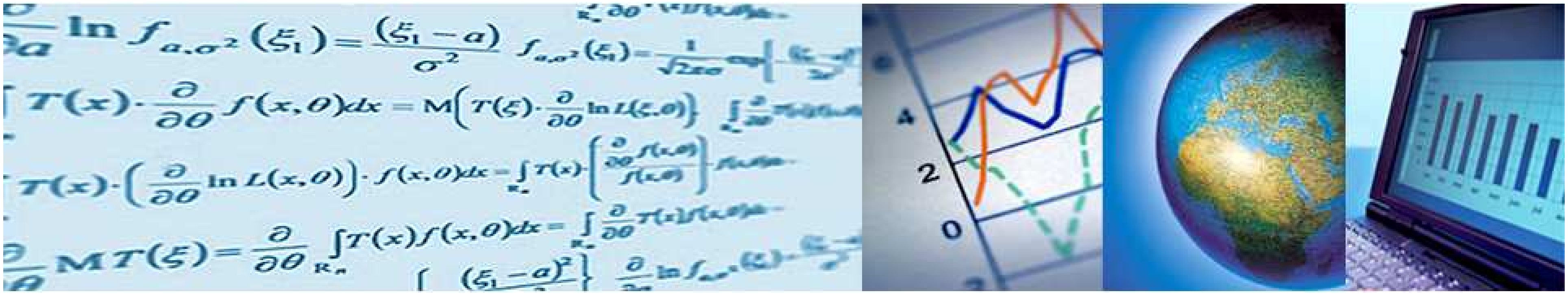
Credit Risk Optimization



Helmut Mausser

Fields Industrial Optimization Seminar

March 3, 2009

With: Ian Iscoe, Alex Kreinin, Oleksandr Romanko

Algorithmics

Overview

Objective: Re-balance a portfolio of financial instruments to minimize the risk of losses due to credit events

- Background
- Portfolio credit risk model
- Optimization models
- Computational results

Background

Corporate Bond Prices

	27-Feb-08		27-Feb-09		
Issue	Price ($)	Yield (%)	Price ($)	Yield (%)	Δ Price ($)
Ford 6.5% 8/1/18	70	11.5	16	45.6	-54
GM 7.7% 4/15/16	82	11.0	13	66.4	-69
Target 6.0% 1/15/18	103	5.6	100	6.0	-3
Walmart 5.375% 4/5/17	103	4.9	105	4.5	2

Automotive bonds lost about 80% of their value in one year

Bonds of discount retailers retained their value

Market is less confident that automative companies will be able to make the required interest and principal payments

Algorithmics

Credit Risk

The risk of monetary loss due to the default, or a change in the perceived likelihood of default, of a counterparty to a contract.

Counterparties (governments, companies) are assigned a credit rating reflecting the likelihood that they will honour their contracts

- Various rating scales (S&P, Moody's, Fitch, DBRS)
  - Range from AAA (best) to Default (worst)
- The lower the rating, the more compensation is required
  - Pay more interest
  - Provide more collateral

Credit Transition Matrix

Specifies the likelihood of migrating from one credit rating (state) to another over a fixed time horizon (usually one year)

e.g., annual transition matrix (% probability)

From \ To	AAA	AA	A	BBB	BB	B	CCC	Default
AAA	92.18	7.06	0.73	0.00	0.02	0.00	0.00	0.01
AA	1.17	90.84	7.63	0.26	0.07	0.01	0.00	0.02
A	0.05	2.39	91.83	5.07	0.50	0.13	0.01	0.02
BBB	0.05	0.24	5.20	88.49	4.88	0.80	0.16	0.18
BB	0.01	0.05	0.50	5.45	85.12	7.05	0.55	1.27
B	0.01	0.03	0.13	0.43	6.52	83.20	3.04	6.64
CCC	0.00	0.00	0.00	0.58	1.74	4.18	68.00	25.50
Default	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00

Credit Losses

Associated with each future credit state is a change in the monetary value of the contract

- e.g., a BBB-rated bond that is worth \$100 today may, one year from now, be worth \$92 if the issuer is rated BB or \$104 if the issuer is rated A
- For simplicity, assume that value depends only on credit rating

Each counterparty loss (L) has a discrete distribution (F_L)

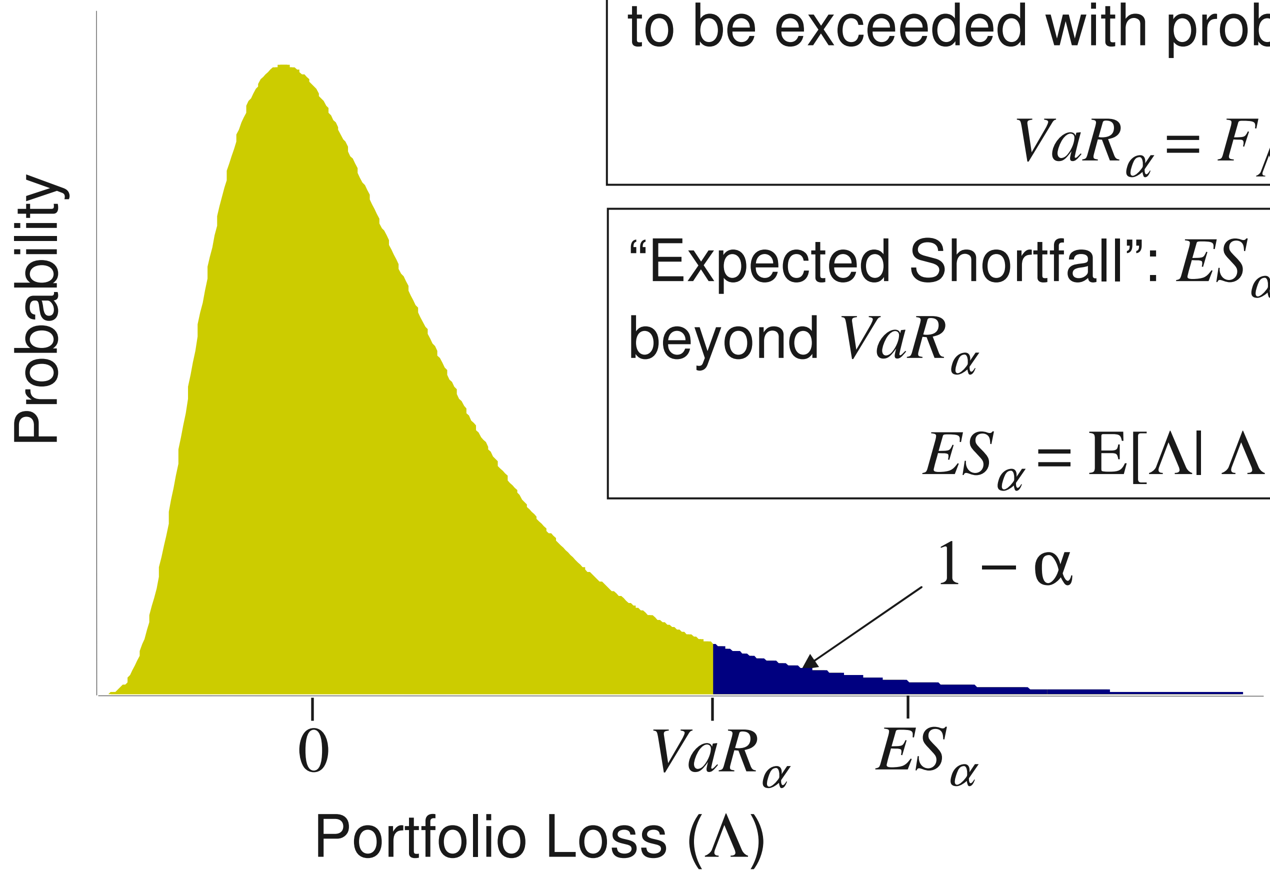
- e.g., for a BBB-rated counterparty

	AAA	AA	A	BBB	BB	B	CCC	Default
Loss per \$1	-0.07	-0.06	-0.04	0.00	0.08	0.20	0.45	1.00
Probability (%)	0.05	0.24	5.20	88.49	4.88	0.80	0.16	0.18

- Note that losses are positive and gains are negative

Credit Risk Measures

Portfolio loss distribution (F_Λ) is positively skewed with mode zero



"Value-at-Risk": VaR_α is the loss that is likely to be exceeded with probability $(1 - \alpha)$

$$VaR_\alpha = F_\Lambda^{-1}(\alpha)$$

"Expected Shortfall": ES_α is the average loss beyond VaR_α

$$ES_\alpha = \mathrm{E}[\Lambda | \Lambda \geq VaR_\alpha]$$

Credit Risk Optimization

We want to adjust the composition of the portfolio to "shrink" the right tail of the portfolio loss distribution

- Let x_j denote the size of the position in counterparty j
- Let L^j denote the loss in value per unit of counterparty j
- The loss for a portfolio of J counterparties is

$$\mathrm{L}\left(\boldsymbol{x}\right) = \sum_{j=1}^{J} L^j x_j \quad \leftarrow L^j\text{'s are co-dependent}$$

Minimize$_{\boldsymbol{x} \in \Omega}$ $g(\Lambda(\boldsymbol{x}))$ where g is

- VaR_α
- ES_α
- Variance
- Second moment, i.e., $\mathrm{E}[\Lambda(\boldsymbol{x})^2] = \mathrm{var}[\Lambda(\boldsymbol{x})] + \mathrm{E}[\Lambda(\boldsymbol{x})]^2$

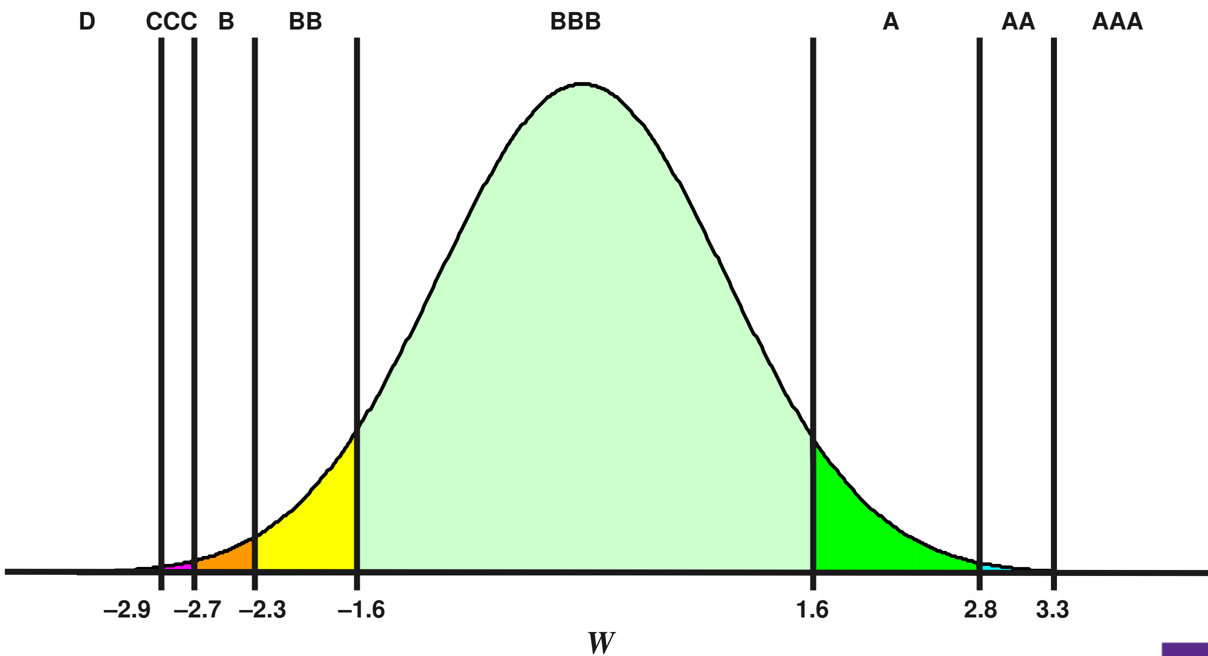
Algorithmics

Portfolio Credit Risk Model

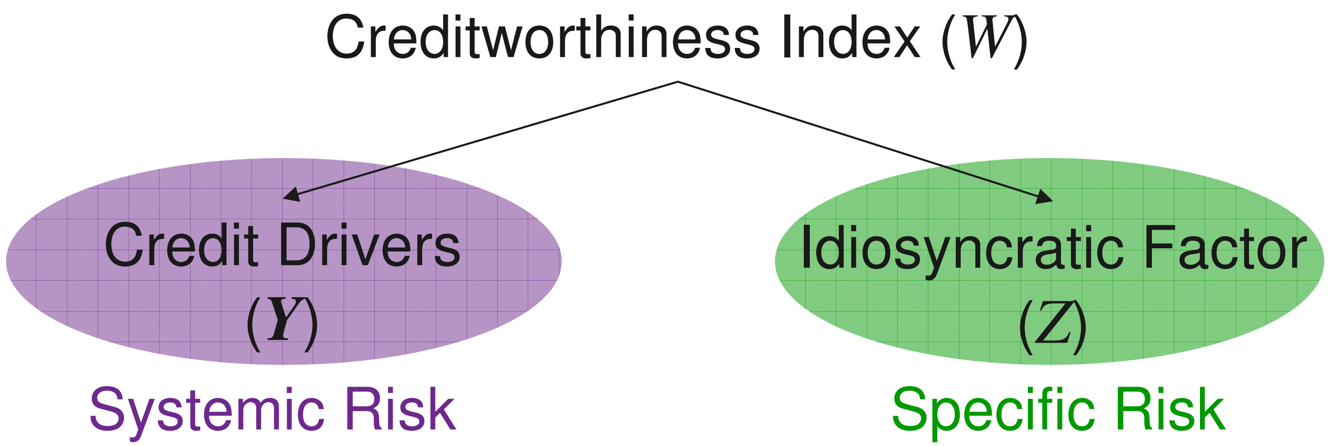
Structural Models of Portfolio Credit Risk

