Block Coordinate and Stochastic Gradient Decent Methods

Yinyu Ye

Department of Management Science and Engineering

Stanford University

Stanford, CA 94305, U.S.A.

http://www.stanford.edu/~yyye

(Chapter 8)

Recall Generic Algorithms for Minimization and Global Convergence Theorem

A Generic Algorithm: A point to set mapping in a subspace of R^n.

Theorem 1 *Let A be an "algorithmic mapping" defined over set X, and let sequence $\{\mathbf{x}^k\}$, starting from a given point $\mathbf{x}^0$, be generated from*

$$\mathbf{x}^{k+1} \in A(\mathbf{x}^k).$$

Let a solution set $S \subset X$ be given, and suppose

i) all points $\{\mathbf{x}^k\}$ are in a compact set;

ii) there is a continuous (merit) function $z(\mathbf{x})$ such that if $\mathbf{x} \notin S$, then $z(\mathbf{y}) < z(\mathbf{x})$ for all $\mathbf{y} \in A(\mathbf{x})$; otherwise, $z(\mathbf{y}) \leq z(\mathbf{x})$ for all $\mathbf{y} \in A(\mathbf{x})$;

iii) the mapping A is closed at points outside S.

Then, the limit of any convergent subsequences of $\{\mathbf{x}^k\}$ is a solution in S.

Block Coordinate Descent Method for Unconstrained Optimization I

$$\min_{\mathbf{x}\in R^N} \quad f(\mathbf{x}) = f((\mathbf{x}_1;\ \mathbf{x}_2,\ ...;\ \mathbf{x}_n)), \quad \text{where } \mathbf{x} = (\mathbf{x}_1;\ \mathbf{x}_2;\ ...;\ \mathbf{x}_n).$$

For presentation simplicity, we let each $\mathbf{x}_j$ be a scalar variable so that $N = n$.

Let $f(\mathbf{x})$ be differentiable every where and satisfy the (first-order) β-Coordinate Lipschitz condition, that is, for any two vectors $\mathbf{x}$ and $\mathbf{d}$

$$\|\nabla_j f(\mathbf{x} + \mathbf{e}_j .* \mathbf{d}) - \nabla_j f(\mathbf{x})\| \leq \beta_j \|\mathbf{e}_j .* \mathbf{d}\| \tag{1}$$

where $\mathbf{e}_j$ is the unit vector that $e_j = 1$ and zero everywhere else, and $.*$ is the component-wise product.

Cyclic Block Coordinate Descent (CBCD) Method (Gauss-Seidel):

$$\begin{aligned} \mathbf{x}_1 &\longleftarrow \arg\min_{\mathbf{x}_1} f(\mathbf{x}_1, \ldots, \mathbf{x}_n), \\ &\vdots \\ \mathbf{x}_n &\longleftarrow \arg\min_{\mathbf{x}_n} f(\mathbf{x}_1, \ldots, \mathbf{x}_n). \end{aligned}$$

Aitken Double Sweep Method:

$$\begin{aligned}
\mathbf{x}_1 &\longleftarrow \arg\min_{\mathbf{x}_1} f(\mathbf{x}_1, \ldots, \mathbf{x}_n), \\
&\vdots \\
\mathbf{x}_n &\longleftarrow \arg\min_{\mathbf{x}_n} f(\mathbf{x}_1, \ldots, \mathbf{x}_n), \\
\mathbf{x}_{n-1} &\longleftarrow \arg\min_{\mathbf{x}_{n-1}} f(\mathbf{x}_1, \ldots, \mathbf{x}_n), \\
&\vdots \\
\mathbf{x}_2 &\longleftarrow \arg\min_{\mathbf{x}_2} f(\mathbf{x}_1, \ldots, \mathbf{x}_n).
\end{aligned}$$

Gauss-Southwell Method:

- Compute the gradient vector $\nabla f(\mathbf{x})$ and let $i^* = \arg\max\{|\nabla f(\mathbf{x})_j|\}$.
-

$$\mathbf{x}_{i^*} \longleftarrow \arg\min_{\mathbf{x}_i} f(\mathbf{x}_1, \ldots, \mathbf{x}_n).$$

Block Coordinate Descent Method for Unconstrained Optimization II

Randomly-Permuted Cyclic Block Coordinate Descent (RCBCD) Method:

