Properties of Multivariate $q$-Gaussian Distribution and its application to Smoothed Functional Algorithms for Stochastic Optimization

Debarghya Ghoshdastidar

M.E. (System Science and Automation)
S.R.No: 4910-412-101-07653

Supervised by:
Dr. Ambedkar Dukkipati
Prof. Shalabh Bhatnagar
Outline

1. Problem Framework
2. Smoothed Functional method
   - Optimization using SF
3. $q$-Gaussian distribution
   - Generalized Co-moments
4. Gradient-based Algorithms
   - Convergence of algorithms
5. Newton-based Algorithms
6. Simulation Results
   - Queuing network
   - Results
7. Conclusion
Outline

1. Problem Framework

2. Smoothed Functional method
   - Optimization using SF

3. \( q \)-Gaussian distribution
   - Generalized Co-moments

4. Gradient-based Algorithms
   - Convergence of algorithms

5. Newton-based Algorithms

6. Simulation Results
   - Queuing network
   - Results

7. Conclusion
System: Discrete event system, \( \{Y_m : m \geq 0\} \), controlled by parameter \( \theta \in C \), closed and convex subset of \( \mathbb{R}^N \).

Cost: Long run average cost, \( J(\theta) = E_{\nu_\theta}[h(Y)] \), where \( \nu_\theta \) is the stationary distribution of process, and \( h(Y) \) is the single-stage cost.

Objective: Minimize \( J(\theta) \) with respect to \( \theta \in C \).

Issue: No analytical relationship between \( J(\theta) \) and \( \theta \).

Solution: Perform optimization with derivatives of \( J \), estimated using Smoothed Functional approach.
Assumptions:

- The process is ergodic for a given $\theta$, i.e., for large $L$

$$J(\theta) = \mathbb{E}_{\nu_0}[h(Y)] \approx \frac{1}{L} \sum_{m=0}^{L-1} h(Y_m).$$

- $J(.)$ is twice continuously differentiable for all $\theta \in C$.
- The process remains stable under the sequence of parameter updates.
  (Technically, we assume existence of a stochastic Lyapunov function under any non-anticipative parameter sequence.)
Outline

1. Problem Framework
2. Smoothed Functional method
   - Optimization using SF
3. q-Gaussian distribution
   - Generalized Co-moments
4. Gradient-based Algorithms
   - Convergence of algorithms
5. Newton-based Algorithms
6. Simulation Results
   - Queuing network
   - Results
7. Conclusion
Let \(f : C \mapsto \mathbb{R}\) be any function, then given a function \(G_\beta : \mathbb{R}^N \mapsto \mathbb{R}\) satisfying the Rubinstein conditions\(^1\), we have

**Definition (Smoothed Functional)**

\[
S_\beta[f(\theta)] = \int_{\mathbb{R}^N} G_\beta(\eta) f(\theta - \eta) \, d\eta
\]

---

**Figure:** Unsmoothed function,

\[f(x) = x^2 - \frac{1}{4} e^{-x^2} \cos(8\pi x)\]

**Figure:** Smoothed function,

\[S_{0.1}[f(x)]\]

---

Rubinstein conditions:

- \( G_\beta(\eta) = \frac{1}{\beta N} G \left( \frac{\eta}{\beta} \right) \), where \( G \left( \frac{\eta}{\beta} \right) = G_1 \left( \frac{\eta^{(1)}}{\beta}, \frac{\eta^{(2)}}{\beta}, \ldots, \frac{\eta^{(N)}}{\beta} \right) \)
- \( G_\beta(\eta) \) is piecewise differentiable in \( \eta \)
- \( G_\beta(\eta) \) is a probability distribution function, i.e.,

\[
S_\beta[f(\theta)] = E_{G_\beta(\eta)}[f(\theta - \eta)]
\]

- \( \lim_{\beta \to 0} G_\beta(\eta) = \delta(\eta) \), where \( \delta(\eta) \) is the Dirac delta function
- \( \lim_{\beta \to 0} S_\beta[f(\theta)] = f(\theta) \)