Structural models infer a counterparty's future credit state from a continuous random variable called a creditworthiness index (W)

- e.g., if $T_{BBB} \leq W < T_A$ then new credit state is BBB
  - Thresholds are chosen so that $P(T_{BBB} \leq W < T_A)$ is consistent with the credit transition matrix



Creditworthiness Index



Creditworthiness index of counterparty j :

$$W_j = \sum_{k=1}^{K} \beta_{jk} Y_k + \sigma_j Z_j$$

$N(0, 1)$

K credit drivers are correlated standard Normal variates with joint distribution function $F_{\boldsymbol{Y}}$

Algorithmics

Sampling Credit Drivers

Generate samples $\boldsymbol{y}_m$, $m = 1, \ldots, M$ from $F_{\boldsymbol{Y}}$

- Effect is to shift the transition probabilities for counterparties

	AAA	AA	A	BBB	BB	B	CCC	Default
Loss per $1	-0.07	-0.06	-0.04	0.00	0.08	0.20	0.45	1.00
Probability (%)	0.05	0.24	5.20	88.49	4.88	0.80	0.16	0.18
Probability | $\boldsymbol{y}$ (%)	0.01	0.05	1.73	83.59	10.83	2.41	0.57	0.79

- Creditworthiness indices are conditionally independent given $\boldsymbol{y}$

The portfolio loss distribution conditional on $\boldsymbol{y}_m$ is the convolution of the conditional counterparty loss distributions

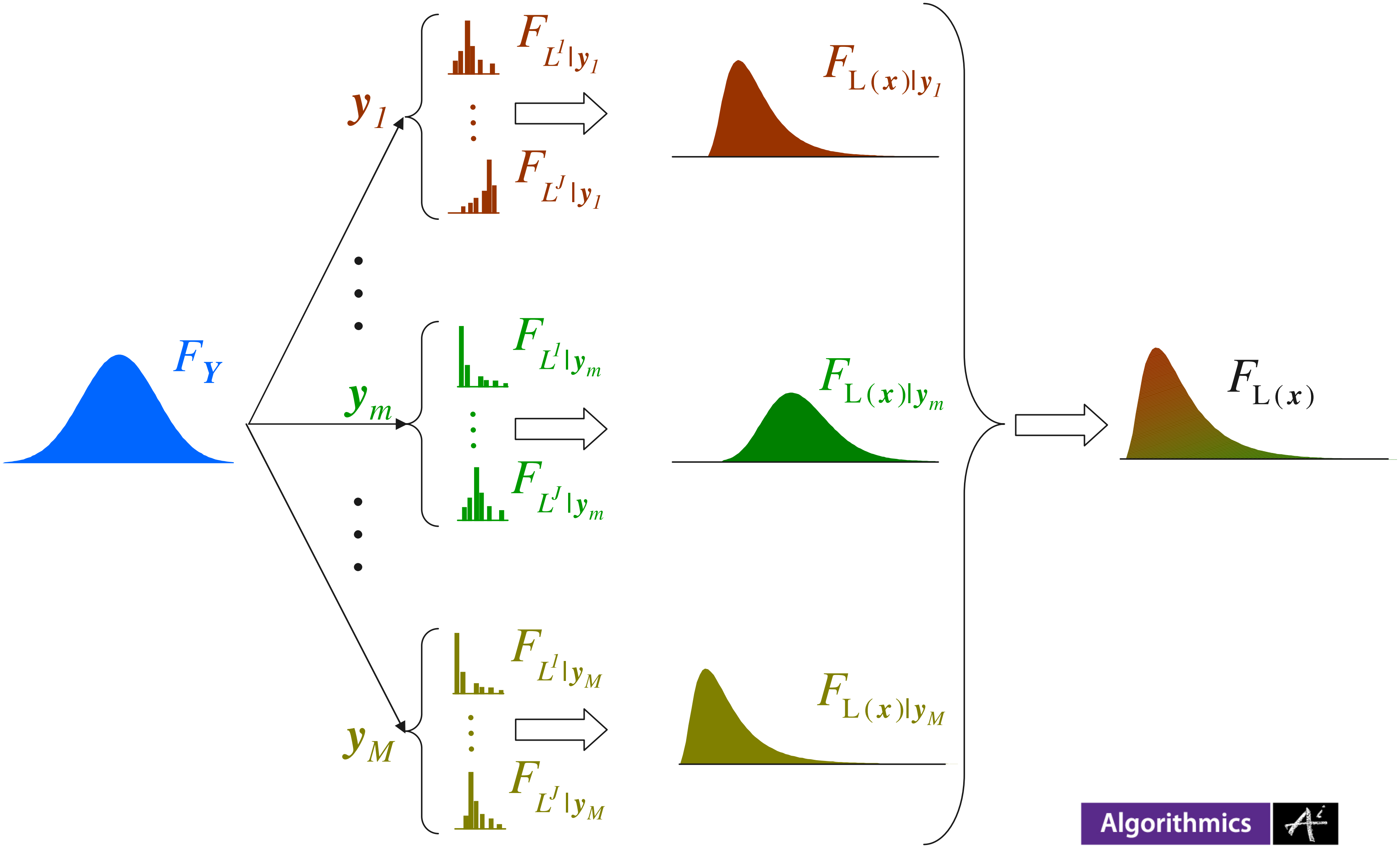
$$F_{\mathrm{L}(\boldsymbol{x})|\boldsymbol{y}} = F_{L^1 x_1|\boldsymbol{y}} * F_{L^2 x_2|\boldsymbol{y}} * \quad \cdots \quad * F_{L^J x_J|\boldsymbol{y}}$$

The unconditional portfolio loss distribution is the mixture of the conditional portfolio loss distributions

$$F_{\mathrm{L}(\boldsymbol{x})}(\ell) = \frac{1}{M} \sum_{m=1}^{M} F_{\mathrm{L}(\boldsymbol{x})|\boldsymbol{y}}(\ell)$$

Algorithmics

Conditional Independence Framework



Algorithmics

Optimization Challenges

Minimizing $\mathrm{E}[\Lambda(\boldsymbol{x})]$ or $\mathrm{var}[\Lambda(\boldsymbol{x})]$ is easy (compute unconditional means and covariances of counterparty losses from $F_{L|\mathbf{y}}$) but minimizing VaR_α or ES_α is more challenging

Formulating an optimization model using convolutions is not practical

- 8 credit states, J counterparties $\rightarrow$ 8^J possible portfolio losses for each $\boldsymbol{y}$

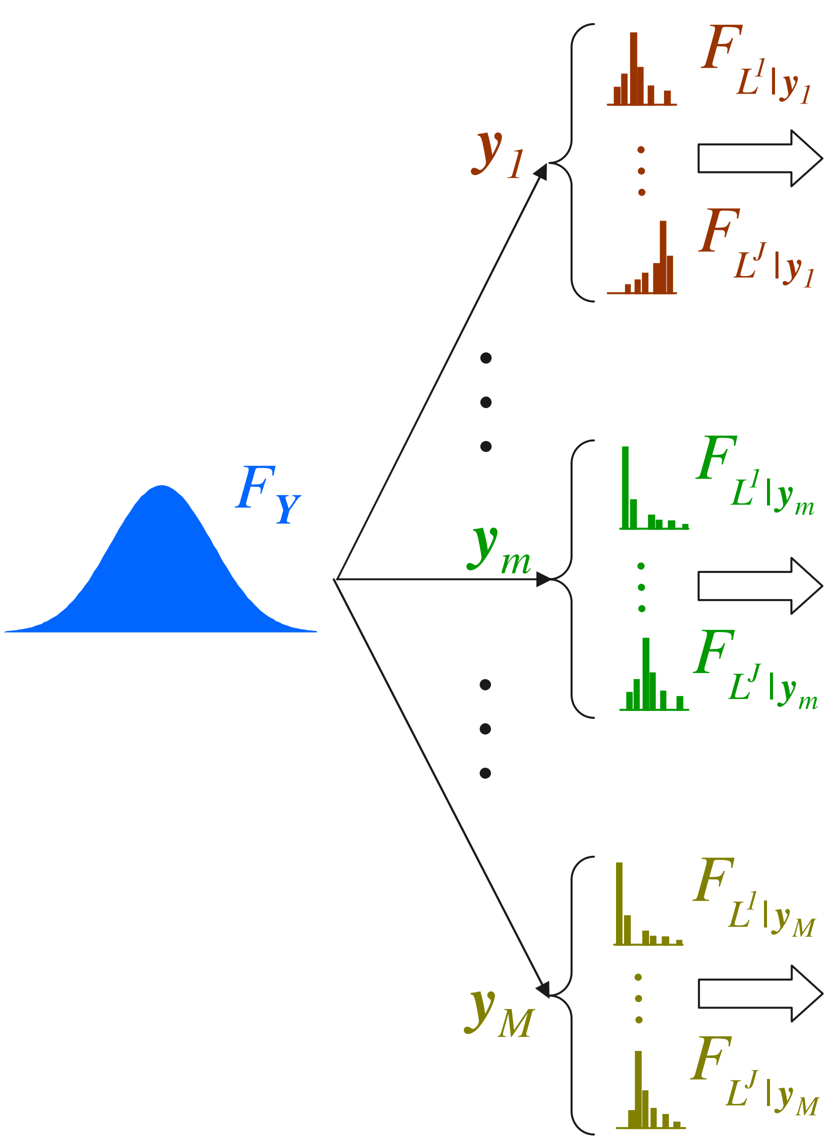
Consider approximations to the conditional loss distribution $F_{\Lambda(\boldsymbol{x})|\boldsymbol{y}}$

- Monte Carlo sampling
- Normal distribution
- Conditional mean

Algorithmics

Monte Carlo Sampling Approximation

Monte Carlo Sampling Approximation



	$L^1 \mid y_1$	…	$L^J \mid y_1$	$\Lambda(x) \mid y_1$
1	l^1_{11}		l^J_{11}	$\lambda_{11}(x)$
⋮				
N	l^1_{1N}		l^J_{1N}	$\lambda_{1N}(x)$

	$L^1 \mid y_m$	…	$L^J \mid y_m$	$\Lambda(x) \mid y_m$
1	l^1_{m1}		l^J_{m1}	$\lambda_{m1}(x)$
⋮				
N	l^1_{mN}		l^J_{mN}	$\lambda_{mN}(x)$

	$L^1 \mid y_M$	…	$L^J \mid y_M$	$\Lambda(x) \mid y_M$
1	l^1_{M1}		l^J_{M1}	$\lambda_{M1}(x)$
⋮				
N	l^1_{MN}		l^J_{MN}	$\lambda_{MN}(x)$

$$\boxed{\ell_{mn}(x) = \sum_{i=1}^{J} l^j_{mn} x_j}$$

$$\boxed{\#\text{ data} \approx NMJ}$$

$\Lambda(x)$
$\lambda_{11}(x)$
⋮
$\lambda_{1N}(x)$
⋮
$\lambda_{m1}(x)$
⋮
$\lambda_{mN}(x)$
⋮
$\lambda_{M1}(x)$
⋮
$\lambda_{MN}(x)$

Algorithmics

Estimating VaR and ES from Samples

e.g., 100 random samples (each has probability 0.01) sorted in increasing sequence

1	2	3	4	5	6		95	96	97	98	99	100
-400	-350	-300	-225	-150	-100	…	825	850	875	900	950	1100

- $VaR_{0.95}$ = 850 is the fifth-largest observation
- $ES_{0.95}$ = 935 is the average of the five largest observations

$$
\begin{aligned}
ES_{0.95} &= \frac{1}{5}(850+875+900+950+1100) \\
&= \underbrace{850}_{VaR_{0.95}} + \frac{1}{5}\underbrace{(0+25+50+100+250)}_{VaR_{0.95}\text{ exceedance}}
\end{aligned}
$$

Algorithmics

Monte Carlo Optimization Models

ES_α can be minimized with linear programming

Rockafellar, R. T. and S. Uryasev (2000), "Optimization of conditional Value at Risk," *The Journal of Risk* 2(3), 21-41

$$\min_{\boldsymbol{x}\in\Omega} \quad z + \frac{1}{MN(1-\alpha)}\sum_{i=1}^{MN}\left[\ell_i(\boldsymbol{x}) - z\right]^+$$

$$\boxed{\text{Recall}: \quad ES_{0.95} = 850 + \frac{1}{5}(0+25+50+100+250)}$$

VaR_α minimization is an integer program (MN binary variables)

- Use a heuristic approach based on successive ES_α optimization
- Iteratively fix the samples in the tail of the distribution

Larsen, N., Mausser H., and S. Uryasev (2002), "Algorithms for Optimization of Value-at-Risk," in *Financial Engineering, e-commerce and Supply Chain*, P. Pardalos and V.K. Tsitsiringos (Eds.), 129-157.