- Draw a random permutation $\sigma = \{\sigma(1), \ldots, \sigma(n)\}$ of $\{1, \ldots, n\}$;
- $$\begin{aligned} \mathbf{x}_{\sigma(1)} &\longleftarrow \arg\min_{\mathbf{x}_{\sigma(1)}} f(\mathbf{x}_1, \ldots, \mathbf{x}_n), \\ &\vdots \\ \mathbf{x}_{\sigma(n)} &\longleftarrow \arg\min_{\mathbf{x}_{\sigma(n)}} f(\mathbf{x}_1, \ldots, \mathbf{x}_n). \end{aligned}$$

Randomized Block Coordinate Descent (RBCD) Method:

- Randomly choose $i^* \in \{1, 2, ..., n\}$.
- $$\mathbf{x}_{i^*} \longleftarrow \arg\min_{\mathbf{x}_{i^*}} f(\mathbf{x}_1, \ldots, \mathbf{x}_n).$$

Convergence of the BCD Methods

The following theorem gives some conditions under which the deterministic BCD method will generate a sequence of iterates that converge.

Theorem 2 *Let $f: R^n \to R$ be given. For some given point $x^0 \in R^n$, let the level set*

$$X^0 = \{\mathbf{x} \in R^n : f(\mathbf{x}) \leq f(\mathbf{x}^0)\}$$

be bounded. Assume further that f is continuously differentiable on the convex hull of X^0. Let $\{\mathbf{x}^k\}$ be the sequence of points generated by the Cyclic Block Coordinate Descent Method initiated at $\mathbf{x}^0$. Then every accumulation point of $\{\mathbf{x}^k\}$ is a stationary point of f.

For strictly convex quadratic minimization with Hessian Q, e.g., the linear convergence rate of Gauss-Southwell is

$$\left(1 - \frac{\lambda_{min}(Q)}{\lambda_{max}(Q)(n-1)}\right)^{n-1} \geq 1 - \frac{\lambda_{min}(Q)}{\lambda_{max}(Q)} \geq \left(\frac{\lambda_{max}(Q) - \lambda_{min}(Q)}{\lambda_{max}(Q) + \lambda_{min}(Q)}\right)^2.$$

Randomized Block Coordinate Gradient Descent Method

At the kth Iteration of RBCGD:

- Randomly choose $i^k \in \{1, 2, ..., n\}$.
- $$\mathbf{x}_{i^k}^{k+1} = \mathbf{x}_{i^k}^k - \frac{1}{\beta_{i^k}} \nabla_{i^k} f(\mathbf{x}^k),$$
$$\mathbf{x}_i^{k+1} = \mathbf{x}_i^k, \ \forall i \neq i^k.$$

Theorem 3 *(Expected Error Convergence Estimate Theorem) Let the objective function $f(\mathbf{x})$ be convex and satisfy the (first-order) β-Coordinate Lipschitz condition, and admit a minimizer $\mathbf{x}^*$. Then*

$$E_{\xi^k}[f(\mathbf{x}^{k+1})] - f(\mathbf{x}^*) \leq \frac{n}{n+k+1} \left(\frac{1}{2} \|\mathbf{x}^0 - \mathbf{x}^*\|_\beta^2 + f(\mathbf{x}^0) - f(\mathbf{x}^*) \right),$$

where random vector $\xi_{k-1} = (i^0, i^1, ..., i^{k-1})$ and norm-square $\|\mathbf{x}\|_\beta^2 = \sum_j \beta_j x_j^2$.

Proof: Denote by $\delta^k = f(\mathbf{x}^k) - f(\mathbf{x}^*)$, $\Delta^k = \mathbf{x}^k - \mathbf{x}^*$, and

$$(r^k)^2 = \|\mathbf{x}^k - \mathbf{x}^*\|_\beta^2 = \sum_j \beta_j (x_j^0 - x_j^*)^2.$$

Then, from the RBCGD iteration

$$(r^{k+1})^2 = (r^k)^2 - 2\nabla_{i^k} f(\mathbf{x}^k)(x_{i^k}^k - x_{i^k}^*) + \frac{1}{\beta_{i^k}} (\nabla_{i^k} f(\mathbf{x}^k))^2.$$