Examples of smoothing kernels:

- Gaussian distribution with covariance matrix \( \beta^2 I_{N \times N} \)
- Cauchy distribution with scale parameter \( \beta \)
- Uniform distribution on interval \( \left[ -\frac{\beta}{2}, \frac{\beta}{2} \right]^N \)
Optimization methods:

- Gradient descent algorithm

\[ x_{n+1} = P_{[-1,1]} \left( x_n - \frac{1}{n} \nabla_x f(x_n) \right) \]

- Newton based search method

\[ x_{n+1} = P_{[-1,1]} \left( x_n - \frac{1}{n} \left( \nabla^2_x f(x_n) \right)^{-1} \nabla_x f(x_n) \right) \]

Figure: Optimum found using Gradient (red), Newton (blue), SF (yellow).
Smoothed Gradient

\[ \nabla_\theta S_\beta [f(\theta)] = E_{G(\eta)} [g_1(\eta)f(\theta + \beta \eta) | \theta] \]

Smoothed Hessian

\[ \nabla^2_\theta S_\beta [f(\theta)] = E_{G(\eta)} [g_2(\eta)f(\theta + \beta \eta) | \theta] \]

Stochastic framework:

- Random vector \( \eta \sim G = G_1 \).
- Process \( \{Y_m\} \) controlled by parameter \( (\theta + \beta \eta) \).
- Functions \( g_1, g_2 \) depend on nature of \( G \).
Two-sided Smoothed Functional

\[ S'_{\beta}[f(\theta)] = \frac{1}{2} \int_{-\infty}^{\infty} G_{\beta}(\eta) \left( f(\theta - \eta) + f(\theta + \eta) \right) d\eta \]

Two-sided derivatives

\[ \nabla_{\theta} S'_{\beta}[f(\theta)] = E_{G(\eta)} \left[ g'_1(\eta) \left( f(\theta + \beta \eta) - f(\theta - \beta \eta) \right) \right|_{\theta} \]
\[ \nabla^2_{\theta} S'_{\beta}[f(\theta)] = E_{G(\eta)} \left[ g'_2(\eta) \left( f(\theta + \beta \eta) + f(\theta - \beta \eta) \right) \right|_{\theta} \]

- Two simultaneous processes controlled by parameters \((\theta + \beta \eta)\) and \((\theta - \beta \eta)\).
Outline

1. Problem Framework
2. Smoothed Functional method
   - Optimization using SF
3. $q$-Gaussian distribution
   - Generalized Co-moments
4. Gradient-based Algorithms
   - Convergence of algorithms
5. Newton-based Algorithms
6. Simulation Results
   - Queuing network
   - Results
7. Conclusion
**q-Gaussian distribution**

\[
G_{q,\beta}(x) = \frac{1}{\beta_q K_q} \left( 1 - \frac{(1 - q)}{(3 - q)\beta_q^2} (x - \mu_q)^2 \right)^{\frac{1}{1-q}} + \quad \text{for all } x \in \mathbb{R},
\]

- **q-mean**, \( \mu_q = \frac{\int_{\mathbb{R}} x p(x)^q \, dx}{\int_{\mathbb{R}} p(x)^q \, dx} \)
- **q-variance**, \( \beta_q^2 = \frac{\int_{\mathbb{R}} (x - \mu_q)^2 p(x)^q \, dx}{\int_{\mathbb{R}} p(x)^q \, dx} \)
- \( y_+ = \max(y, 0) \) is Tsallis cut-off condition

\[
K_q = \begin{cases} 
\frac{\sqrt{\pi} \sqrt{3-q}}{\sqrt{1-q}} \frac{\Gamma\left(\frac{2-q}{1-q}\right)}{\Gamma\left(\frac{5-3q}{2(1-q)}\right)} & \text{for } -\infty < q < 1 \\
\frac{\sqrt{\pi} \sqrt{3-q}}{\sqrt{1-q}} \frac{\Gamma\left(\frac{3-q}{2(q-1)}\right)}{\Gamma\left(\frac{1}{q-1}\right)} & \text{for } 1 < q < 3
\end{cases}
\]
<table>
<thead>
<tr>
<th>Framework</th>
<th>SF Method</th>
<th>q-Gaussian</th>
<th>Gq-SF</th>
<th>Nq-SF</th>
<th>Simulation</th>
<th>Conclusion</th>
</tr>
</thead>
</table>

