T^j}\phi^j(t)\varepsilon_t^j$$

and

$$\Delta M_t^1 \triangleq \frac{1}{2}\sum_{j=1}^{M} \mathbb{1}_{t\in T^j}\phi^j(t)\varepsilon_t^j \left(h(w^\star(t)) - h(w^j(t))\right),$$

and,

$$\Delta M_t^2 \triangleq \frac{1}{2}\sum_{j=1}^{M} \mathbb{1}_{t\in T^j}\phi^j(t)\varepsilon_t^j \left(h(w^j(t)) - H\left(\mathbf{z}_{t+1}^j, w^j(t)\right)\right).$$

The recursion (1) writes,

$$w^\star(t+1) = w^\star(t) - \varepsilon_t^\star h(w^\star(t)) + \Delta M_t^1 + \Delta M_t^2.$$

Theorem (Asynchronous G-Lemma)

Assume that one has,

1. $\sum_{t=0}^{\infty} \varepsilon_t^\star = \infty$ *and* $\varepsilon_t^\star \xrightarrow[t\to\infty]{} 0$.
2. *The sequences* $\{w^\star(t)\}_{t=0}^{\infty}$ *and* $\{h(w^\star(t))\}_{t=0}^{\infty}$ *are bounded with probability 1.*
3. *The series* $\sum_{t=0}^{\infty} \Delta M_t^{(1)}$ *and* $\sum_{t=0}^{\infty} \Delta M_t^{(2)}$ *converge a.s. in* $\left(\mathbb{R}^d\right)^\kappa$.
4. *There exists a l.s.c. map* $G : \left(\mathbb{R}^d\right)^\kappa \longrightarrow \mathbb{R}_+$, *s.t.*

$$\sum_{t=0}^{\infty} \varepsilon_{t+1}^\star G\left(w^\star(t)\right) < \infty, \quad a.s..$$

Then there exists a connected component Ξ *of* $\{G = 0\}$ *s.t.*

$$\lim_{t\to\infty} \operatorname{dist}\left(w^\star(t), \Xi\right) = 0, \quad a.s..$$

$$\widehat{G}(w) \triangleq \liminf_{v \in \mathcal{G}^{\kappa} \cap \mathcal{D}_*^{\kappa}, v \to w} \|\nabla C(v)\|^2 .$$

Assumption (Trajectories in $\mathcal{G}^{\kappa}$)

$$\mathbb{P}\left\{ w^j(t) \in \mathcal{G}^{\kappa} \right\} = 1, \ \forall j \ \forall t \geq 0.$$

$$\widehat{G}(w) \triangleq \liminf_{v \in \mathcal{G}^{\kappa} \cap \mathcal{D}_*^{\kappa}, v \to w} \|\nabla C(v)\|^2 .$$

Assumption (Trajectories in $\mathcal{G}^{\kappa}$)

$$\mathbb{P}\left\{ w^j(t) \in \mathcal{G}^{\kappa} \right\} = 1, \ \forall j \ \forall t \geq 0.$$

Lemma

Assume that Assumptions [Decreasing steps] and [Trajectories in $\mathcal{G}^{\kappa}$] are satisfied. Then, for all $t \geq 0$

$$\|w^{\star}(t) - w^{i}(t)\| \leq \sqrt{\kappa} M \operatorname{diam}(\mathcal{G}) A K_2 \theta_t, \quad a.s.,$$

where $\theta_t \triangleq \sum_{\tau=-1}^{t-1} \frac{1}{\tau \vee 1} \rho^{t-\tau}$.

Remark

Under assumptions [Trajectories in $\mathcal{G}^{\kappa}$],

1. $w^{\star}(t) - w^{i}(t) \xrightarrow{a.s.} 0$ *as* $t \to \infty$,
2. $w^{j}(t) - w^{i}(t) \xrightarrow{a.s.} 0$ *as* $t \to \infty$ *for* $i \neq j$.

Lemma

Assume that Assumptions [Decreasing steps] and [Trajectories in $\mathcal{G}^\kappa$] are satisfied. Then, for all $t \geq 0$

$$\|w^\star(t) - w^i(t)\| \leq \sqrt{\kappa} M \operatorname{diam}(\mathcal{G}) A K_2 \theta_t, \quad a.s.,$$

where $\theta_t \triangleq \sum_{\tau=-1}^{t-1} \frac{1}{\tau \vee 1} \rho^{t-\tau}$.