It follows from the β-Coordinate Lipschitz condition,

$$\begin{aligned} f(\mathbf{x}^{k+1}) - f(\mathbf{x}^k) &\leq \nabla_{i^k} f(\mathbf{x}^k)(x_{i^k}^{k+1} - x_{i^k}^k) + \tfrac{1}{2\beta_{i^k}} (\nabla_{i^k} f(\mathbf{x}^k))^2 \\ &= \tfrac{-1}{2\beta_{i^k}} (\nabla_{i^k} f(\mathbf{x}^k))^2. \end{aligned}$$

Combining the two inequalities, we have

$$(r^{k+1})^2 \leq (r^k)^2 - 2\nabla_{i^k} f(\mathbf{x}^k)(x_{i^k}^k - x_{i^k}^*) + 2(f(\mathbf{x}^k) - f(\mathbf{x}^{k+1})).$$

Dividing both sides by 2 and taking expectation with respect to i^k yields

$$E_{i^k}[\frac{1}{2}(r^{k+1})^2] \leq \frac{1}{2}(r^k)^2 - \frac{1}{n}\nabla f(\mathbf{x}^k)^T(\mathbf{x}^k - \mathbf{x}^*) + f(\mathbf{x}^k) - E_{i^k}[f(\mathbf{x}^{k+1})],$$

which together with convexity assumption $\nabla f(\mathbf{x}^k)^T(\mathbf{x}^* - \mathbf{x}^k) \leq f(\mathbf{x}^*) - f(\mathbf{x}^k)$ gives

$$E_{i^k}[\frac{1}{2}(r^{k+1})^2] \leq \frac{1}{2}(r^k)^2 + \frac{1}{n}f(\mathbf{x}^*) + \frac{n-1}{n}f(\mathbf{x}^k) - E_{i^k}[f(\mathbf{x}^{k+1})],$$

Rearranging gives, for each $k \geq 0$,

$$E_{i^k}[\frac{1}{2}(r^{k+1})^2 + \delta^{k+1}] \leq \left(\frac{1}{2}(r^k)^2 + \delta^k\right) - \frac{1}{n}\delta^k.$$

Taking expectation with respect to ξ^{k-1} on both sides

$$\begin{aligned} E_{\xi^k}[\tfrac{1}{2}(r^{k+1})^2 + \delta^{k+1}] &\leq E_{\xi^{k-1}}[\tfrac{1}{2}(r^k)^2 + \delta^k] - \tfrac{1}{n}E_{\xi^{k-1}}[\delta^k] \\ &= E_{\xi^k}[\tfrac{1}{2}(r^k)^2 + \delta^k] - \tfrac{1}{n}E_{\xi^k}[\delta^k]. \end{aligned}$$

Recursively applying the inequalities from and noting that $E_{\xi^k}[f(\mathbf{x}^{k+1})]$ is monotonically decreasing

$$\begin{aligned} E_{\xi^k}[\delta^{k+1}] &\leq E_{\xi^k}[\tfrac{1}{2}(r^{k+1})^2 + \delta^{k+1}] \\ &\leq (\tfrac{1}{2}(r^0)^2 + \delta^0) - \tfrac{1}{n}\textstyle\sum_{j=0}^{k} E_{\xi^k}[\delta^j] \\ &\leq (\tfrac{1}{2}(r^0)^2 + \delta^0) - \tfrac{k+1}{n}E_{\xi^k}[\delta^{k+1}] \end{aligned}$$

which leads to the desired result.

Worst-Case Convergnece Comparison of BCDs

There is a convex quadratic minimization problem of dimension n:

$$\min \quad \mathbf{x}^T Q \mathbf{x}, \quad \text{where for } \gamma \in (0,\ 1)$$

$$Q = \begin{pmatrix} 1 & \gamma & ... & \gamma \\ \gamma & 1 & ... & \gamma \\ ... & ... & ... & ... \\ \gamma & \gamma & ... & 1 \end{pmatrix}.$$

- CBCD is $\frac{n}{2\pi^2}$ times slower than SDM;
- CBCD is $\frac{n^2}{2\pi^2}$ times slower than RBCD (each iteratione randomly select n corordinate to update);
- CBCD is $\frac{n(n+1)}{2\pi^2}$ times slower than RCBCD;

Randomization makes a difference.