**Properties:**

- Obtained by maximizing Tsallis entropy,

\[
H_q(p) = \frac{1 - \int_X [p(x)]^q \, dx}{q - 1}, \quad q \in \mathbb{R}.
\]

- Probability distribution only for \( q < 3 \).
- It provides a family of distributions, with behavior controlled by parameter \( q \).
- Finite support for \( q < 1 \), and infinite support for \( q > 1 \).
- Variance is finite for \( q < \frac{5}{3} \), given by \( \beta^2 = \beta_q^2 \left( \frac{3-q}{5-3q} \right) \).

**Special cases:**

- Gaussian distribution as \( q \to 1 \)
- Cauchy distribution for \( q = 2 \)
- Uniform distribution as \( q \to -\infty \)
Multivariate $q$-Gaussian distribution$^2$:

$$G_{q,\beta}(X) = \frac{1}{\beta^N K_{q,N}} \left( 1 - \frac{(1-q) \|X\|^2}{\beta^2 \left( (N+4) - (N+2)q \right)} \right)^{\frac{1}{1-q}},$$

for all $X \in \mathbb{R}^N$, with the normalizing constant

$$K_{q,N} = \begin{cases} 
\left( \frac{(N+4)-(N+2)q}{1-q} \right)^{\frac{N}{2}} \pi^{N/2} \Gamma\left( \frac{2-q}{1-q} \right) \frac{\Gamma\left( \frac{2-q}{1-q} + \frac{N}{2} \right)}{\Gamma\left( \frac{2-q}{1-q} + N \right)} & \text{for } q < 1 \\
\left( \frac{(N+4)-(N+2)q}{q-1} \right)^{\frac{N}{2}} \pi^{N/2} \Gamma\left( \frac{1}{q-1} - \frac{N}{2} \right) \frac{\Gamma\left( \frac{1}{q-1} \right)}{\Gamma\left( \frac{1}{q-1} - \frac{N}{2} \right)} & \text{for } 1 < q < \frac{N+4}{N+2} 
\end{cases}$$

Support set:

\[ \Omega_q = \begin{cases} 
\left\{ x \in \mathbb{R}^N : \|x\|^2 \leq \frac{(N+4)-(N+2)q}{(1-q)} \beta^2 \right\} & \text{for } q < 1 \\
\mathbb{R}^N & \text{for } 1 < q < \frac{N+4}{N+2}
\end{cases} \]

Consider

- \( X^{(1)}, X^{(2)}, \ldots, X^{(N)} \) identical \( q \)-Gaussian distributed with 
  \( q \in \left( -\infty, \frac{N+4}{N+2} \right), q \neq 1 \).
- \( E \left[ X^{(i)} \right]^2 = 1 \) for all \( i = 1, \ldots, N \).
- \( E \left[ X^{(i)} X^{(j)} \right] = 0 \) for all \( i, j = 1, \ldots, N, i \neq j \).
- \( \rho(X) = \left( 1 - \frac{(1-q)}{(N+4)-(N+2)q} \|X\|^2 \right) \).
- \( b, b_1, b_2, \ldots, b_N \in \mathbb{N} \).
Generalized co-moments:

\[ E_{G_q} \left[ \frac{(X^{(1)})^{b_1} (X^{(2)})^{b_2} \ldots (X^{(N)})^{b_N}}{\rho(X)^b} \right] = \begin{cases} \bar{K} \left( \frac{(N + 4) - (N + 2)q}{1 - q} \right) \sum_{i=1}^{N} \frac{b_i}{2} \left( \prod_{i=1}^{N} \frac{b_i!}{2^{b_i} (\frac{b_i}{2})!} \right) & \text{if } b_i \text{ is even for all } i = 1, 2, \ldots, N \\ 0 & \text{otherwise} \end{cases} \]

