Piecewise linear approximation of empirical distributions under a Wasserstein distance constraint*

William Guevara-Alarcón[1] joint work with Philipp Arbenz[2]

[1]Université de Lausanne
[2]SCOR

Swiss Statistics Meeting 2018

*Publication in press in *Journal of Statistical Computation and Simulation*

SCOR The Art & Science of Risk

UNIL | Université de Lausanne

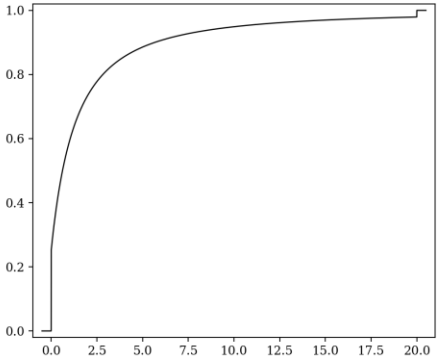
Outline

- Motivation
- PWL distributions and Wasserstein distance
- Admissible approximation and algorithm
- Conclusions

Main idea

Problem Empirical distributions of samples of large size imply a substantial amount of information to be stored.

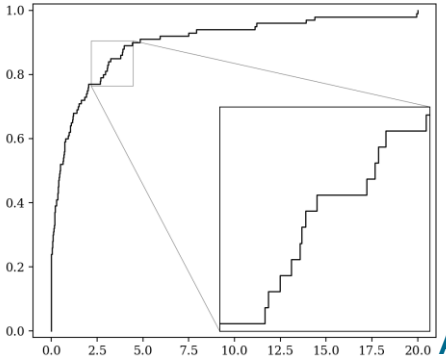
Model distribution F



Distribution not necessarily analytic.

Simulation

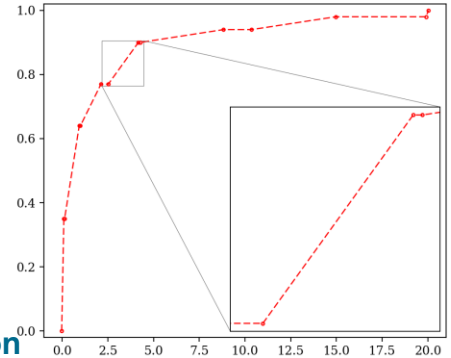
Empirical distribution $\mathbf{F_n}$



Sample of large size n.

Approximation

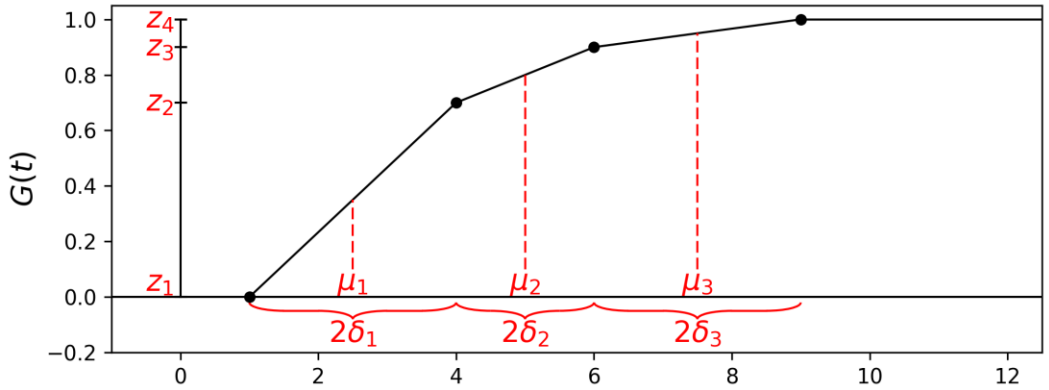
Piecewise linear approximation G



Preserves distribution shape with a smaller number of segments $S \ll n$

Solution Approximate empirical distribution with a PWL distribution with an **approximation error much smaller than the sampling error.**

Piecewise linear distributions



$S = 4$

- $\boldsymbol{z} = (0, 0.7, 0.9, 1)$
- $\boldsymbol{\mu} = (2.5, 5, 7.5)$
- $\boldsymbol{\delta} = (1.5, 1, 1.5)$

$X \sim G = PWL(\boldsymbol{z}, \boldsymbol{\mu}, \boldsymbol{\delta})$ with ***S-1* segments:**