Algorithmics

Normal Approximation

Central Limit Theorem (CLT)

If the number of counterparties is large and contracts are relatively small then the conditional portfolio loss distribution is close to Normal

$$\mathrm{L}(\boldsymbol{x}) \mid \boldsymbol{y}_m \xrightarrow{D} \mathrm{N}\left(\sum_{j=1}^{J} \mu_{L^j \mid \boldsymbol{y}_m} x_j, \sum_{j=1}^{J} \sigma^2_{L^j \mid \boldsymbol{y}_m} x_j^2\right) \equiv \mathrm{N}\left(\mu_m(\boldsymbol{x}), \sigma_m^2(\boldsymbol{x})\right)$$

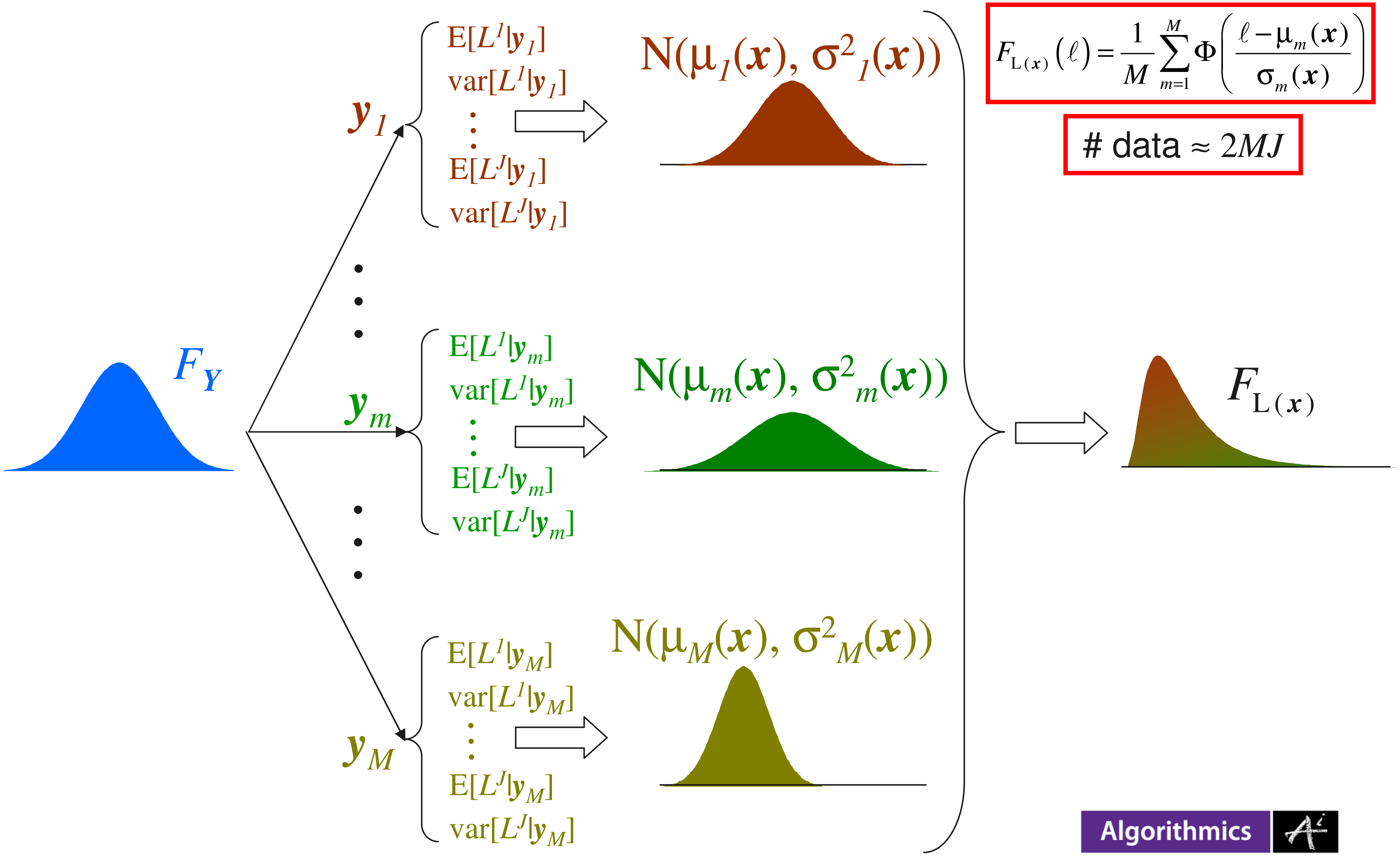
- Need the size restriction (Lyapunov or Lindeberg condition) because conditional counterparty losses are independent but not identically distributed

Portfolio loss distribution is

$$F_{\mathrm{L}(\boldsymbol{x})}(\ell) = \frac{1}{M} \sum_{m=1}^{M} \Phi\left(\frac{\ell - \mu_m(\boldsymbol{x})}{\sigma_m(\boldsymbol{x})}\right)$$

Algorithmics

Normal (CLT) Approximation



CLT Optimization Models

VaR_α minimization is a (non-convex*) non-linear program

$$\begin{aligned} \min_{x\in\Omega} \quad & \ell(\boldsymbol{x}) \\ \text{s.t.} \quad & \frac{1}{M}\sum_{m=1}^{M}\Phi\left(\frac{\ell(\boldsymbol{x})-\mu_m(\boldsymbol{x})}{\sigma_m(\boldsymbol{x})}\right)=\alpha \end{aligned}$$

ES_α minimization is a non-linear program

$$\begin{aligned} \min_{x\in\Omega} \quad & \frac{1}{M(1-\alpha)}\sum_{m=1}^{M}\left[\mu_m(\boldsymbol{x})\left(1-\Phi\left(\frac{\ell(\boldsymbol{x})-\mu_m(\boldsymbol{x})}{\sigma_m(\boldsymbol{x})}\right)\right)+\sigma_m(\boldsymbol{x})\phi\left(\frac{\ell(\boldsymbol{x})-\mu_m(\boldsymbol{x})}{\sigma_m(\boldsymbol{x})}\right)\right] \\ \text{s.t.} \quad & \frac{1}{M}\sum_{m=1}^{M}\Phi\left(\frac{\ell(\boldsymbol{x})-\mu_m(\boldsymbol{x})}{\sigma_m(\boldsymbol{x})}\right)=\alpha \end{aligned}$$

Conditional Mean Approximation

Conditional Mean (CM)

If the portfolio comprises an extremely large number of almost identical contracts then the conditional portfolio loss is approximated by the sum of the conditional mean counterparty losses

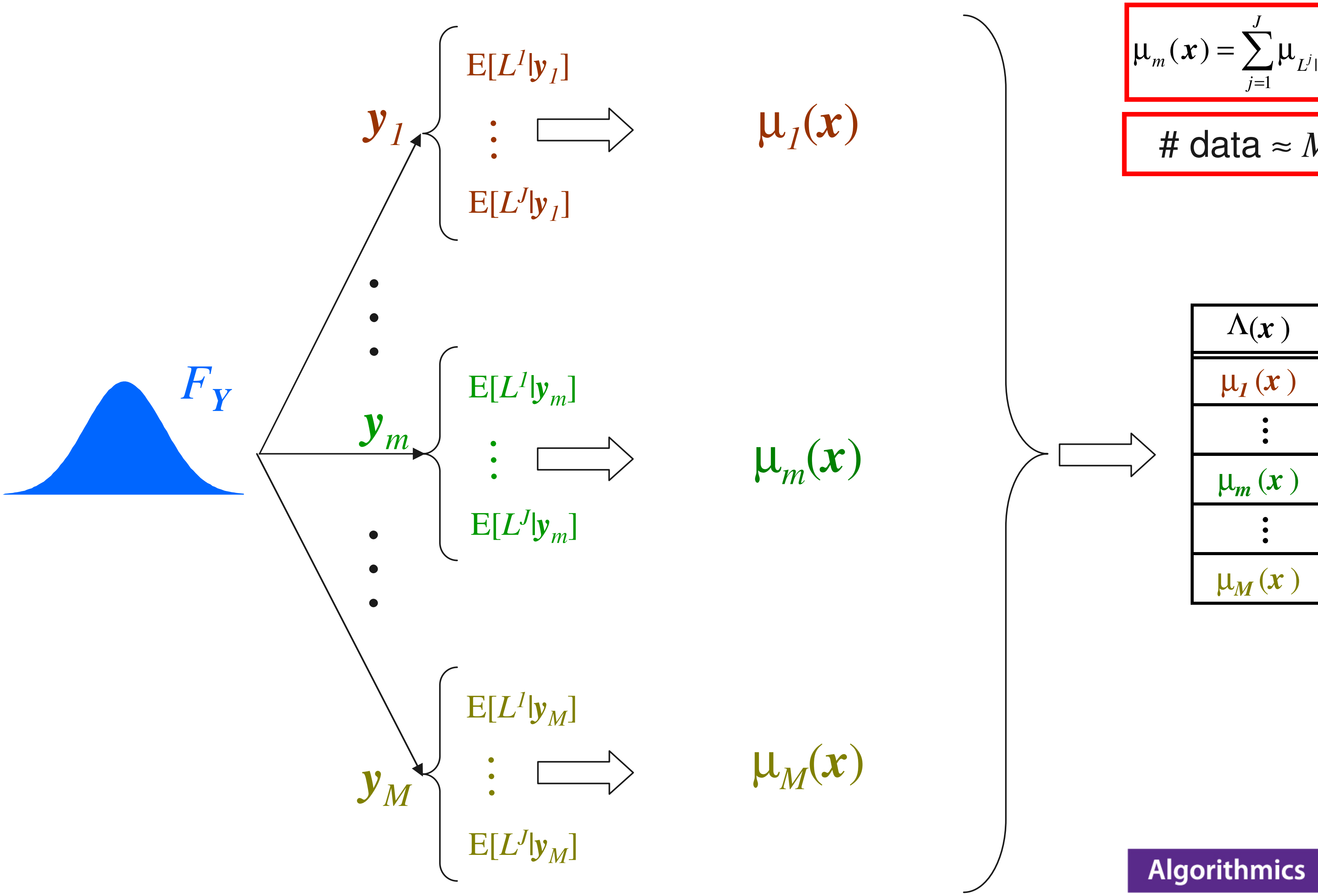
$$\mathrm{L}(\boldsymbol{x}) \mid \boldsymbol{y}_m \approx \sum_{j=1}^{J} \mu_{L^j \mid \boldsymbol{y}_m} x_j \equiv \mu_m(\boldsymbol{x})$$

Assume: diversification eliminates all specific risk

Portfolio loss distribution is approximated by a sample of size M

- Optimization models are same as those for Monte Carlo sampling

Conditional Mean (CM) Approximation



$$\boxed{\mu_m(\boldsymbol{x}) = \sum_{j=1}^{J} \mu_{L^j|\mathbf{y}_m} x_j}$$

$$\boxed{\#\ \text{data} \approx MJ}$$

Algorithmics

Computational Results

Test Portfolio

3000 counterparties and 50 credit drivers (from ISDA/IACPM 2006)