with

\[ \bar{K} = \begin{cases} \frac{\Gamma\left(\frac{1}{1-q} - b + 1\right)\Gamma\left(\frac{1}{1-q} + 1 + \frac{N}{2}\right)}{\Gamma\left(\frac{1}{1-q} + 1\right)\Gamma\left(\frac{1}{1-q} - b + 1 + \frac{N}{2} + \sum_{i=1}^{N} \frac{b_i}{2}\right)} & \text{if } q \in (-\infty, 1) \\ \frac{\Gamma\left(\frac{1}{q-1}\right)\Gamma\left(\frac{1}{q-1} + b - \frac{N}{2} - \sum_{i=1}^{N} \frac{b_i}{2}\right)}{\Gamma\left(\frac{1}{q-1} + b\right)\Gamma\left(\frac{1}{q-1} - \frac{N}{2}\right)} & \text{if } q \in \left(1, 1 + \frac{2}{N+2}\right) \end{cases} \]
Remarks:

- Expression holds only if Gamma functions exist, i.e.,
  \[ b < (1 + \frac{1}{1-q}) \] for \( q < 1 \),
  \[ \left( \frac{1}{q-1} + b - \frac{N}{2} - \sum_{i=1}^{N} \frac{b_i}{2} \right) > 0 \] for \( 1 < q < \frac{N+4}{N+2} \).

- Putting \( b = 0 \), we get an expression for higher order moments and co-moments.

- All moments exists for \( q < 1 \).

- Generalized co-moments required for to prove convergence of algorithms.
Outline

1. Problem Framework

2. Smoothed Functional method
   - Optimization using SF

3. $q$-Gaussian distribution
   - Generalized Co-moments

4. Gradient-based Algorithms
   - Convergence of algorithms

5. Newton-based Algorithms

6. Simulation Results
   - Queuing network
   - Results

7. Conclusion
Gradient Estimation using $q$-Gaussian SF:

**One-simulation $q$-SF Gradient**

\[ \nabla_{\theta} S_{q, \beta}[J(\theta)] = \mathbb{E}_{G(\eta)} \left[ g_1(\eta) J(\theta + \beta \eta) \bigg\vert \theta \right] \]

where \[ g_1(\eta) = \frac{2\eta}{\beta \rho(\eta) ((N + 4) - (N + 2)q)} \]

**Two-simulation $q$-SF Gradient**

\[ \nabla_{\theta} S'_{\beta}[J(\theta)] = \mathbb{E}_{G(\eta)} \left[ \frac{1}{2} g_1(\eta) \left( J(\theta + \beta \eta) - J(\theta - \beta \eta) \right) \bigg\vert \theta \right] \]
Approximation:

$$\nabla_{\theta} S_{q, \beta}[J(\theta)] \approx \frac{1}{ML} \sum_{n=0}^{M-1} \left( g_1(\eta_n) \sum_{m=0}^{L-1} h(Y_{nL+m}) \right),$$

where \(\{Y_{nL+m}\}\) controlled by \((\theta + \beta \eta_n)\), for large \(M, L\).

Gradient descent method:

1. Fix \(M, L, q\) and \(\beta\).
2. Set parameter update \(\theta_0 = \theta_{\text{initial}}\).
3. For \(k = 0\) to a fixed number of steps
   1. Estimate \(\nabla_{\theta_k} S_{q, \beta}[J(\theta_k)]\) using above approximation.
   2. Update \(\theta_{k+1} = \mathcal{P}_C (\theta_k - a_k \nabla_{\theta_k} S_{q, \beta}[J(\theta_k)])\).
4. Output final parameter vector.
Multi-timescale approach:

- Consider a sequence initialized at $Z_0 = 0$, and

$$Z_{nL+m+1} = (1 - b_n)Z_{nL+m} + b_n g_1(\eta_n)h(Y_{nL+m})$$

for $m = 0$ to $L - 1$.