- $(z_s, z_{s+1}]$: initial and end points
- $\mu_s = G^{-1}\left(\frac{z_s + z_{s+1}}{2}\right)$: local segment mean
- δ_s: slope parameter
- $z_s \in [0,1], \mu_s \in \mathbb{R}, \delta_s \geq 0$

Equivalent to a **discrete mixture of S-1 uniform:**

$$X = \sum_{s=1}^{S-1} \mathrm{I}\{J = s\} \left[2\delta_s U_s + \mu_s - \delta_s\right]$$

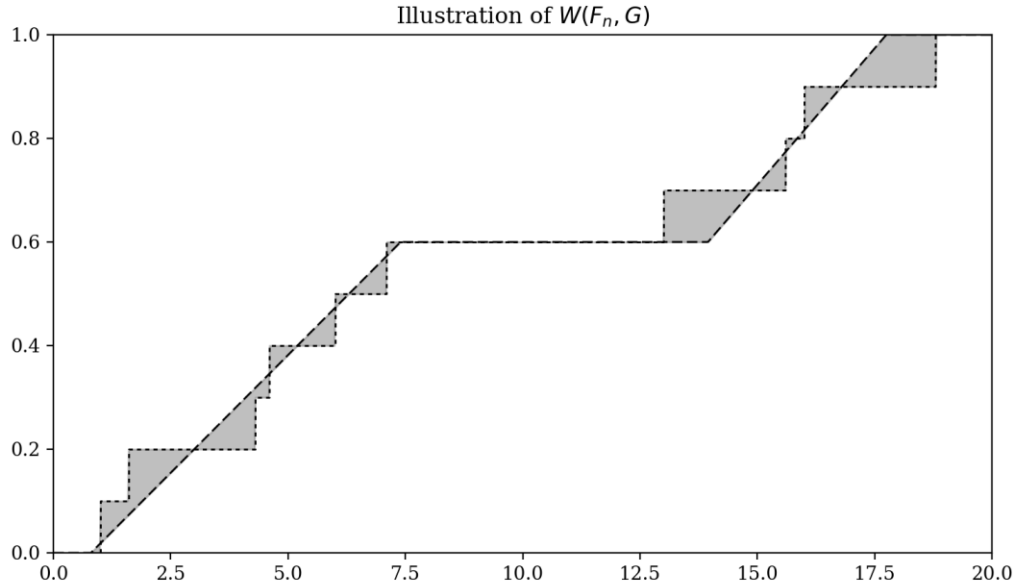
$U_s \sim U(0,1), \quad P[J = s] = z_{s+1} - z_s$

$s = 1, \ldots, S-1 \quad U_s, J$ indep.

Wasserstein distance

The Wasserstein distance between two distributions F_n and G is:

$$W(F_n, G) = \int_0^1 |F_n^{-1}(t) - G^{-1}(t)|\, dt = \int_{-\infty}^{\infty} |F_n(x) - G(x)|\, dx$$



Admissible PWL approximation

Idea:

Select a PWL distribution $G \sim PWL(\boldsymbol{z}, \boldsymbol{\mu}, \boldsymbol{\delta})$ that has an **approximation error at least one order of magnitude smaller than the expected sampling error:**