- Credit drivers are industry/country indices
- Each counterparty depends on one credit driver ($0.42 \leq \beta \leq 0.65$)
- Initial contract values are identical

Consider individual counterparties and groups

- Can be impractical to take action at counterparty level
- Counterparties maintain their initial weightings within groups
- Grouping is done at random
  - 10 groups of 300
  - 50 groups of 60
  - 300 groups of 10
  - 3000 groups of 1

Formulations

	$VaR_{0.999}$	$ES_{0.999}$	Variance, 2nd Moment
MC Sampling (N = 1, 20)	Linear (Heuristic)	Linear	
Normal Approximation	Non-convex* Non-linear	Convex Non-linear	
Cond. Mean Approximation	Linear (Heuristic)	Linear	
Unconditional			Convex Quadratic

Constraints (Ω)

- Maintain initial value of portfolio
- Earn at least the initial expected return
- Trading limits [0, 2] for each counterparty
  - Can eliminate or double the initial position

Algorithmics

Methodology

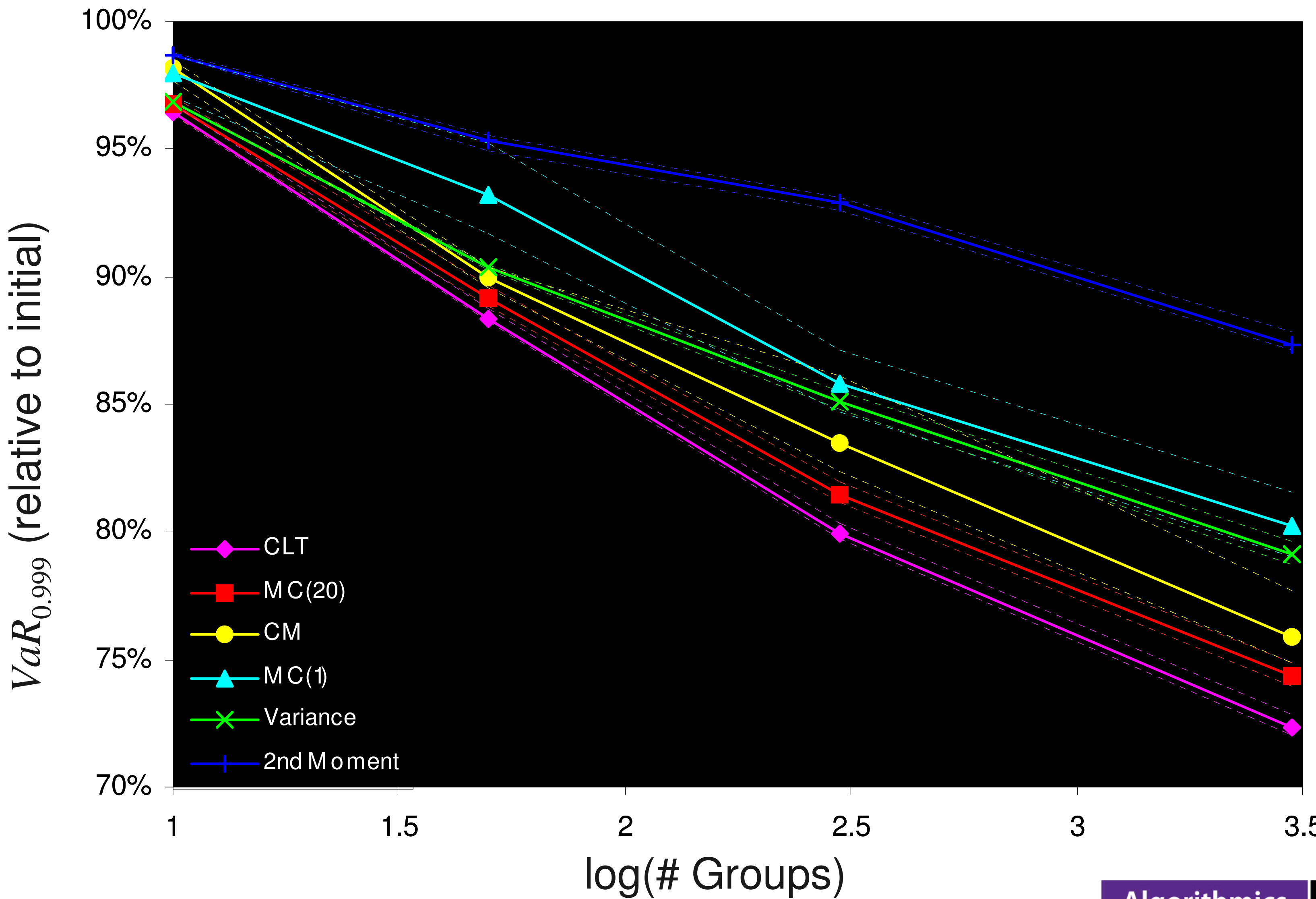
Perform 5 trials, each with $M = 10{,}000$ credit driver samples

- Report the average over 5 trials

Evaluate optimal portfolios by computing $VaR_{0.999}$ and $ES_{0.999}$

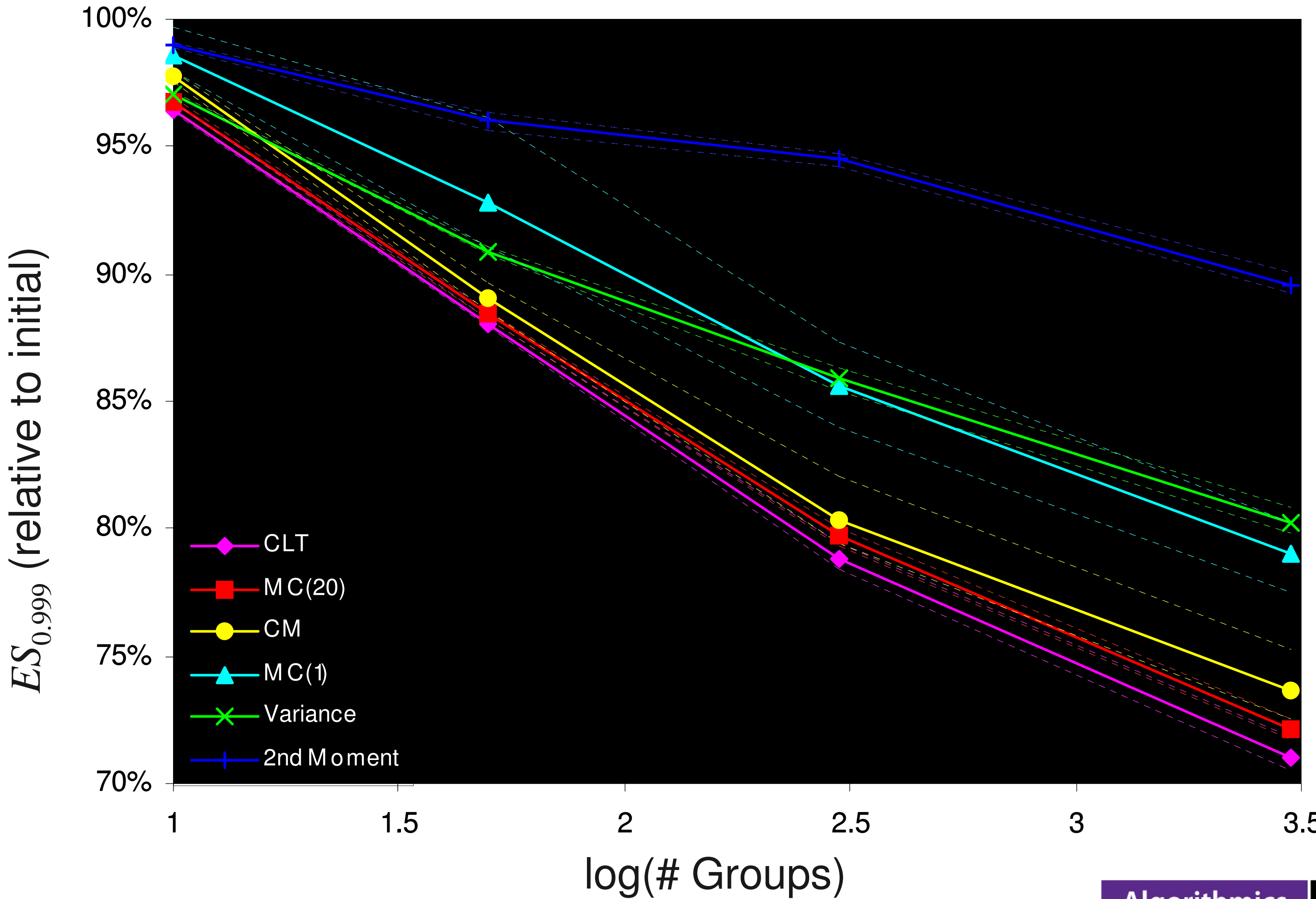
- Out-of-Sample
  - M = 6,000,000, N =1 (assume to be the true loss distribution)
  - Determine effects of systemic sampling error and model approximation error
- In-sample
  - N = 150 (assume to be the true conditional loss distribution)
  - Isolate effects of model approximation error