Recall Optimization under Uncertainty

In many applications, the objective value is partially determined by decision makers and partially determined by "Nature".

$$\begin{array}{lll} (OPT) & \min_{\mathbf{x}} & f(\mathbf{x}, \omega) \\ & \text{s.t.} & \mathbf{c}(\mathbf{x}, \omega) \in K \subset R^m. \end{array} \tag{2}$$

where ω represents uncertain data and $\mathbf{x} \in R^n$ is the decision vector, and K is a constraint set.

For deterministic optimization, we assume ξ is known and fixed. In reality, we may have

- the (exact) probability distribution ξ of data ω.
- the sample distribution and/or few moments of data ω.
- knowledge of ω belonging to a given uncertain set U.

In the following we consider the unconstrained case.

Stochastic Optimization and Stochastic Gradient Descent (SGD) Methods

$$\min_{\mathbf{x}} \quad F(\mathbf{x}) := \mathsf{E}_{\xi}[f(\mathbf{x}, \omega)].$$

Sample Average Approximation (SAA) Method:

$$\min_{\mathbf{x}} \quad \frac{1}{M} \sum_{i=1}^{M} f(\mathbf{x}, \omega^i).$$

Two Implementations:

- Sample-First and Optimize-Second, in particular, SAA: collect enough examples then search a solution of an approximated deterministic optimization problem.
- Sample and Optimize concurrently, in particular, SGD: collect a sample set S^k of few samples of ω at iteration k:

$$\hat{\mathbf{g}}^k = \frac{1}{|S^k|} \sum_{i \in S^k} \nabla f(\mathbf{x}^k, \omega^i) \quad \text{and} \quad \mathbf{x}^{k+1} = \mathbf{x}^k - \alpha^k \hat{\mathbf{g}}^k.$$

Questions are: how many samples are sufficient for an ϵ approximate solution to the original stochastic optimization problem. This is an information complexity issue, besides the computation complexity issue, in optimization.

Information Complexity and Sample Size in SAA

- In SAA, the required number of samples, M, should be larger than the dimension of decision vector and should grow polynomially with the increase of dimensionality. In specific, let $\mathbf{x}^{SAA}$ be the optimal solution from the SAA method. Then to ensure probability

$$P[F(\mathbf{x}^{SAA}) - F(\mathbf{x}^*) \leq \epsilon] \geq 1 - \alpha,$$

$$M = O(\frac{1}{\epsilon^2})(n \ln(\frac{1}{\epsilon}) + \ln(\frac{1}{\alpha})).$$

- If $\mathbf{x}^*$ is sparse or it can be approximated by a sparse solution with cardinality $p << n$, then by adding a regulative penalty function into the objective

$$\min_{\mathbf{x}} \quad \frac{1}{M} \sum_{i=1}^{M} f(\mathbf{x}, \omega^i) + P(\mathbf{x}),$$

the sample size can be reduced to

$$M = O(\frac{1}{\epsilon^2})(\frac{p}{\epsilon} \ln^{1.5}(\frac{n}{\epsilon}) + \ln(\frac{1}{\alpha})); \quad \text{or in convex case: } M = O(\frac{1}{\epsilon^2})(p \ln(\frac{n}{\epsilon}) + \ln(\frac{1}{\alpha})).$$

SGD and its Advantages

Collect SGD with one ω^k sampled uniformly at iteration k:

$$\hat{\mathbf{g}}^k = \nabla f(\mathbf{x}^k, \omega^k) \quad \text{and} \quad \mathbf{x}^{k+1} = \mathbf{x}^k - \alpha^k \hat{\mathbf{g}}^k.$$

- Works with the step size rule:

$$\alpha^k \to 0 \quad \text{and} \quad \left(\sum_{k=0}^{\infty} \alpha^k\right) \to \infty \quad \text{(e.g., } \alpha_k = O(k^{-1})\text{)}.$$

- A great technology to potentially reduce the computation complexity – need fewer samples at the beginning.
- Potentially only select important and sensitive samples – learn where to sample.
- Dynamically incorporate new empirical observations to tune-up the probability distribution.