- Using results presented by Borkar$^3$, we show

$$\|Z_{nL} - g_1(\eta_n)J(\theta + \beta \eta_n)\| \to 0$$

almost surely as $n \to \infty$.

- Above claim holds even when $\theta$ varies very slowly with respect to $Z$ updates.

---

Step-sizes:

\[(a_n)_{n \geq 0}, (b_n)_{n \geq 0} \subset \mathbb{R}^+ \text{ satisfying } \sum_{n=0}^{\infty} a_n^2 < \infty, \sum_{n=0}^{\infty} b_n^2 < \infty, \]
\[
\sum_{n=0}^{\infty} a_n = \sum_{n=0}^{\infty} b_n = \infty \text{ and } a_n = o(b_n), \text{ i.e., } \lim_{n \to \infty} \frac{a_n}{b_n} = 0.
\]

The Gq-SF1 Algorithm:

1. Fix $M$, $L$, $q$ and $\beta$.
2. Set gradient update $Z_0 = 0$, parameter update $\theta_0 = \theta_{\text{initial}}$.
3. For $n = 0$ to $M - 1$
   1. Generate $\eta \in \mathbb{R}^N$, $\eta \sim G_{q,1}$.
   2. For $m = 0$ to $L - 1$
      1. Simulate $Y_{nL+m}$ with parameter $(\theta_n + \beta \eta_n)$.
      2. $Z_{nL+m+1} = (1 - b_n)Z_{nL+m} + b_ng_1(\eta_n)h(Y_{nL+m})$.
   3. Update $\theta_{n+1} = \mathcal{P}_C (\theta_n - a_n Z_{(n+1)L})$.
4. Output final parameter vector.
Convergence of algorithm:

- Updates

\[ \theta_{n+1} = \mathcal{P}_C(\theta_n - a_n g_1(\eta_n)J(\theta_n + \beta \eta_n)) \]
\[ = \mathcal{P}_C(\theta_n + a_n [-\nabla_{\theta_n} J(\theta_n) + \Delta(\theta_n) + \xi_n]) , \]

- Noise term

\[ \xi_n = \nabla_{\theta_n} S_{q,\beta}[J(\theta_n)] - g_1(\eta_n)J(\theta_n + \beta \eta_n), \]

- It is martingale difference noise, with bounded variance for \( q \in (0, \frac{N+4}{N+2}) \), \( q \neq 1 \). Hence, cancels out as \( n \to \infty \).

- Updates track the ODE

\[ \dot{\theta}(t) = \tilde{\mathcal{P}}_C(-\nabla_{\theta(t)} J(\theta(t)) + \Delta(\theta(t)) ) \]
Convergence of algorithm:

- Error term
  \[ \Delta(\theta) = \nabla_\theta J(\theta) - \nabla_\theta S_{q,\beta}[J(\theta)], \]

- Error satisfies \( \|\Delta(\theta)\| = o(\beta) \) for \( q < \frac{N+4}{N+2}, q \neq 1. \)

- So, for small value of \( \beta \), updates track ODE
  \[ \dot{\theta}(t) = \tilde{P}_C(\nabla_{\theta(t)} J(\theta(t))) \]

where \( \tilde{P}_C(f(x)) = \lim_{\epsilon \downarrow 0} \left( \frac{P_C(x+\epsilon f(x))-x}{\epsilon} \right) \).

- Hence, results by Kushner and Clark\(^4\) lead to the conclusion that algorithms converge to a local minimum of \( J(\theta) \).

