# GPU Applications for Modern Large Scale Asset Management

## GTC 2014 – San José, California

Dr. Daniel Egloff

QuantAlea & IncubeAdvisory

March 27, 2014

# Outline

QuantAlea

## Portfolio Construction

# Outline

QuantAlea

# Outline

QuantAlea

Portfolio Construction

Constructing Asset Return Distributions

I) Entropy Approach

Large Scale Convex Optimization with GPUs

# Outline

QuantAlea

Portfolio Construction

Constructing Asset Return Distributions

I) Entropy Approach

Large Scale Convex Optimization with GPUs

II) Bayesian Approach

Parallel Metropolis Hastings on GPUs

# Outline

QuantAlea

# Portfolio Construction

QuantAlea

- Markowitz mean variance portfolio optimization
  - Optimally balance risk and performance

$$w_\lambda^* = \underset{w,\ \mathbf{1}^T w = b}{\arg\max}\ U_\lambda(w), \quad U_\lambda(w) = \mu^\top w - \lambda w^\top \Sigma w \tag{1}$$

- Risk budgeting
  - Construct portfolio so that risk contributions satisfy $RC_i(w) = b_i$ for given risk budgets $b_i$
  - For volatility risk measure we obtain

$$RC_i(w) = w_i \frac{(\Sigma w)_i}{\sqrt{w^\top \Sigma w}} \tag{2}$$

- For both approaches we need means $\mu$ and covariances $\Sigma$ of asset returns

# Asset Return Distributions

QuantAlea

Two steps

1. Prior asset return distribution from historical data or inverse optimization principle
2. Posterior asset return distribution should reflect expert views and beliefs
   - Views are distortions of a prior distribution
   - Views translate into constraints
   - Linear constraints on mean return pioneered by Black and Litterman

We discuss two approaches

- Entropy based approach
- Bayesian methods

# Entropy Approach

QuantAlea

- Minimum discrimination principle (Kullback and Leibler 1951) infers a posterior distribution such that it
  - Satisfies a set of moments constraints
  - Is as hard as possible to differentiate from given prior distribution
- Minimum discrimination principle suitable methodology to incorporate constraints on a prior distribution which are
  - Linear
  - Quadratic
  - Convex

# Nonparametric Setup

QuantAlea

- Sample the probability distributions
  - Discrete sample space $\mathcal{X} = \{x_1, \ldots, x_m\}$
  - Probabilities elements of the standard simplex

$$p = (p_1, \ldots, p_m)^\top \in \Delta^m \subset \mathbb{R}^m$$

- Implement views as distortions of prior probabilities $p_i$
- Express views as feature functions $f_j : \mathcal{X} \to \mathbb{R}$, $j = 1, \ldots, n$
- Identify feature vector $\mathbf{f} = (f_1, \ldots, f_n)^\top : \mathcal{X} \to \mathbb{R}^n$ with matrix $F = (f_{ji}) \in \mathbb{R}^{n \times m}$, with $f_{ji} = f_j(x_i)$

# Linear Constraints on Features

QuantAlea

- Expectation of feature $f : \mathcal{X} \to \mathbb{R}$ w.r.t measure $q \in \Delta^m$ is

$$E_q[f] = \sum q_i f(x_i) = q^\top f$$

- Minimum discrimination principle defines probability $p^*$ as

$$p^* = \underset{q \in \Delta^m}{\arg\min}\, q^\top (\log(q) - \log(p)) \tag{9}$$

subject to constraints

$$\mathbb{E}_q[f_j] \leq c_j, j = 1, \ldots, n, \quad \mathbb{E}_q[\tilde{f}_j] = \tilde{c}_j, j = 1, \ldots, \tilde{n}. \tag{10}$$

# Optimization Problem

QuantAlea

- Expectation views (10) lead to linear constraints

$$p^* = \underset{\substack{q \in \mathbb{R}_+^m, \mathbf{1}^\top q = 1 \\ Fq \leq c, \tilde{F} q = \tilde{c}}}{\arg\min} \; q^\top (\log(q) - \log(p)) \qquad (11)$$

- Need large number of samples for accurate representation of distributions with $m \simeq 10^6$
- With linear constraints: Lagrange duality to solve the optimal solution
  - Strong duality, i.e. no duality gap
  - Converting problem of dimension $m$ to much smaller problems of dimension $n + \tilde{n}$ to find optimal Lagrange multipliers

# Convex Constraints

QuantAlea

- How to extend to variance and covariance constraints?
- Lower variance constraint $q^\top f^2 - (q^\top f)^2 \geq c$ convex but dual problem is ill-conditioned
- Upper variance constraint $q^\top f^2 - (q^\top f)^2 \leq c$ not even convex
- Approximation by linearization $q^\top f^2 \leq c + (p^\top f)^2$ to get a second moment constraint
- But .... linearized version often not accurate enough