Remark

Under assumptions [Trajectories in $\mathcal{G}^\kappa$],

1. $w^\star(t) - w^i(t) \xrightarrow{a.s.} 0$ *as* $t \to \infty$,
2. $w^j(t) - w^i(t) \xrightarrow{a.s.} 0$ *as* $t \to \infty$ *for* $i \neq j$.

The Asynchronous Theorem.

Assumption (Parted component assumption)

1. $\mathbb{P}\left\{w^\star(t) \in \mathcal{D}_*^\kappa\right\} = 1$, *for all* $t \geq 0$.
2. $\mathbb{P}\left\{\liminf_{t\to\infty} \operatorname{dist}\left(w^\star(t), \complement\mathcal{D}_*^\kappa\right)\right\} = 1$.

Theorem (Asynchronous Theorem)

If assumptions [Trajectories in $\mathcal{G}^\kappa$], [Parted component assumption] hold then,

$$w^\star(t) \xrightarrow[t\to\infty]{a.s.} \Xi_\infty,$$

and,

$$w^i(t) \xrightarrow[t\to\infty]{a.s.} \Xi_\infty, \quad \text{for all processors } i.$$

Where Ξ_∞ is some connected component of $\{\nabla C = 0\}$.

The Asynchronous Theorem.

Assumption (Parted component assumption)

1. $\mathbb{P}\left\{w^{\star}(t) \in \mathcal{D}_{*}^{\kappa}\right\} = 1$, *for all* $t \geq 0$.
2. $\mathbb{P}\left\{\liminf_{t\to\infty} \operatorname{dist}\left(w^{\star}(t), \complement\mathcal{D}_{*}^{\kappa}\right)\right\} = 1$.

Theorem (Asynchronous Theorem)

If assumptions [Trajectories in $\mathcal{G}^{\kappa}$], [Parted component assumption] hold then,

$$w^{\star}(t) \xrightarrow[t\to\infty]{a.s.} \Xi_{\infty},$$

and,

$$w^{i}(t) \xrightarrow[t\to\infty]{a.s.} \Xi_{\infty}, \quad \text{for all processors } i.$$

Where Ξ_{∞} is some connected component of $\{\nabla C = 0\}$.

Bibliography

- G. Pagès, “A space vector quantization for numerical integration,” *Journal of Applied and Computational Mathematics*, vol. 89, 1997.
- J.-C. Fort and G. Pagès, “Convergence of stochastic algorithms: From the kushner-clark theorem to the lyapounov functional method,” *Advances in Applied Probability*, vol. 28, 1996.
- J. Tsitsiklis, D. Bertsekas, and M. Athans, “Distributed asynchronous deterministic and stochastic gradient optimization algorithms,” *Automatic Control, IEEE Transactions on*, vol. 31, 1986.
- D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and distributed computation: numerical methods*. USA: Prentice-Hall, Inc., 1989.
- V. Blondel, J. Hendrickx, A. Olshevsky, and J. Tsitsiklis, “Convergence in multiagent coordination, consensus, and flocking,” *Decision and Control, 2005 and 2005 European Control Conference*, 2005.

D. Pollard, "Quantization and the method of k-means," *IEEE Transactions on Information Theory*, 1982.

——, "Stong consistency of k-means clustering," *The annals of statistics*, vol. 9, 1981.

E. A. Abaya and G. L. Wise, "Convergence of vector quantizers with applications to optimal quantization," *SIAM Journal on Applied Mathematics*, 1984.

D. Pollard, "A central limit theorem for k-means clustering," *The annals of probability*, vol. 28, 1982.

P. A. Chou, "The distortion of vector quantizers trained on n vectors decreases to the optimum at $o_p(1/n)$," *IEEE transactions on information theory*, vol. 8, 1994.

Z. K. Linder T., Lugosi G., "Rates of convergence in the source coding theorem, in empirical quantizer design, and in universal lossy source coding," *IEEE Transactions on Information Theory*, vol. 40.

L. G. Bartlett P.L., Linder T., "The minimax distortion redundancy in empirical quantizer design," *IEEE Transactions on Information Theory*, vol. 44, 1998.

M. Inaba, N. Katoh, and H. Imai, "Applications of weighted voronoi diagrams and randomization to variance-based k-clustering," in *SCG '94: Proceedings of the tenth annual symposium on Computational geometry*. USA: ACM, 1994.

L. Thomas, "On the training distortion of vector quantizers," *IEEE Transactions on Information Theory*, vol. 46, 2000.