Outline

1 Problem Framework

2 Smoothed Functional method
   - Optimization using SF

3 $q$-Gaussian distribution
   - Generalized Co-moments

4 Gradient-based Algorithms
   - Convergence of algorithms

5 Newton-based Algorithms

6 Simulation Results
   - Queuing network
   - Results

7 Conclusion
Hessian Estimation using $q$-Gaussian SF:

**One-simulation $q$-SF Hessian**

\[
\nabla^2_{\theta} S_{q,\beta}[J(\theta)] = \mathbb{E}_{G(\eta)} \left[ g_2(\eta) J(\theta + \beta \eta) \mid \theta \right],
\]

where $g_2(\eta) \in \mathbb{R}^{N \times N}$ such that

\[
[g_2(\eta)]_{i,j} = \begin{cases} 
\frac{4q}{\beta^2((N+4)-(N+2)q)^2} \frac{\eta^{(i)} \eta^{(j)}}{\rho(\eta)^2} & \text{for } i \neq j \\
\left( \frac{4q}{\beta^2((N+4)-(N+2)q)^2} \frac{(\eta^{(i)})^2}{\rho(\eta)^2} - \frac{2}{\beta^2((N+4)-(N+2)q)\rho(\eta)} \right) & \text{for } i = j
\end{cases}
\]
The $N_q$-SF1 Algorithm:

1. Fix $M$, $L$, $q$ and $\beta$.
2. Set gradient update $Z_0 = 0$, Hessian update $W_0 = 0$, and parameter update $\theta_0 = \theta_{\text{initial}}$.
3. For $n = 0$ to $M - 1$
   - Generate $\eta \in \mathbb{R}^N$, $\eta \sim G_{q,1}$.
   - For $m = 0$ to $L - 1$
     - Simulate $Y_{nL+m}$ with parameter $(\theta_n + \beta \eta_n)$.
     - $Z_{nL+m+1} = (1 - b_n) Z_{nL+m} + b_n g_1(\eta_n) h(Y_{nL+m})$.
     - $W_{nL+m+1} = (1 - b_n) W_{nL+m} + b_n g_2(\eta_n) h(Y_{nL+m})$.
   - Project $W_{(n+1)L}$ onto set of positive definite matrices as $\mathcal{P}_{pd}(W_{(n+1)L})$.
4. Update $\theta_{n+1} = \mathcal{P}_C (\theta_n - a_n \mathcal{P}_{pd}(W_{(n+1)L})^{-1} Z_{(n+1)L})$.
5. Output final parameter vector.
Variants :

- **Jacobi variant**
  The updates $W$ is diagonal in expected. So, off-diagonal terms set to zero and diagonal elements updated.

- **Different step-sizes**
  Updates of $Z$ and $W$ are independent over the $L$ steps. So, they are updated at different timescales. Bhatnagar\(^5\) showed that updating $W$ on a faster timescale improves performance.

- **Two-simulation $q$-SF Hessian**

\[
\nabla^2_{\theta} S^\prime_{\beta}[J(\theta)] = E_{G(\eta)} \left[ \frac{1}{2} g_2(\eta) \left( J(\theta + \beta \eta) + J(\theta - \beta \eta) \right) \right]_{\theta}
\]

Outline

1. Problem Framework
2. Smoothed Functional method
   - Optimization using SF
3. $q$-Gaussian distribution
   - Generalized Co-moments
4. Gradient-based Algorithms
   - Convergence of algorithms
5. Newton-based Algorithms
6. Simulation Results
   - Queuing network
   - Results
7. Conclusion
For node $i$:

- External arrival $\sim \text{Exp}(\lambda_i)$.
- Probability of leaving system after service $p_i$.
- Service time $S^n_i(\theta_i) = U_i(n) \left( \frac{1}{R_i} + \|\theta_i(n) - \bar{\theta}_i\|^2 \right)$, where
  - $U_i \sim \text{Uniform}(0,1)$
  - $R_i$ constant
  - $\theta_i$ lies in a $N_i$-dimensional hypercube
- Cost (at each event) - total length of queues, or total waiting time of customers.

Performance measure: $d(\theta) = \sum_{i=1}^{K} \|\theta_i - \bar{\theta}_i\|^2$
4-dimensional case:

- $K = 2, N_1 = N_2 = 2$, i.e., $N = 4$.
- $\lambda = (0.2, 0.1), p = (0, 0.4), R = (10, 20)$.
- $\theta_{min} = 0.1, \theta_{max} = 0.6$, i.e., $C = [0.1, 0.6]^4$.
- $\bar{\theta} = (0.3, 0.3, 0.3, 0.3), \theta_{initial} = (0.1, 0.1, 0.6, 0.6)$
- $M = 1000, L = 100$.