Out-of-Sample *VaR*



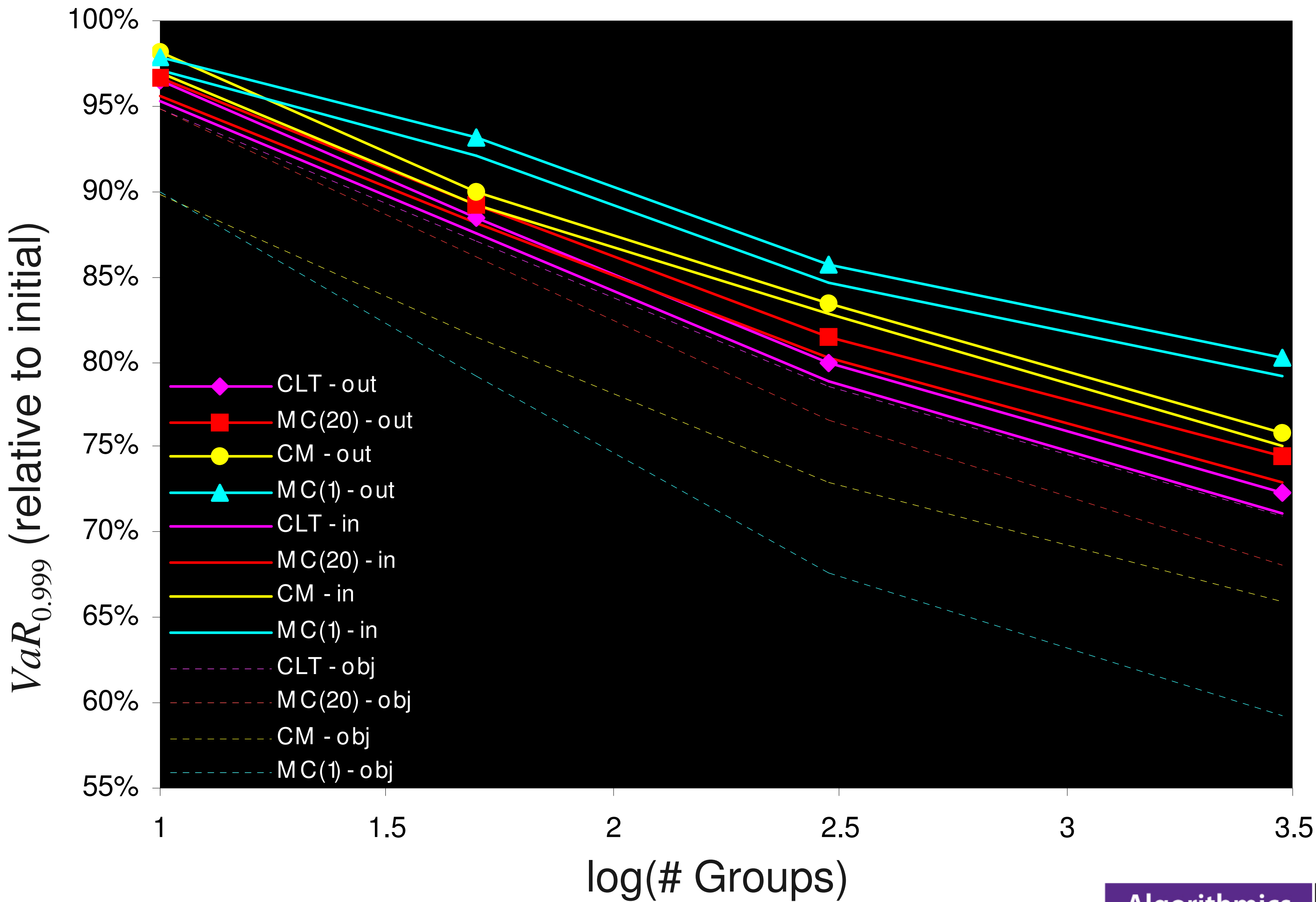
Algorithmics

Out-of-Sample *ES*



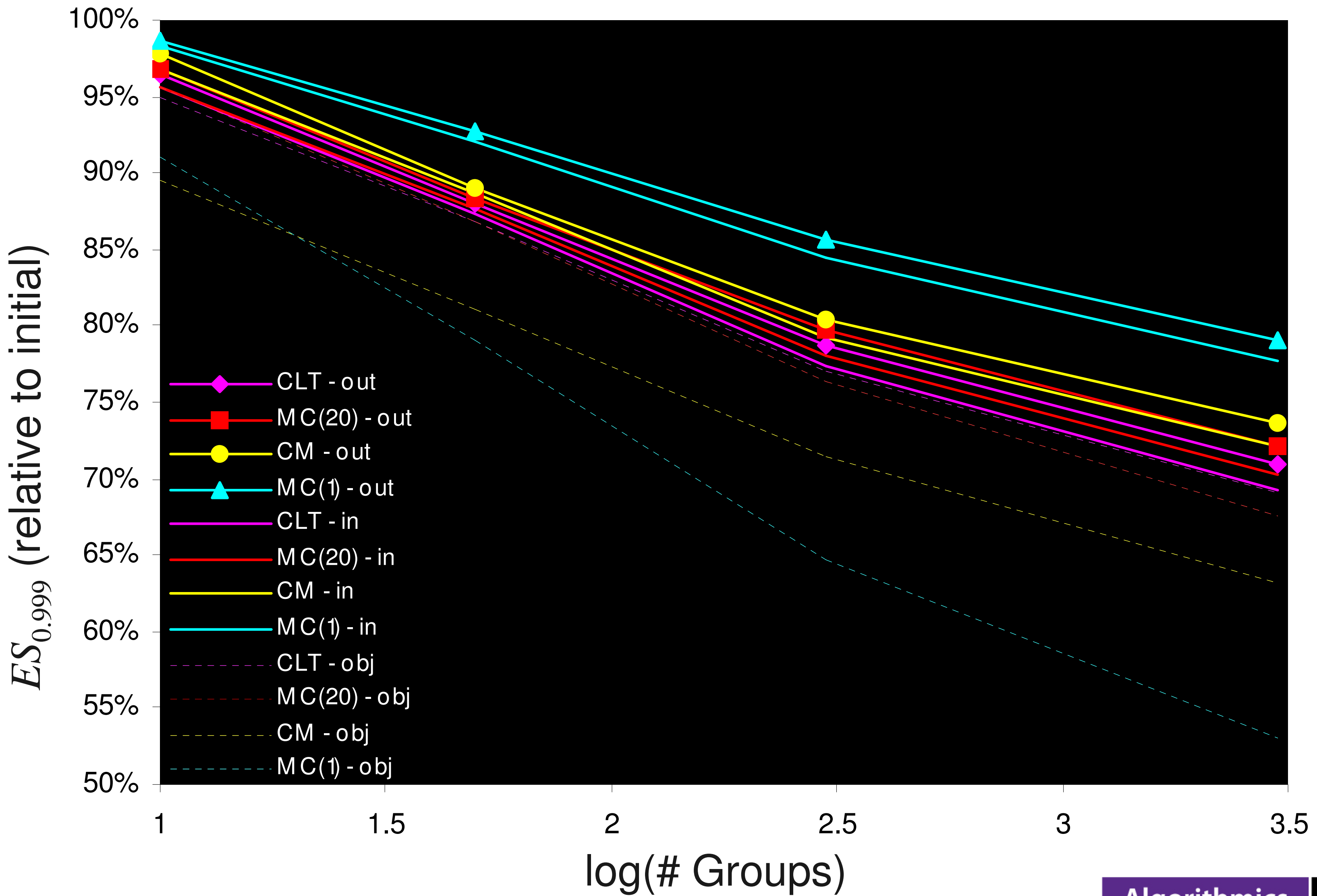
Algorithmics

Approximation Quality for *VaR*



Algorithmics

Approximation Quality for *ES*



Algorithmics

Granularity Effects

What happens as the portfolio becomes more granular (smallness condition is violated)? e.g., 50 groups with wider trading limits

$VaR_{0.999}$	Trading limits [0, 2]			Trading limits [-3, 15]		
	(Out / In) - 1	(In / Obj) - 1	HHI	(Out / In) - 1	(In / Obj) - 1	HHI
CLT	0.90%	0.66%	0.0345	0.76%	2.70%	0.1556
MC(20)	1.22%	2.25%	0.0322	1.15%	3.77%	0.1311
CM	0.92%	9.54%	0.0385	-0.27%	267.81%	0.7456
MC(1)	1.15%	16.38%	0.0345	0.25%	35.63%	0.1712

$ES_{0.999}$	Trading limits [0, 2]			Trading limits [-3, 15]		
	(Out / In) - 1	(In / Obj) - 1	HHI	(Out / In) - 1	(In / Obj) - 1	HHI
CLT	0.82%	0.61%	0.0342	0.65%	3.10%	0.1624
MC(20)	0.95%	0.83%	0.0338	0.94%	3.14%	0.1296
CM	0.53%	9.37%	0.0399	-0.13%	294.00%	0.7482
MC(1)	0.71%	16.48%	0.0363	0.09%	52.47%	0.1729

Approximations to the conditional distribution get worse, especially for CM

- HHI is the Herfindahl-Hirschman Index

Algorithmics

Systemic Sampling Effects

How does the number of systemic samples affect out-of-sample performance?

$VaR_{0.999}$	10,000 Systemic Samples			50,000 Systemic Samples		
	10 Groups	50 Groups	300 Groups	10 Groups	50 Groups	300 Groups
CLT	96.5%	88.4%	80.0%	96.3%	88.3%	79.6%
MC(20), (4)	96.7%	89.2%	81.4%	96.6%	89.2%	81.3%
CM	98.2%	90.0%	83.4%	97.4%	89.5%	82.1%
MC(1)	97.9%	93.2%	85.8%	97.1%	90.4%	82.9%

$ES_{0.999}$	10,000 Systemic Samples			50,000 Systemic Samples		
	10 Groups	50 Groups	300 Groups	10 Groups	50 Groups	300 Groups
CLT	96.5%	88.1%	78.8%	96.4%	87.9%	78.4%
MC(20), (4)	96.7%	88.4%	79.7%	96.6%	88.5%	79.3%
CM	97.8%	89.1%	80.4%	97.6%	89.4%	79.7%
MC(1)	98.6%	92.8%	85.6%	96.9%	89.5%	80.8%

Slight improvement for MC(1), negligible for others

- CLT with 10,000 systemic samples does better than other models with 50,000 systemic samples

Performance

		$ES_{0.999}$			
Model	Solver	10 grp	50 grp	300 grp	3000 grp
CLT	IPOPT	4 - 8	6 - 8	14 - 83	181 - 1090
CM	CPLEX	1	1 - 2	6 - 8	73 - 86
MC(1)	CPLEX	1	1 - 2	6 - 10	14 - 115
MC(20)	CPLEX	137 - 155	233 - 279	461 - 578	1050 - 1280

		$VaR_{0.999}$			
Model	Solver	10 grp	50 grp	300 grp	3000 grp
CLT	IPOPT	4 - 25	5 - 7	14 - 55	400 - 1643
CM	CPLEX	2 - 3	6 - 8	46 - 50	791 - 1025
MC(1)	CPLEX	2 - 3	8 - 10	46 - 69	2436 - 3312
MC(20)*	CPLEX	3620 - 4080	2382 - 2777	6522 - 8563	39273 - 86383

		Variance			
Model	Solver	10 grp	50 grp	300 grp	3000 grp
Uncond	MOSEK	< 1	< 1	1	682 - 719

Elapsed time (sec)

Server : 8 x Opteron 885 CPU, 16 cores (jobs run on 1 core), 64 Gb RAM

* VaR optimization for MC(20) was run in parallel mode on 4 threads

Conclusions

Normal approximation is attractive for optimization

- Consistently better than Monte Carlo sampling with only 10% of the data
- Acceptable performance solving non-linear model
- Relatively robust to violations of smallness condition

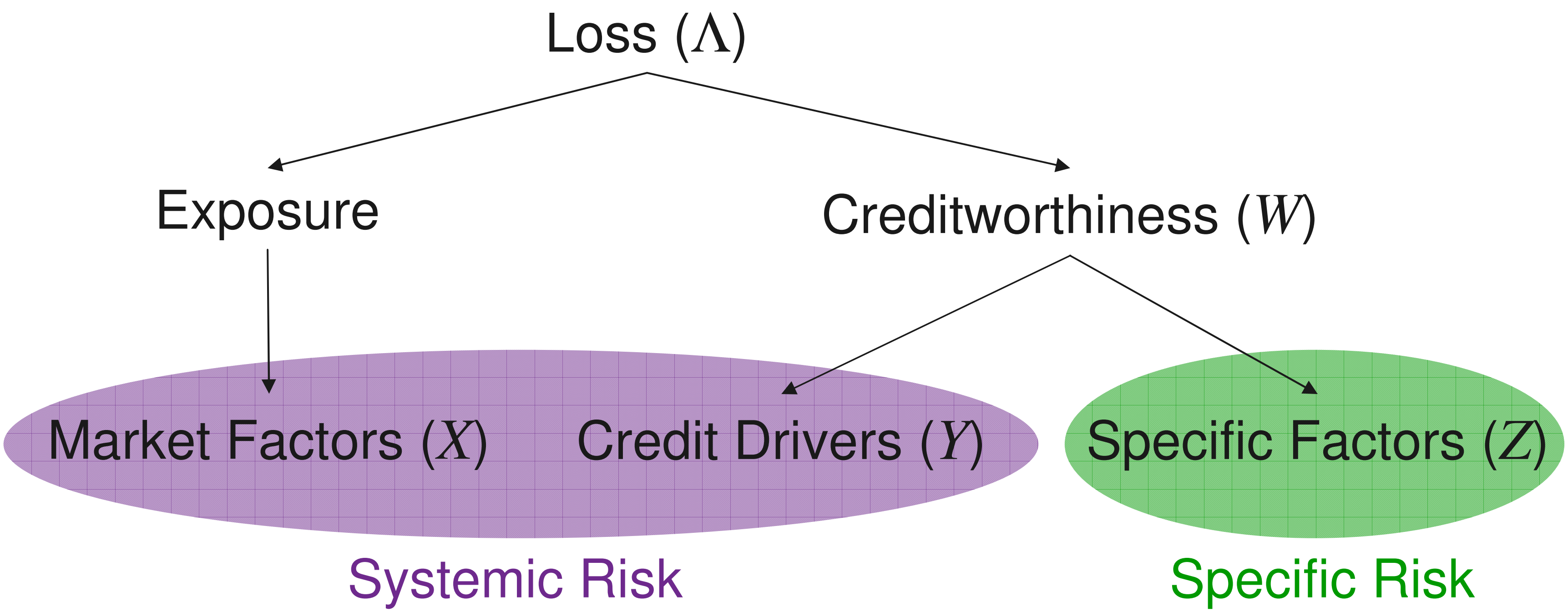
Tests with more realistic counterparty groupings yield consistent results

Further work:

- Improve VaR for Monte Carlo sampling
- Vary credit driver sensitivities, quantiles