# Large Scale Convex Optimization

QuantAlea

- For convex constraints solve directly the primal problem
- Several accelerated first order methods (FOM) for convex optimization algorithms have been developed
- Modern algorithms adjust to the problem's geometry
- Examples
  - Mirror descent (MD) algorithm
  - Level bundle methods
  - Bundle mirror descent
  - Many different variations

# Exploiting Geometry

QuantAlea

- Gradient methods regularized with quadratic proximity term

$$y = \arg\min_{x \in \mathcal{X}} \{\gamma \langle \xi, \nabla f(x) \rangle + c\|\xi - x\|^2\}$$

- Replace $\|\cdot\|^2$ with some distance like function $D(\cdot, \cdot)$ that better exploits the geometry of $\mathcal{X}$ and is simple to calculate
- Candidates: Bregman type distance based on kernel $\omega : \mathcal{X} \to \mathbb{R}$

$$D(x, y) = D_\omega(x, y) \equiv \omega(x) - \omega(y) - \langle x - y, \omega'(y) \rangle$$

# Bregman Distances

QuantAlea

- Using $\omega(u) = \frac{1}{2}\|u\|^2$ gives Euclidian norm distance

$$D_\omega(x, y) = \frac{1}{2}\|x - y\|^2$$

- The entropy $\omega(x) = x^\top \log(x)$ on $\mathcal{X} = \Delta^m$ leads to

$$D_\omega(x, y) = x^\top \log\left(\frac{x}{y}\right)$$

which is 1-strongly convex with respect to the $L_1$ norm

$$D_\omega(x, y) \geq \frac{1}{2}\|x - y\|_1^2 \quad \forall x, y \in \Delta^m$$

# Constrained Mirror Descent Setup

QuantAlea

- Convex problem $\min_{x \in \mathcal{X}} f_0(x)$ subject to $f_1(x) \leq 0$
- $\mathcal{X} \subset E$, closed convex subset of f.d. vector space $E$
- $\|\cdot\|$ norm on $E$, $\|\cdot\|_*$ dual norm on $E^*$
- Duality pairing $\langle \cdot, \cdot \rangle : E^* \times E \to \mathbb{R}$
- Bregman distance $D_\omega(x, y)$ from kernel $\omega : \mathcal{X} \to \mathbb{R}$
- Prox mapping $\text{Prox}_x : E^* \to \mathcal{X}^\circ$

$$\text{Prox}_x(\xi) = \arg\min_{u \in \mathcal{X}} \{\langle \xi, u \rangle + D_\omega(u, x)\}$$

# Constrained Mirror Descent

QuantAlea

- Mirror descent with constraints is a simple FOM
- Initial search point $x_1 = x_\omega = \arg\min_{x \in \mathcal{X}} \omega(x)$
- For $t = 1, \ldots, N$ do
  - If $f_1(x_t) \leq tol$ then $i(t) = 0$ else $i(t) = 1$
  - Select step size $\gamma_t$
  - Update search point $x_{t+1} = \mathrm{Prox}_{x_t}(\gamma_t f'_{i(t)}(x_t))$
- Approximate solution after $N$ steps

$$\hat{x}_N = \underset{x \in \{x_t | i(t)=0\}}{\arg\min} f_0(x)$$

- Possible choice $tol = \gamma \|f'_1(x_t)\|_*$ and $\gamma_t = \gamma \|f'_{i(t)}(x_t)\|_*^{-1}$

# Geometry on Simplex

QuantAlea

- Optimization problem (11) formulated on simplex $\mathcal{X} = \Delta^m$
- Norm $\|\cdot\|_1$ with dual norm $\|\cdot\|_\infty$
- Prox mapping allows analytical expression