Variance Reduction in Stochastic Algorithm Design

- The VR technique has been used extensively in the design of fast stochastic methods for solving large–scale optimization problems in machine learning.
- High Level Idea: Reduce the variance of an estimate X by using another estimate Y with known expectation.
- Specifically, consider $Z_\alpha = \alpha(X - Y) + \mathsf{E}[Y]$.
  - $\mathsf{E}[Z_\alpha] = \alpha \cdot \mathsf{E}[X] + (1 - \alpha) \cdot \mathsf{E}[Y]$
  - $\mathrm{var}(Z_\alpha) = \mathsf{E}\left[(Z_\alpha - \mathsf{E}[Z_\alpha])^2\right] = \alpha^2 \left[\mathrm{var}(X) + \mathrm{var}(Y) - 2\mathrm{cov}(X, Y)\right]$
  - When $\alpha = 1$, we have $\mathsf{E}[Z_\alpha] = \mathsf{E}[X]$, which is useful for establishing concentration bounds.
  - When $\alpha < 1$, Z_α will potentially have a smaller variance than X, but we no longer have $\mathsf{E}[Z_\alpha] = \mathsf{E}[X]$. (In what follows, we let $\alpha = 1$.)
  - Overall, variance reduction occur if $\mathrm{cov}(X, Y) > 0$.

VR Illustration: Finite–Sum Minimization I

- Consider the following so–called finite–sum minimization problem:

$$\min_{\mathbf{x}} \left\{ F(\mathbf{x}) = \frac{1}{M} \sum_{i=1}^{M} f_i(\mathbf{x}) \right\}. \tag{3}$$

  Here, $f_1, \ldots, f_M$ are smooth (convex) loss functions and M is huge so that the computation of $\nabla F(\cdot)$ is costly.

- Examples
  - Linear regression: $f_i(\mathbf{x}) = (\mathbf{a}_i^T \mathbf{x} - b_i)^2$
  - Logistic regression: $f_i(\mathbf{x}) = \ln\left(1 + \exp\left(b_i \mathbf{a}_i^T \mathbf{x}\right)\right)$

- Stochastic Gradient Descent (SGD): choose i_k from $\{1, ..., M\}$ uniformly at random and let

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \alpha^k \nabla f_{i_k}(\mathbf{x}^k).$$

  - We have $\mathsf{E}\left[\nabla f_{i_k}(\mathbf{x}^k)\right] = \nabla F(\mathbf{x}^k)$, but variance of the estimate can be large.
  - To guarantee convergence, we generally need diminishing step sizes (e.g., $\alpha_k = O(k^{-1})$).

VR Illustration: Finite–Sum Minimization II

- Now let $X = \nabla f_{i_k}(\mathbf{x}^k)$ for estimating $\nabla F(\mathbf{x}^k)$. What Y should we use to reduce the variance of the estimate?
  - Try $Y = \nabla f_{i_k}(\tilde{\mathbf{x}}^k)$ for some fixed $\tilde{\mathbf{x}}^k$.
  - Note that $\mathsf{E}[Y] = \nabla F(\tilde{\mathbf{x}}^k)$.
- Now, form $Z = X - Y + \mathsf{E}[Y] = \nabla f_{i_k}(\mathbf{x}^k) - \nabla f_{i_k}(\tilde{\mathbf{x}}^k) + \nabla F(\tilde{\mathbf{x}}^k)$ and set

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \alpha_k \left(\nabla f_{i_k}(\mathbf{x}^k) - \nabla f_{i_k}(\tilde{\mathbf{x}}^k) + \nabla F(\tilde{\mathbf{x}}^k) \right).$$

- Since the computation of $\nabla F(\tilde{\mathbf{x}}^k)$ is costly, we don't want to update $\tilde{x}^k$ too often but only once for a while.
- This is the core idea behind Johnson and Zhang's stochastic variance–reduced gradient (SVRG) method, which has generated much recent research; see Johnson, Zhang. *Accelerating Stochastic Gradient Descent Using Predictive Variance Reduction*, NIPS 2013.

VR Illustration: Finite–Sum Minimization III

- One choice is to update $\tilde{\mathbf{x}}$ at a uniform (or geometric) pace, that is, when $k = rK$ (or $k = 2^r$) for a nonnegative integer r, we let $\tilde{\mathbf{x}}^k = \mathbf{x}^k$ and it remains unchanged from iteration k to $k + K$ (or $2k$).
- Thus, from iteration 1 to k, $\tilde{\mathbf{x}}^k$ is updated, or $\nabla F(\tilde{\mathbf{x}}^k)$ is computed, only k/K (or $\log(k)$) times.
- Moreover, most likely $\mathrm{cov}(\mathbf{x}^k, \tilde{\mathbf{x}}^k) > 0$ during the iteration period k to $k + K$, since both $\mathbf{x}^k$ and $\tilde{\mathbf{x}}^k$ converge to the same limit solution.