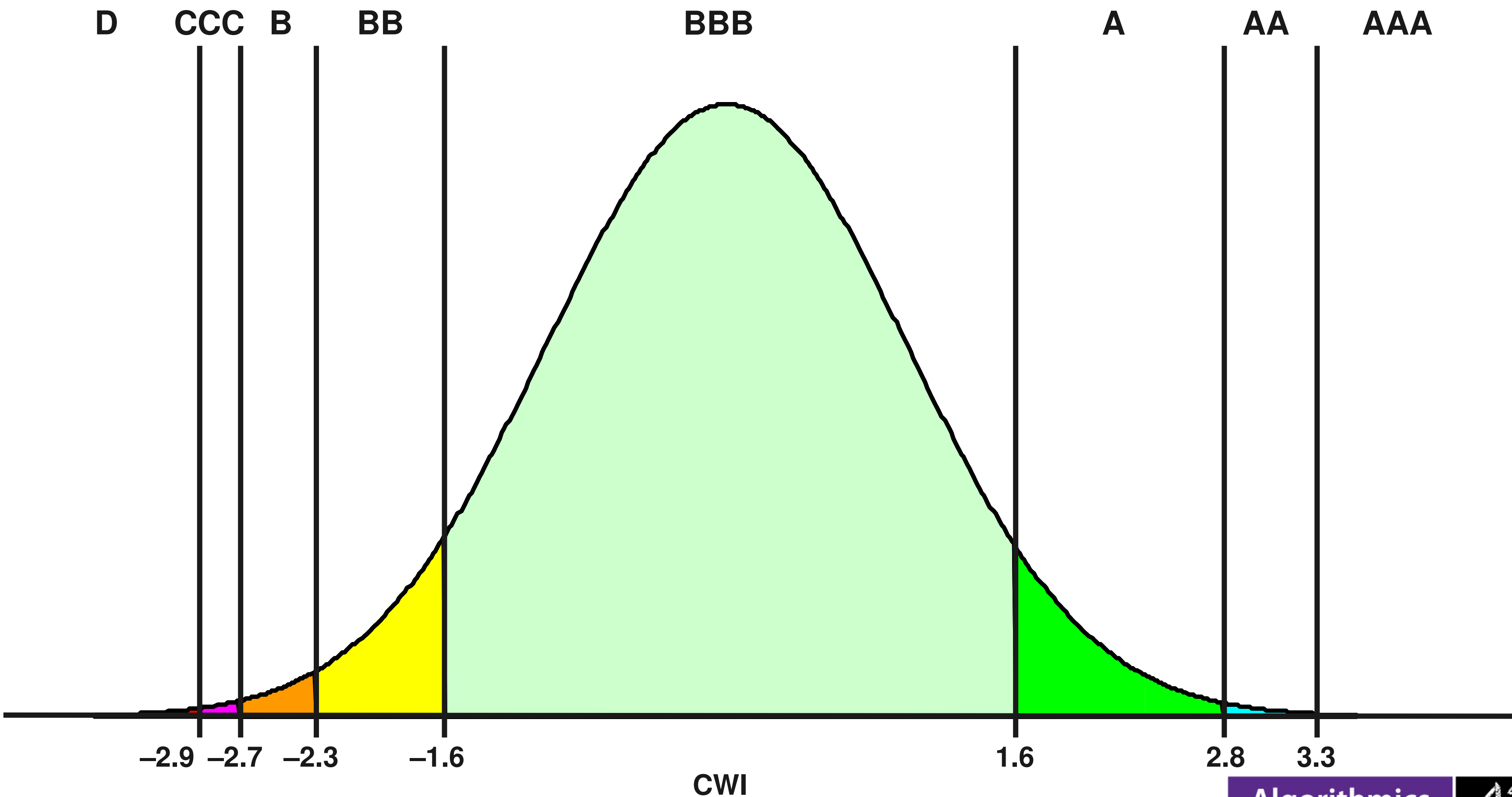
Algorithmics

Integrated Market-Credit Loss Model



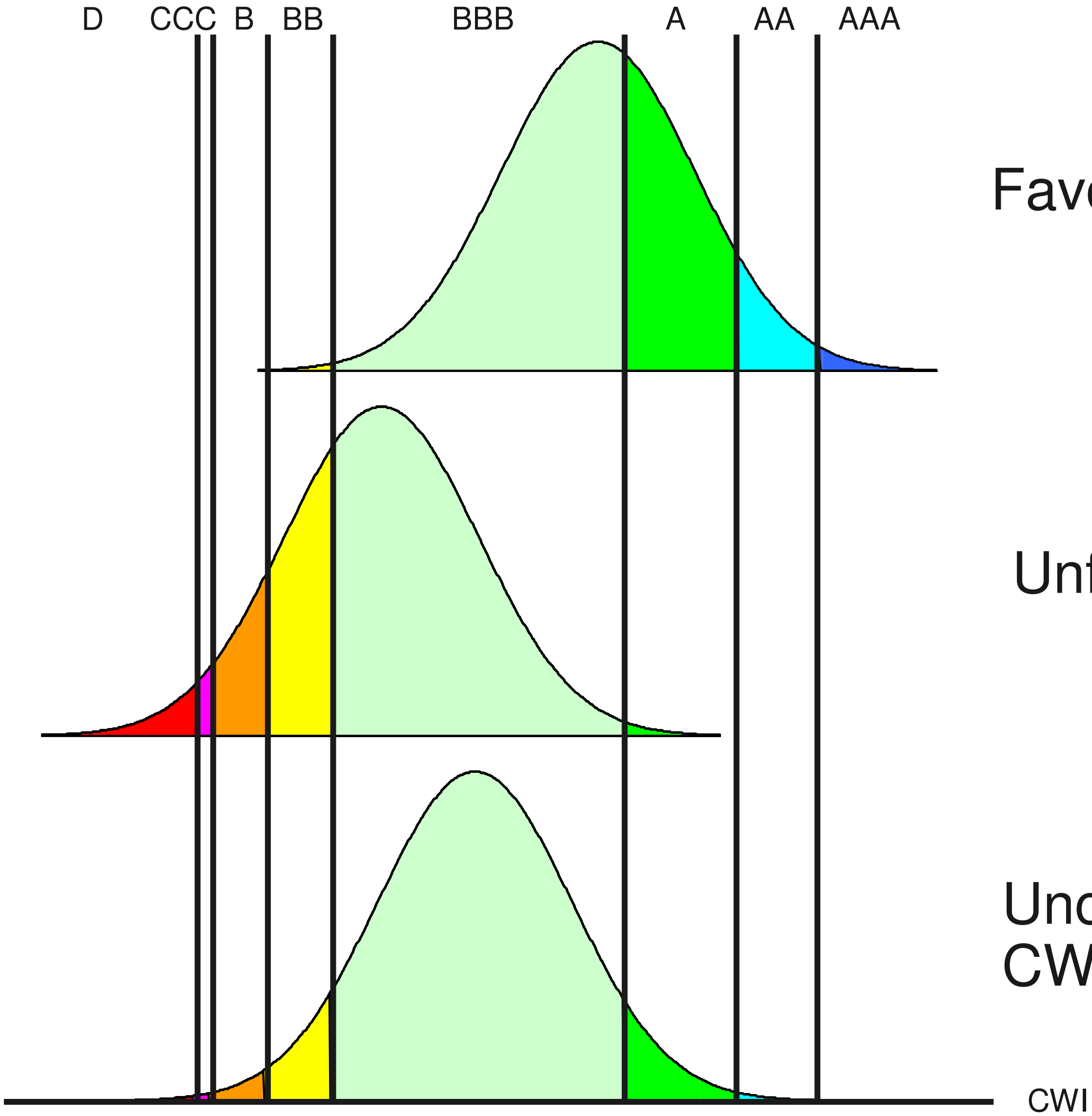
Creditworthiness Index and Transitions

	Default	CCC	B	BB	BBB	A	AA	AAA
Probability (%)	0.18	0.16	0.80	4.88	88.49	5.20	0.24	0.05
Value of $1	0.00	0.55	0.80	0.92	1.00	1.04	1.06	1.07
Loss per $1	1.00	0.45	0.20	0.08	0.00	-0.04	-0.06	-0.07



Algorithmics

Conditional Transition Probabilities for BBB



Favourable systemic

Unfavourable systemic

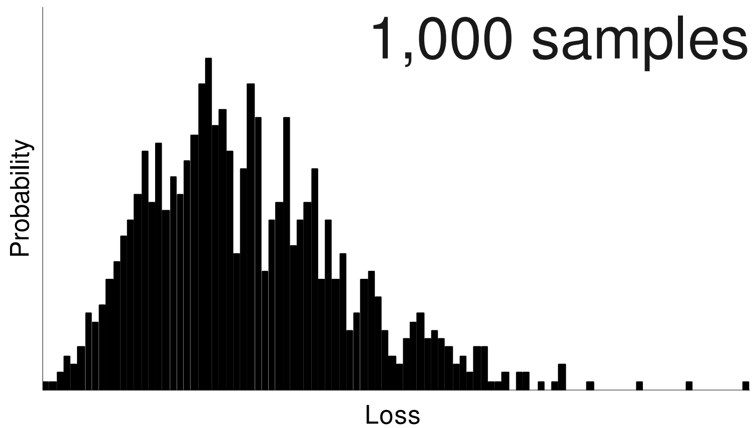
Unconditional (calibrate CWI thresholds)

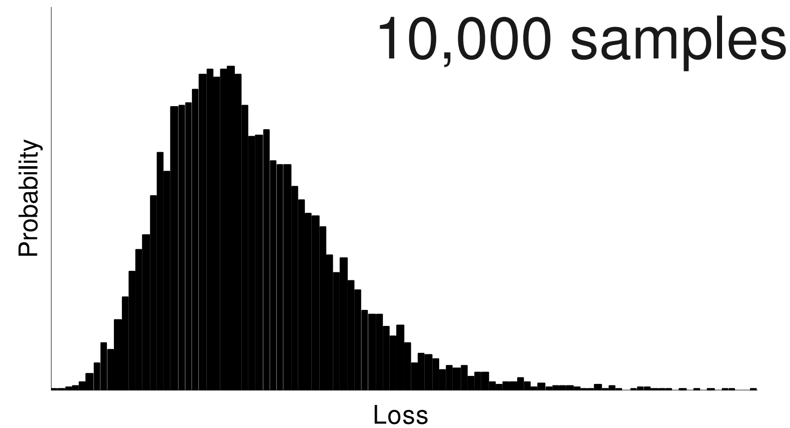
Algorithmics

Number of Samples

For α close to 1, we need a lot of samples to get good estimates of VaR_α and ES_α

- $\alpha \geq 0.995$ is common for credit risk





Possible to reduce the number of samples by “careful” selection?

Algorithmics

Monte Carlo Sampling Optimization Models

VaR_α

$$
\begin{aligned}
&\min_{x\in\Omega} \quad z \\
&\text{s.t.} \\
&\qquad \sum_{j=1}^{J} l_i^j x_j - z - Bd_i \le 0 \qquad \mathit{for} \quad i = 1,\dots,MN \\
&\qquad \sum_{i=1}^{m} d_i \le MN(1-\alpha) \\
&\qquad d_i \in \{0,1\} \qquad \mathit{for} \quad i = 1,\dots,MN
\end{aligned}
$$

ES_α

$$
\begin{aligned}
&\min_{x\in\Omega} \quad z + \frac{1}{MN\left(1-\alpha\right)} \sum_{i=1}^{MN} y_i \\
&\text{s.t.} \\
&\qquad \sum_{j=1}^{J} l_i^j x_j - z - y_i \le 0 \qquad \mathit{for} \quad i = 1,\dots,MN
\end{aligned}
$$

VaR Minimization Heuristic

Step 0. Initialization

1. Set $\alpha_0 = \alpha$, $k = 0$, $H_0 = \{s : s = 1, \ldots, M\}$.
2. Assign value to the parameter for discarding scenarios ε, $0 < \varepsilon < 1$.

Step 1. Optimization sub-problem

1. Minimize α_k-CVaR

$$
\begin{array}{lll}
\min\limits_{x,z,\ell,\gamma} & \ell + \nu_k \sum\limits_{s \in H_k} \pi_s z_s & \\
\text{s.t.} & \sum_i \mu_{i,s} x_i \leq \ell + z_s,\ z_s \geq 0 & s \in H_k, \\
& \sum_i \mu_{i,s} x_i \leq \gamma & s \in H_k, \\
& \sum_i \mu_{i,s} x_i \geq \gamma & s \notin H_k, \\
& \sum_i x_i = 1 & \\
& \sum_i r_i x_i \geq R & \\
& x_i - x_i^0 \leq y_i, & i = 1, \ldots, N \\
& x_i^0 - x_i \leq y_i, & i = 1, \ldots, N \\
& \sum_i y_i \leq \triangle x & \\
& \underline{x}_i \leq x_i \leq \overline{x}_i, & i = 1, \ldots, N
\end{array}
$$

   where $\nu_k = 1/((1-\alpha_k)M)$. Denote the optimal solution of this problem by x_k^*.

2. Order the scenarios $y_s x_k^*$, $s = 1, \ldots, M$ in ascending order and denote ordered scenarios by s_j, $j = 1, \ldots, M$.

Step 2. Estimating VaR
Calculate VaR estimate $j_k = y_{j(\alpha)} x_k^*$, where $j(\alpha) = \min\{j : j/M \geq \alpha\}$.