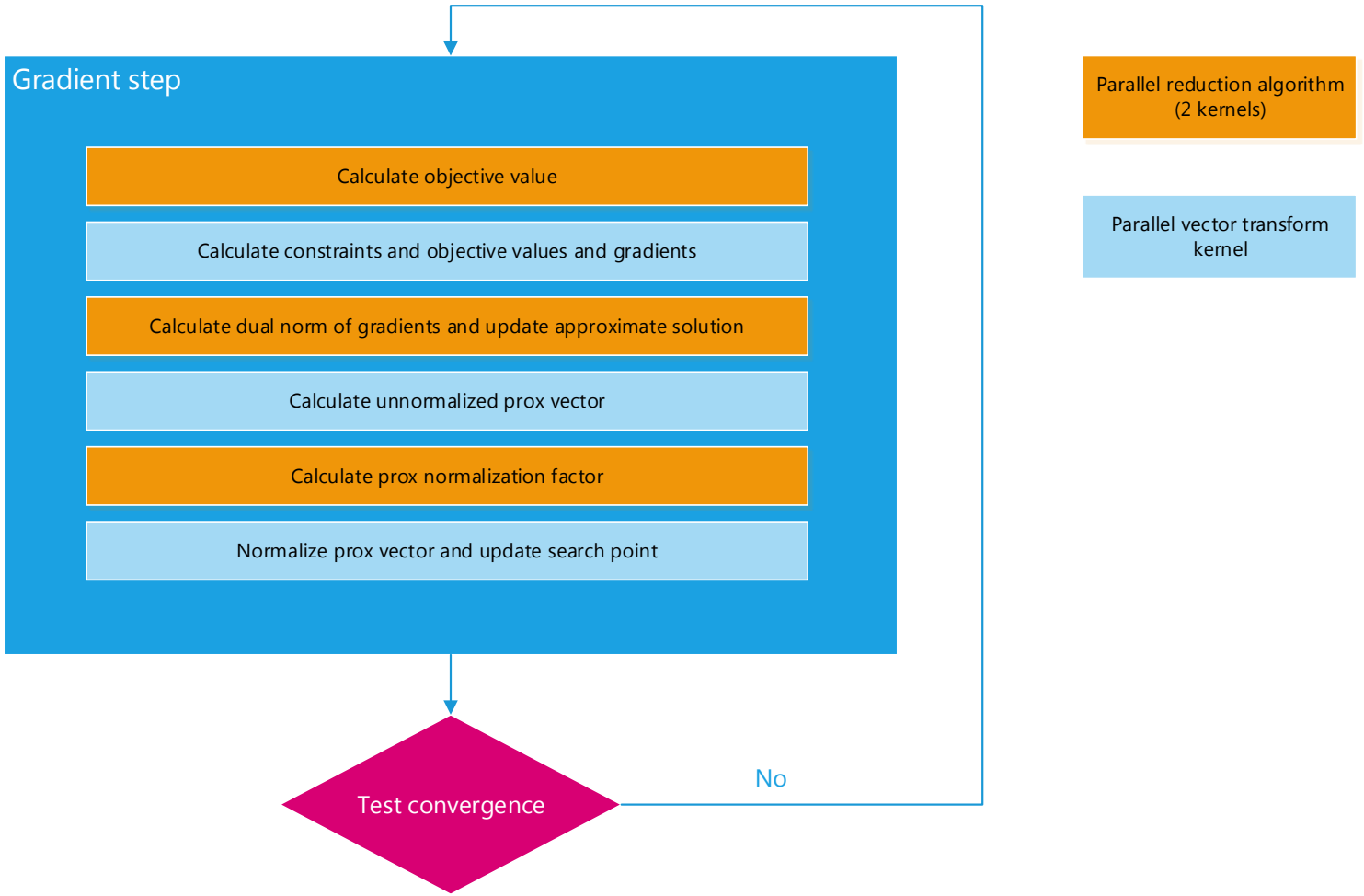
$$\text{Prox}_x(\xi) = \left(x^\top e^{-\xi}\right)^{-1} x \bullet e^{-\xi}$$

- Initial point $x_\omega = n^{-1}\mathbf{1}_n$ the uniform distribution

# Constrained Mirror Descent on GPU QuantAlea

- GPU implementation of constrained mirror descent algorithms for very large $n$
  - Parallel coordinate-wise calculation of gradients for objective function and constraints
  - Vector reductions to calculate objective value, norms and scalar products
  - Parallel coordinate-wise calculation of prox vector, search points and execution of gradient step
- Gradient steps can be further parallelized by coordinate aggregation or randomization methods

# Schematic Gradient Step



# Tuning with Kernel Fusion

QuantAlea

- Usually kernel launch time ($\approx 50\mu s$) not an issue
- For algorithms with many iterations, this becomes crucial
- Use kernel fusion technique to reduce number of kernel calls per gradient step
- Minimize number of arguments to pass to each kernel

# Bayesian Approach

QuantAlea

- Bayesian methods treat the model parameters as random variables
- Views go into the prior distributions of the model parameters
- Historical data is used to update the prior distribution
- Not as flexible as entropy approach
- Does not rely on optimization hence less fragile
- Work horse is a parallel Metropolis Hastings Markov chain Monte Carlo sampler

# Parallel Metropolis Hastings

QuantAlea

- Parallelization of Metropolis Hastings through multiple parallel chains
- Further parallelization through pre-fetching
- Require efficient parallel random number generation and branch free methods to sample from proposal distribution