VR Illustration: Finite–Sum Minimization IV

- The VR-SGD method can be shown to converge linearly when F satisfies the so–called error bound condition: there exists a $\tau > 0$ such that

$$\mathrm{dist}(\mathbf{x}, \mathcal{X}^*) \leq \tau \|\nabla F(\mathbf{x})\|_2 \quad \text{for all } \mathbf{x}, \tag{4}$$

  where $\mathcal{X}^*$ is the set of optimal solutions.

- If F is strongly convex, then it satisfies 4. However, the converse need not hold; for details, see So, Zhou. *Non-Asymptotic Convergence Analysis of Inexact Gradient Methods for Machine Learning Without Strong Convexity*. Optim. Methods Softw. 32(4): 963–992, 2017.

- Extensions of the VR-SGD method to the case where F is non–convex have been proposed and analyzed in Reddi, Hefny, Sra, Póczós, Smola. *Stochastic Variance Reduction for Nonconvex Optimization*. ICML 2016, and Allen–Zhu, Hazan. *Variance Reduction for Faster Nonconvex Optimization*. ICML 2016.

Variance Reduction in Stochastic Value Iteration for MDP I

Let $\mathbf{y} \in \mathbf{R}^m$ represent the cost-to-go values of the m states, ith entry for ith state, of a given policy. The MDP problem entails choosing the fixed-point value vector $\mathbf{y}^*$ such that it satisfies:

$$y_i^* = \min_{j \in \mathcal{A}_i} \{c_j + \gamma \mathbf{p}_j^T \mathbf{y}^*\}, \ \forall i.$$

The Value-Iteration (VI) Method is, starting from any $\mathbf{y}^0$,

$$y_i^{k+1} = \min_{j \in \mathcal{A}_i} \{c_j + \gamma \mathbf{p}_j^T \mathbf{y}^k\}, \ \forall i.$$

If the initial $\mathbf{y}^0$ is strictly feasible for state i, that is, $y_i^0 < c_j + \gamma \mathbf{p}_j^T \mathbf{y}^0, \ \forall j \in \mathcal{A}_i$, then y_i^k would be increasing in the VI iteration for all i and k.

The computation work for state i at iteration k, is to compute $\mathbf{p}_j^T \mathbf{y}^k = \mu_j(\mathbf{y}^k)$ for each $j \in \mathcal{A}_i$. This needs $O(m)$ operations.

Could we approximate $\mu_j(\mathbf{y}^k)$ by sampling (in some RL models, $\mathbf{p}_j$ is not explicitly given so that the mean of $\mathbf{y}^k$ has to be estimated by sampling)?

Variance Reduction in Stochastic Value Iteration for MDP II

We carry out the VI iteration as:

$$y_i^{k+1} = \min_{j \in \mathcal{A}_i} \{c_j + \gamma \mathbf{p}_j^T \tilde{\mathbf{y}}^k + \gamma \mathbf{p}_j^T (\mathbf{y}^k - \tilde{\mathbf{y}}^k)\}, \ \forall i,$$

where $\tilde{\mathbf{y}}^k$ is updated at the geometric pace as before. Or compute once a while for a hash vector

$$\tilde{c}_j^k = c_j + \gamma \mathbf{p}_j^T \tilde{\mathbf{y}}^k, \ \forall j$$

and do

$$y_i^{k+1} = \min_{j \in \mathcal{A}_i} \{\tilde{c}_j^k + \gamma \mathbf{p}_j^T (\mathbf{y}^k - \tilde{\mathbf{y}}^k)\}, \ \forall i.$$

Then we only need to approximate

$$\mathbf{p}_j^T (\mathbf{y}^k - \tilde{\mathbf{y}}^k) = \mu_j (\mathbf{y}^k - \tilde{\mathbf{y}}^k).$$

Since $\mathbf{y}^* \geq \mathbf{y}^k \geq \tilde{\mathbf{y}}^k$ during the period of k to $2k$ and $(\mathbf{y}^k - \tilde{\mathbf{y}}^k)$ monotonically converges to zero, the norm of $(\mathbf{y}^k - \tilde{\mathbf{y}}^k)$ becomes smaller and smaller so that only a constant number of samples are needed to estimate the mean for desired accuracy, which leads to a geometrically convergent algorithm with high probability.