Figure: Convergence behavior of various algorithms for $\beta = 0.1$. 
Table: Performance of all algorithms for different $q$ with $\beta = 0.05$.

<table>
<thead>
<tr>
<th>$q$</th>
<th>$G_q$-SF1</th>
<th>$G_q$-SF2</th>
<th>$N_q$-SF1</th>
<th>$N_q$-SF2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>0.1024±0.0620</td>
<td>0.0734±0.1287</td>
<td>0.0855±0.0229</td>
<td>0.0165±0.0061</td>
</tr>
<tr>
<td>0.4</td>
<td>0.1074±0.0733</td>
<td>0.0274±0.0170</td>
<td>0.0940±0.0375</td>
<td>0.0162±0.0055</td>
</tr>
<tr>
<td>0.7</td>
<td>0.0616±0.0244</td>
<td>0.0262±0.0203</td>
<td>0.0826±0.0305</td>
<td>0.0130±0.0052</td>
</tr>
<tr>
<td>Gaussian</td>
<td>0.0387±0.0152</td>
<td>0.0112±0.0037</td>
<td>0.0761±0.0321</td>
<td>0.0117±0.0046</td>
</tr>
<tr>
<td>1.3</td>
<td>0.0413±0.0205</td>
<td>0.0101±0.0029</td>
<td>0.0970±0.0366</td>
<td>0.0209±0.0097</td>
</tr>
</tbody>
</table>

Table: Performance of $G_q$-SF2 and $N_q$-SF2 algorithms for different $q$ and $\delta$, where $\beta = 0.1$ and the step-sizes are $a(n) = \frac{1}{n}$, $b(n) = \frac{1}{n^{0.75}}$, $c(n) = \frac{1}{n^\delta}$.

<table>
<thead>
<tr>
<th>$q$</th>
<th>$\delta$</th>
<th>0.55</th>
<th>0.65</th>
<th>0.75</th>
<th>$G_q$-SF2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>0.55</td>
<td>0.0133±0.0059</td>
<td>0.0095±0.0031</td>
<td>0.0087±0.0030</td>
<td>0.0229±0.0064</td>
</tr>
<tr>
<td></td>
<td>0.65</td>
<td>0.0112±0.0039</td>
<td>0.0104±0.0040</td>
<td>0.0097±0.0031</td>
<td>0.0250±0.0216</td>
</tr>
<tr>
<td></td>
<td>0.75</td>
<td>0.0121±0.0033</td>
<td>0.0101±0.0040</td>
<td>0.0087±0.0031</td>
<td>0.0098±0.0040</td>
</tr>
<tr>
<td>Gaussian</td>
<td>0.0093±0.0028</td>
<td>0.0076±0.0033</td>
<td>0.0088±0.0032</td>
<td>0.0067±0.0021</td>
<td></td>
</tr>
<tr>
<td>1.3</td>
<td>0.55</td>
<td>0.0072±0.0031</td>
<td>0.0091±0.0024</td>
<td>0.0102±0.0048</td>
<td>0.0086±0.0028</td>
</tr>
</tbody>
</table>
### 50-dimensional case:

- **$K = 5$, and for all $i$,**
  - $N_i = 10, \lambda_i = 0.2, p_i = 0.2, R_i = 10.$
- **$C = [0.1, 0.6]^{50}$.**
- **$\bar{\theta} = (0.3, 0.3, \ldots, 0.3), \theta_{initial} = (0.6, 0.6, \ldots, 0.6)$.**
- **Step-sizes are $a(n) = \frac{1}{n}, b(n) = \frac{1}{n^{0.75}}, c(n) = \frac{1}{n^{0.65}}$.**
Table: Performance $G_q$-SF2 algorithm for different values of $q$ and $\beta$.