Step 3. Stopping and re-initialization

1. $k = k + 1$.
2. $b_k = \alpha + (1-\alpha)(1-\varepsilon)^k$ and $\alpha_k = \alpha / b_k$.
3. $H_k = \{s_j \in H_{k-1} : j/M \leq b_k\}$.
4. If $H_k = H_{k-1}$ then stop the algorithm and return the estimate of the VaR-optimal portfolio x_k^* and VaR ℓ_k, otherwise go to Step 1.

VaR Optimization Alternatives

Convex Approximations

- Assume some structure in the uncertainty

Bertsimas, D. and M. Sim (2004), "The Price of Robustness," *Operations Research* 52(1), 35-53.

Nemirovski, A. and A. Shapiro (2006), "Convex Approximations of Chance Constrained Programs," *Siam Journal on Optimization* 17(4), 969-996.

Worst-Case Scenario

- No assumptions about uncertainty structure

Calafiore, G. and M.C. Campi (2006), "The Scenario Approach to Robust Control Design," *IEEE Transactions on Automatic Control* 51(5), 742-753.

ES_α Objective for Normal Approximation

$$\mathrm{L}(\boldsymbol{x}) \mid \boldsymbol{y}_m \equiv \mathrm{L}_m(\boldsymbol{x}) \sim \mathrm{N}\left(\mu_m(\boldsymbol{x}), \sigma_m^2(\boldsymbol{x})\right)$$

$$\mathrm{E}\left[\mathrm{L}(\boldsymbol{x}) \mid \mathrm{L}(\boldsymbol{x}) \geq VaR_\alpha\right] = \frac{1}{M(1-\alpha)} \sum_{m=1}^{M} \mathrm{E}\left[\mathrm{L}_m(\boldsymbol{x}) \times 1\left\{\mathrm{L}_m(\boldsymbol{x}) \geq VaR_\alpha\right\}\right]$$

$$\mathrm{E}\left[\mathrm{L}_m(\boldsymbol{x}) \mid \mathrm{L}_m(\boldsymbol{x}) \geq \ell\right]$$

$$= \mathrm{E}\left[\left(\mu_m(\boldsymbol{x}) + \sigma_m(\boldsymbol{x})Z\right) \times 1\left\{Z \geq \frac{\ell - \mu_m(\boldsymbol{x})}{\sigma_m(\boldsymbol{x})}\right\}\right]$$

$$= \mu_m(\boldsymbol{x})\left(1 - \Phi\left(\frac{\ell - \mu_m(\boldsymbol{x})}{\sigma_m(\boldsymbol{x})}\right)\right) + \sigma_m(\boldsymbol{x}) \int_{\frac{\ell - \mu_m(\boldsymbol{x})}{\sigma_m(\boldsymbol{x})}}^{\infty} Z \frac{e^{-Z^2/2}}{\sqrt{2\pi}} dZ$$

$$= \mu_m(\boldsymbol{x})\left(1 - \Phi\left(\frac{\ell - \mu_m(\boldsymbol{x})}{\sigma_m(\boldsymbol{x})}\right)\right) + \sigma_m(\boldsymbol{x})\left[-\frac{e^{-Z^2/2}}{\sqrt{2\pi}}\right]_{\frac{\ell - \mu_m(\boldsymbol{x})}{\sigma_m(\boldsymbol{x})}}^{\infty}$$

$$= \mu_m(\boldsymbol{x})\left(1 - \Phi\left(\frac{\ell - \mu_m(\boldsymbol{x})}{\sigma_m(\boldsymbol{x})}\right)\right) + \sigma_m(\boldsymbol{x})\phi\left(\frac{\ell - \mu_m(\boldsymbol{x})}{\sigma_m(\boldsymbol{x})}\right)$$

Conditional Mean Motivation

Chebyshev inequality (basis of LLN)

$$P\left(\left|\frac{S_n}{n}-\mu\right|<\varepsilon\right)\geq 1-\frac{\sigma^2}{n\varepsilon^2} \quad \rightarrow \quad P\left(\left|S_n-n\mu\right|<n\varepsilon\right)\geq 1-\frac{\sigma^2}{n\varepsilon^2}$$

For non-iid, Kolmogorov criterion requires $\sum_{n=1}^{\infty}\frac{\sigma_n^2}{n}<\infty$

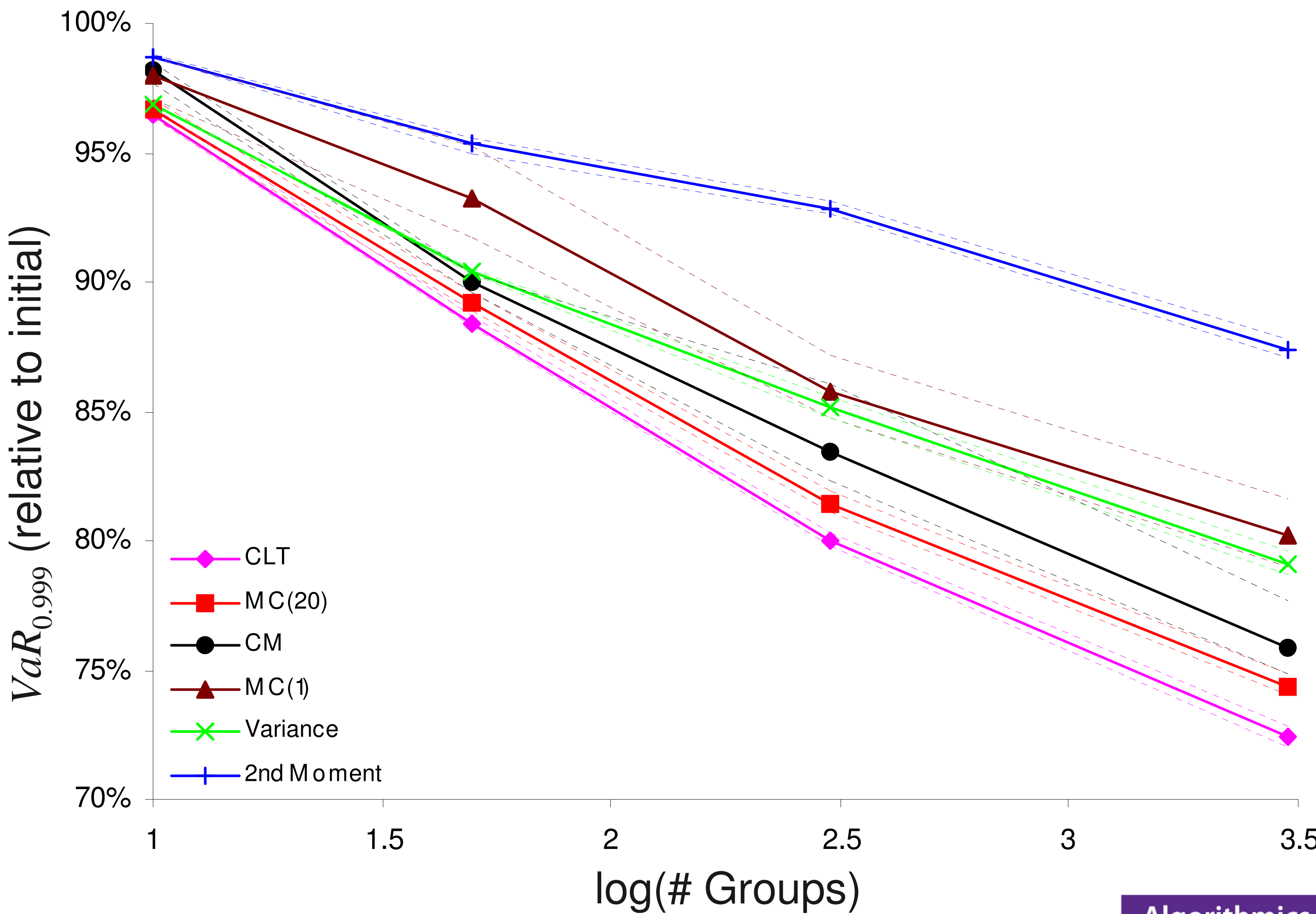
Idea: as the number of counterparties increases, the contribution of the variances to the sum becomes small relative to that of the means

Suppose $\mu_{L^j|\mathbf{y}_m} x_j \approx \mu, \quad \sigma_{L^j|\mathbf{y}_m} x_j \approx \sigma$

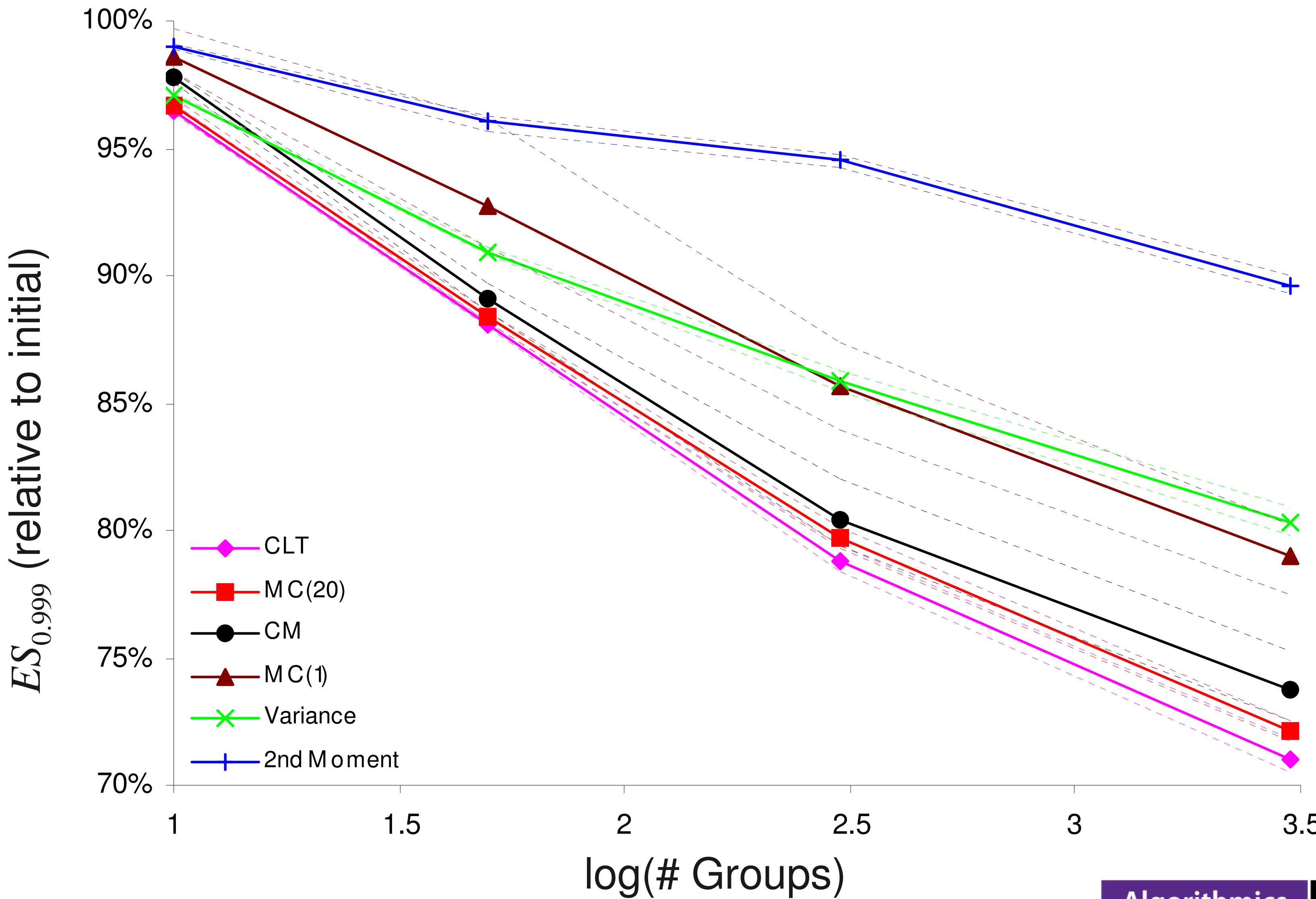
From CLT:

$$\mathrm{L}(\boldsymbol{x})\,|\,\boldsymbol{y}_m \approx J\mu+\sqrt{J}\sigma Z, \quad Z\sim \mathrm{N}\left(0,1\right)$$