$$E[F_n] = E[G]$$

$$W(F_n, G) \leq \epsilon = 0.1 \cdot \widehat{W}(F, F_n)$$

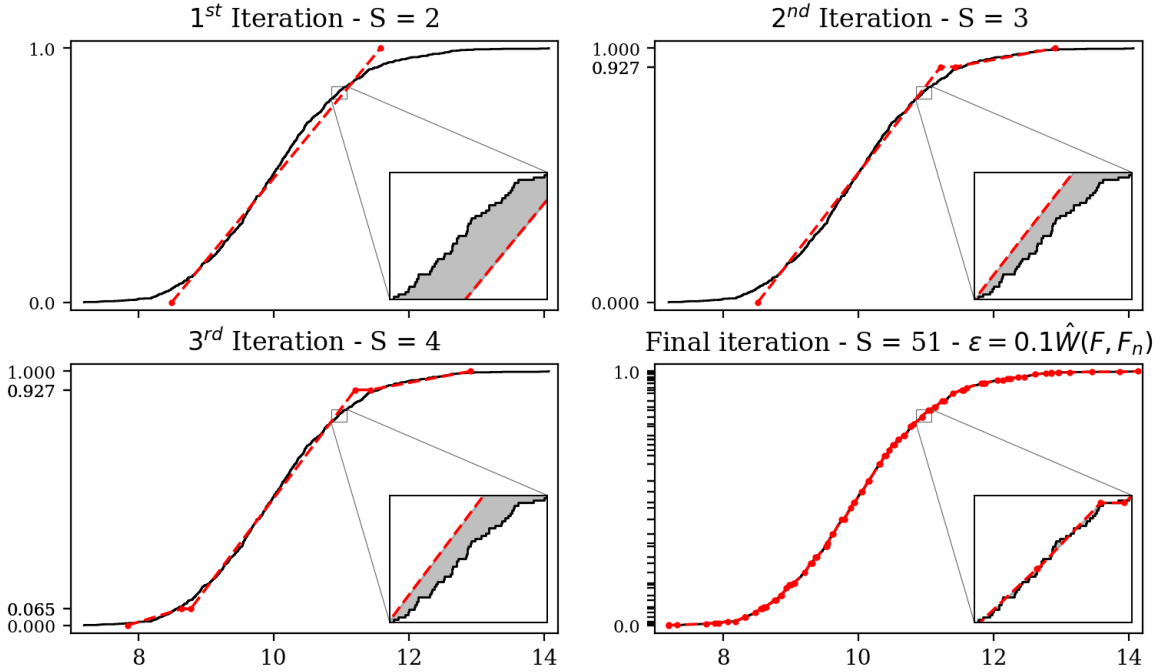
Theorem:

If $X \sim F$ and $\mathrm{E}[X^\tau] < \infty$ for some $\tau > 2$, then $\widehat{W}(F, F_n)$ is a consistent estimator of $\mathrm{E}[W(F, F_n)]$, with:

$$\widehat{W}(F, F_n) = \sqrt{\frac{2}{n\pi}} \sum_{i=1}^{n-1} \sqrt{\frac{i}{n}\frac{(n-i)}{n}} \left(X_{(i+1)} - X_{(i)}\right)$$

Approximation algorithm

Divide-and-conquer algorithm. Set $\hat{\epsilon} = 0.1 \cdot \widehat{W}(F, F_n) > 0$ and initialize with $S = 2$ and $\mathbf{z} = (0,1)$.



Performance of the algorithm

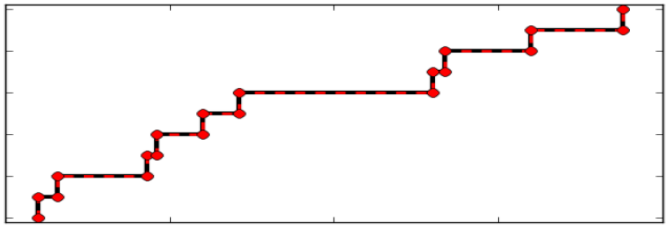
Average results over 100 repetitions for different distributions with mean 10 and standard deviation 12:

Distribution	n	Run time (in milliseconds)	Number of segments S-1	$\hat{W}(F,F_n)$	$W(F_n,G)$
Negative Binomial	10,000	89	51.2	0.1484	0.0142
(r = 0.75, p = 0.069)	100,000	359	66.3	0.0470	0.0045
Normal (μ = 10,	10,000	179	70.9	0.1542	0.0153
σ = 12)	100,000	645	93.0	0.0488	0.0049
Pareto (α = 6.55,	10,000	128	52.9	0.1569	0.0155
θ = 55.5)	100,000	505	78.0	0.0507	0.0050

- If the **sample size *n* increases**, ***ε* decreases**, while ***S* increases** in a **smaller proportion**.
- Run time is less than a second.

Comparison with other strategies

- **Store key statistics**
  - $E[X]$
  - $\mathrm{Var}[X]$
  - $\mathrm{Va}R_{99\%}[X]$
- **Store full sample**



Strategy	Distribution shape	Storage memory efficient
Store key statistics	✗	✓
Store full sample	✓	✗
Store PWL approximation	✓	✓

Conclusions

Piecewise linear approximation algorithm:

- Approximation has the **same mean** and **preserves the shape** of the empirical distribution.
- **The approximation error** is chosen to be **one order of magnitude smaller than the expected sampling error.**
- Algorithm is efficient in terms of **run time and storage memory.**
- The **algorithm is shift and scale invariant.**
- **Open source implementation** in Python is available.

email: William.GuevaraAlarcon@unil.ch
WGuevara-External@scor.com