# Parallel Metropolis Hastings

QuantAlea

GPU Metropolis Hastings sampler generation from generic target and proposal distribution expressions with F# and Alea.cuBase

```
let inline mhsamples (targetpdf : Expr<'T -> 'T>)
                     (proppdf : Expr<'T -> 'T -> 'T -> 'T>)
                     (proprnd : Expr<'T -> 'T -> 'T -> 'T>) = cuda {

    let! proppdf = proppdf |> Compiler.DefineInlineFunction
    let! proprnd = proprnd |> Compiler.DefineInlineFunction
    let! targetpdf = targetpdf |> Compiler.DefineInlineFunction

    let! kernel =
        <@ fun (steps:int) (scale:'T) (initial:deviceptr<'T>)
               (numbers:deviceptr<'T>) (result:deviceptr<'T>) ->
            let idx = blockIdx.x * blockDim.x + threadIdx.x
            let nchains = gridDim.x * blockDim.x

    . . .
```

# Parallel Metropolis Hastings

QuantAlea

Expressions for target distribution and proposal distribution are compiled directly into kernel

```
/// Dynamically generate expressions for MH at runtime
let targetpdf =
    <@ fun x ->
        exp(-x*x)*(2G + sin(5G*x) + sin(2G*x)) @>
let proppdf =
    <@ fun sigma x y ->
        exp(-(y - x)*(y - x)/(2G*sigma*sigma)) / (sigma*__sqrt2pi()) @>
let proprng =
    <@ fun sigma x u -> x + sigma*u @>

/// Compiler integrated into F#, callable through API
let mh = mhsamples targetpdf proppdf proprng |> Util.load Worker.Default
```

# Example

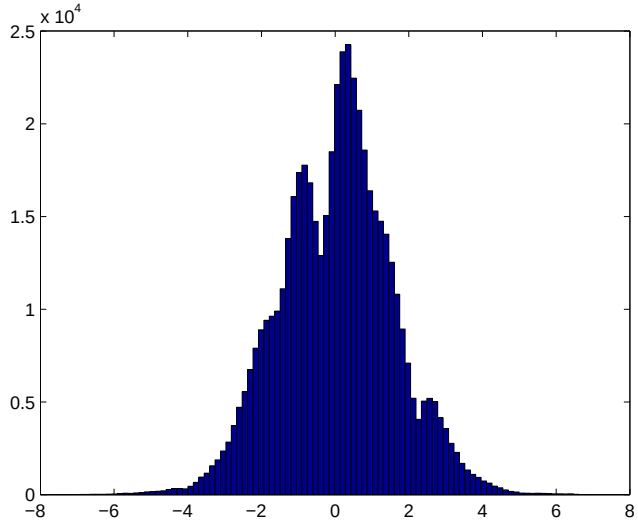



Figure : Sample histogram for target density $e^{-x^2}(2 + \sin(5x) + \sin(2x))$

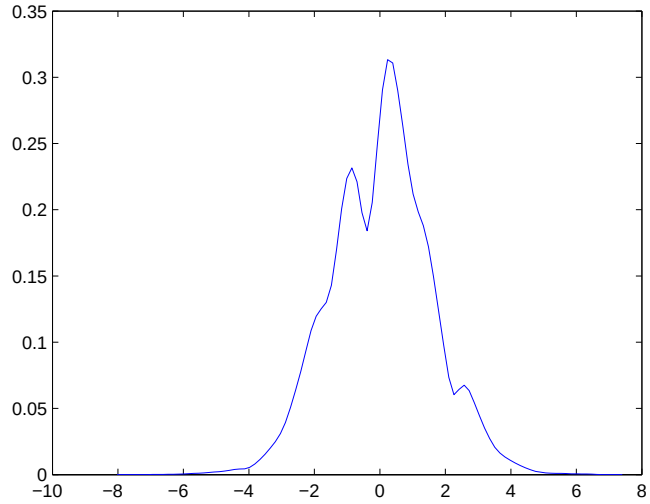



Figure : Kernel density smoothed sample pdf for $e^{-x^2}(2 + \sin(5x) + \sin(2x))$

# Conclusion

QuantAlea

- We have shown two applications of GPUs for quantitative asset management
- More applications in strategy calibration, back-testing and MIP
- Large scale convex optimization occurs in other fields such as
  - Machine learning such as e.g. deep believe network training in a big data context
  - Network analysis
  - Truss topology design
  - Compressed sensing
- Markov chain Monte Carlo methods require flexible ways to specify various parts such as target pdf, the proposal distribution and a dynamic way to derive tuning parameters, for which our Alea.cuBase F# GPU compiler is ideal