<table>
<thead>
<tr>
<th>$q$</th>
<th>$\beta$</th>
<th>0.05</th>
<th>0.075</th>
<th>0.1</th>
<th>0.25</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>0.05</td>
<td>0.6323±0.0477</td>
<td>0.5721±0.0481</td>
<td>0.5325±0.0329</td>
<td>0.3128±0.0305</td>
</tr>
<tr>
<td>0.3</td>
<td>0.075</td>
<td>0.5385±0.0386</td>
<td>0.4913±0.0347</td>
<td>0.4422±0.0328</td>
<td>0.3113±0.1233</td>
</tr>
<tr>
<td>0.5</td>
<td>0.1</td>
<td>0.5039±0.0935</td>
<td>0.4798±0.1551</td>
<td>0.4118±0.1226</td>
<td>0.5363±0.3535</td>
</tr>
<tr>
<td>0.7</td>
<td>0.25</td>
<td>0.9438±0.2537</td>
<td>0.7993±0.2745</td>
<td>0.7315±0.2545</td>
<td>0.9571±0.3265</td>
</tr>
<tr>
<td>0.9</td>
<td>0.05</td>
<td>0.6436±0.3200</td>
<td>0.5634±0.2493</td>
<td>0.5634±0.2893</td>
<td>0.8520±0.2093</td>
</tr>
<tr>
<td>Gaussian</td>
<td>0.05</td>
<td>0.8618±0.0208</td>
<td>0.6182±0.0089</td>
<td>0.7225±0.0115</td>
<td>0.9768±0.0720</td>
</tr>
</tbody>
</table>

Table: Performance $N_q$-SF2 algorithm for different values of $q$ and $\beta$.

<table>
<thead>
<tr>
<th>$q$</th>
<th>$\beta$</th>
<th>0.05</th>
<th>0.075</th>
<th>0.1</th>
<th>0.25</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>0.05</td>
<td>0.3125±0.0316</td>
<td>0.2306±0.0279</td>
<td>0.2013±0.0233</td>
<td>0.4786±0.0745</td>
</tr>
<tr>
<td>0.3</td>
<td>0.075</td>
<td>0.5092±0.1137</td>
<td>0.2917±0.0242</td>
<td>0.2664±0.0307</td>
<td>0.8322±0.1238</td>
</tr>
<tr>
<td>0.5</td>
<td>0.1</td>
<td>0.9714±0.2154</td>
<td>0.5552±0.1214</td>
<td>0.4624±0.1020</td>
<td>1.2536±0.1482</td>
</tr>
<tr>
<td>0.7</td>
<td>0.25</td>
<td>1.2622±0.1383</td>
<td>1.0808±0.1629</td>
<td>0.9673±0.1896</td>
<td>1.4844±0.1196</td>
</tr>
<tr>
<td>0.9</td>
<td>0.05</td>
<td>1.3295±0.1019</td>
<td>1.1470±0.1248</td>
<td>1.1067±0.1631</td>
<td>1.5617±0.1039</td>
</tr>
<tr>
<td>Gaussian</td>
<td>0.05</td>
<td>1.2346±0.0934</td>
<td>1.0459±0.1641</td>
<td>1.0301±0.1617</td>
<td>1.6134±0.0823</td>
</tr>
</tbody>
</table>
Outline

1 Problem Framework

2 Smoothed Functional method
   - Optimization using SF

3 $q$-Gaussian distribution
   - Generalized Co-moments

4 Gradient-based Algorithms
   - Convergence of algorithms

5 Newton-based Algorithms

6 Simulation Results
   - Queuing network
   - Results

7 Conclusion
Contributions:

- Characterized the family of functions, suitable as smoothing kernels.

- Extended Gaussian Smoothed Functional approach to $q$-Gaussian distributions.

- Proposed two-timescale algorithms for stochastic optimization.

- Proved that proposed algorithms converge to a local optimum of objective function.

- Provided conditions for applicability of $q$-Gaussians for stochastic optimization methods.

- Presented simulation results showing that proposed algorithms perform better than their Gaussian counterparts in a considerable number of cases.