Out-of-Sample *VaR*

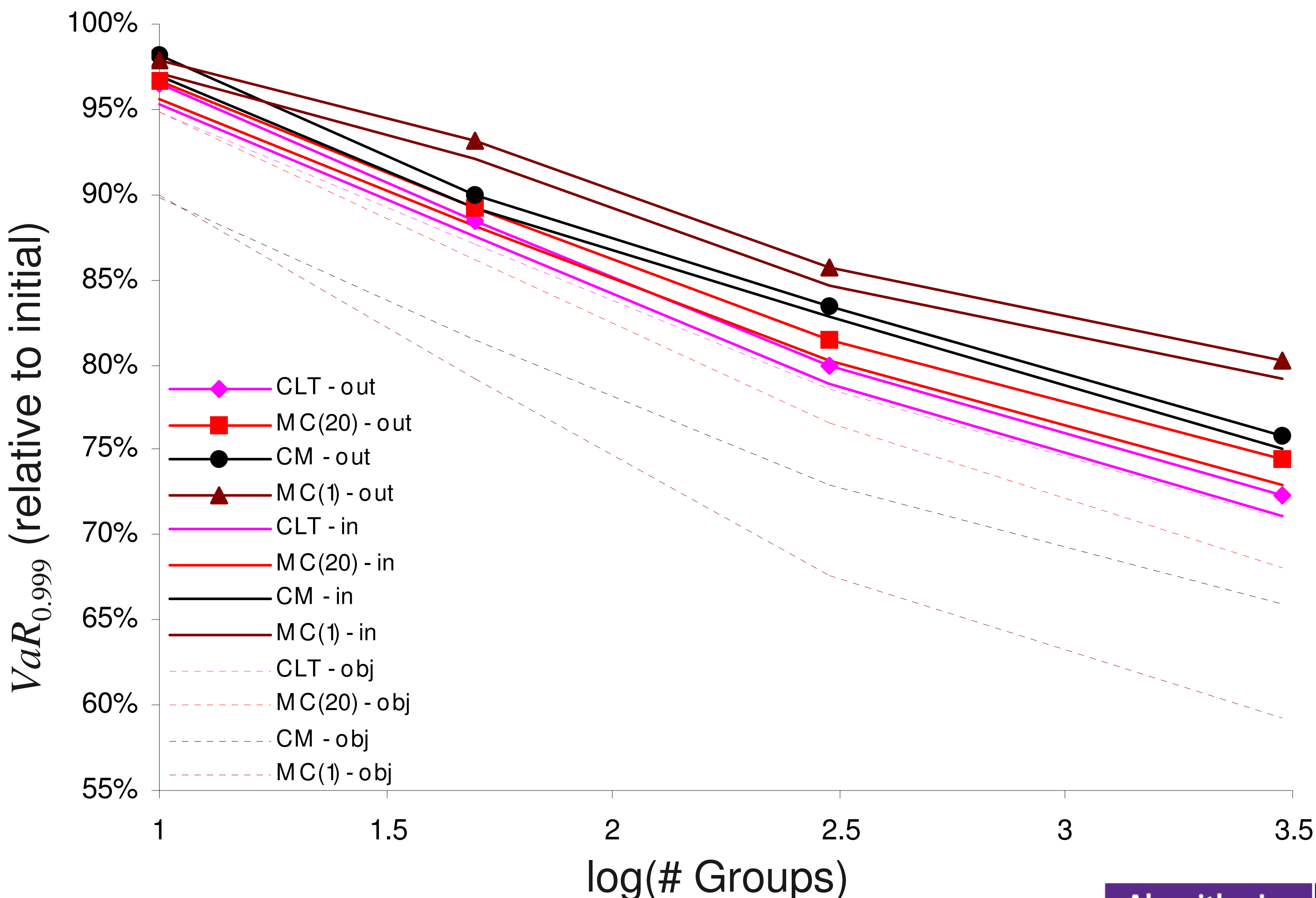


Out-of-Sample *ES*



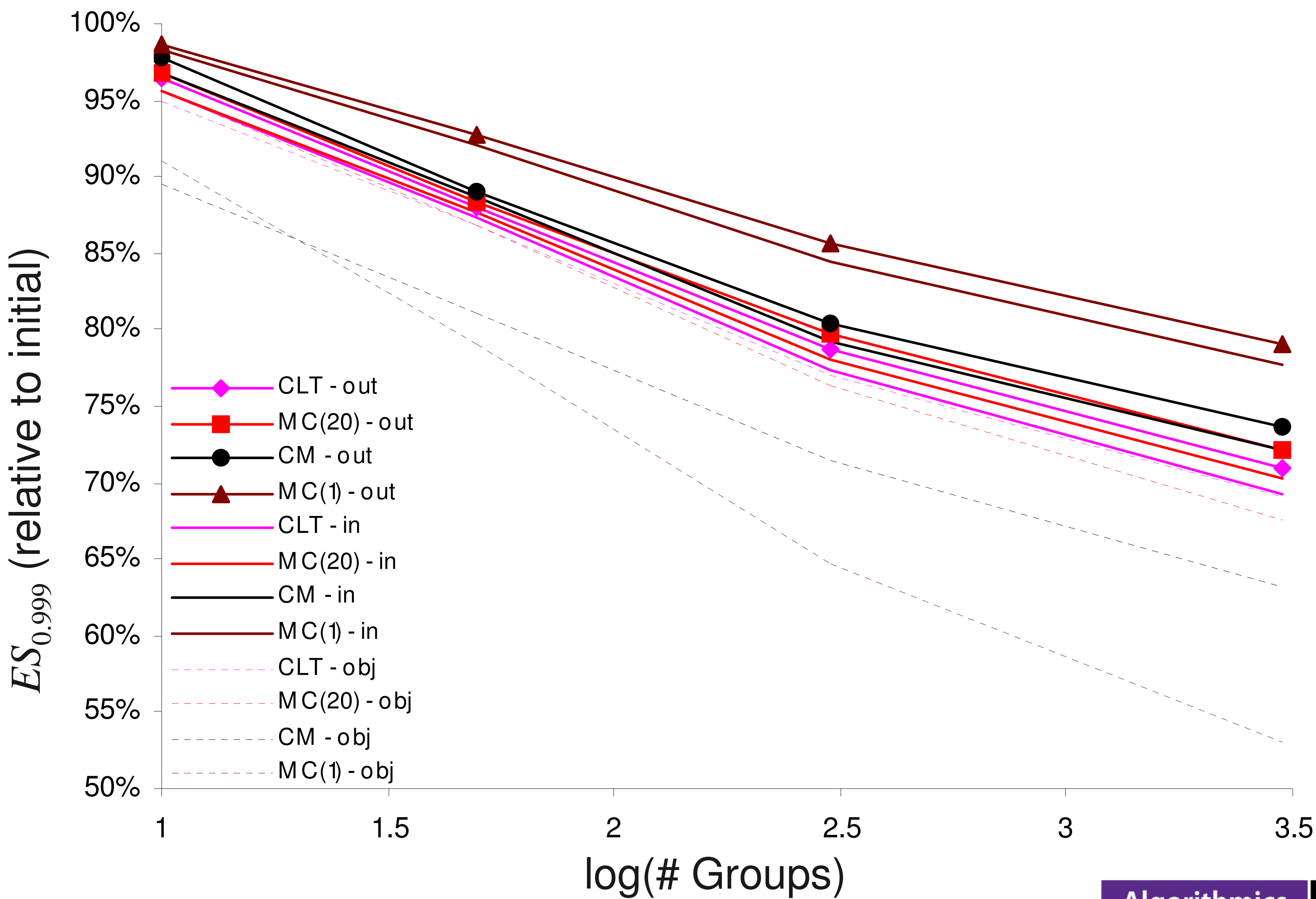
Algorithmics

Approximation Quality for *VaR*



Algorithmics

Approximation Quality for *ES*



Algorithmics

CLT Gradients and Hessians

Calculating gradients

$$\nabla \ell_\alpha(x) = f(\ell_\alpha(x))$$

$$\nabla \mathrm{ES}_\alpha(x) = f(\ell_\alpha(x), \mathrm{ES}_\alpha(x))$$

Calculating Hessians

$$\nabla^2 \ell_\alpha(x) = f(\ell_\alpha(x), \nabla \ell_\alpha(x))$$

$$\nabla^2 \mathrm{ES}_\alpha(x) = f(\ell_\alpha(x), \mathrm{ES}_\alpha(x), \nabla \mathrm{ES}_\alpha(x))$$

Non-linear optimization algorithm

$$x^{k+1} = x^k - (\nabla^2 f(x^k))^{-1} \nabla f(x^k)$$

Algorithmics

Other Test Results

Model	Risk Measure	Init portf	20 Ggps 1 CP Heterog Budget	20 Ggps 1 CP Heterog Budget 99% Qt	20 Ggps 1 CP Heterog Default	20 Ggps 60 CPs Heterog	20 Ggps 150 CPs Heterog	20 Ggps 150 CPs Homog	50 Ggps 10 CPs Heterog	50 Ggps 10 CPs Heterog Budget	500 Ggps 1 CP Heterog	500 Ggps 1 CP Heterog 50000sc	500 Ggps 1 CP Heterog Budget	500 Ggps 1 CP Heterog Default	500 Ggps 6 CPs Heterog
CLT	**ES 99.9%**	100%	59.93%	83.39%	61.95%	**88.32%**	**92.93%**	**86.43%**	**87.61%**	70.49%	**67.23%**	67.05%	**53.98%**	**34.66%**	**76.97%**
	VaR 99.9%	100%	125.04%	79.75%	45.38%	**87.38%**	**92.98%**	**87.38%**	**86.33%**	**70.09%**	**68.61%**	**68.60%**	**56.48%**	**35.61%**	**77.82%**
LLN	**ES 99.9%**	100%	95.58%	91.64%	77.34%	89.59%	93.69%	88.48%	113.19%	115.14%	87.09%	87.04%	114.96%	45.07%	78.66%
	VaR 99.9%	100%	144.06%	46.37%	64.72%	89.53%	94.74%	91.26%	115.91%	111.74%	87.09%	87.13%	118.93%	44.28%	81.00%
MCs	**ES 99.9%**	100%	63.23%	72.17%	49.73%	91.26%	96.82%	89.29%	91.25%	78.09%	72.44%	68.42%	65.66%	40.61%	83.44%
	VaR 99.9%	100%	**89.04%**	47.75%	44.35%	91.02%	96.30%	90.83%	90.06%	74.81%	73.14%	70.73%	65.44%	40.93%	84.09%
MCs (x5)	**ES 99.9%**	100%	**47.23%**	**69.62%**	**44.55%**	89.10%	94.14%	86.95%	88.52%	**70.22%**	68.75%	**67.04%**	57.74%	35.93%	79.08%
	VaR 99.9%	100%	102.22%	49.36%	**41.28%**	88.72%	94.22%	87.97%	87.58%	71.22%	70.65%	69.11%	60.05%	36.41%	81.39%
WMCs	**ES 99.9%**	100%	63.93%	75.07%	50.30%	93.31%	98.59%	91.10%	92.61%	78.79%	72.45%	69.08%	65.66%	40.69%	83.79%
	VaR 99.9%	100%	90.68%	53.78%	45.93%	91.15%	97.50%	91.02%	91.37%	77.07%	73.06%	70.89%	65.66%	41.30%	85.17%
MV (CLT)	**ES 99.9%**	100%	93.03%	87.60%	76.43%	91.15%	96.88%	87.59%	115.38%	138.29%	91.90%	92.31%	136.88%	40.77%	83.24%
	VaR 99.9%	100%	129.89%	**45.23%**	65.64%	89.80%	96.12%	88.27%	111.55%	143.04%	91.91%	92.29%	141.99%	40.96%	83.05%
MV (MCs)	**ES 99.9%**	100%	73.88%	78.80%	56.56%	90.87%	95.60%	87.87%	92.16%	80.20%	78.69%	77.10%	64.71%	38.29%	83.68%
	VaR 99.9%	100%	117.20%	72.99%	44.72%	89.17%	95.03%	88.41%	90.15%	79.07%	77.94%	76.43%	64.18%	38.23%	83.15%

Algorithmics

Performance (50,000 Systemic Samples)

		$VaR_{0.999}$			
Model	Solver	10 grp	50 grp	300 grp	3000 grp
CLT	IPOPT	24 - 30	30 - 35	72 - 443	
CM	CPLEX	22 - 24	66 - 80	500 - 748	
MC(1)	CPLEX	34 - 59	107 - 188	646 - 780	
MC(20)*	CPLEX	3579 - 3715	2393 - 2945	6820 - 8990	

		$ES_{0.999}$			
Model	Solver	10 grp	50 grp	300 grp	3000 grp
CLT	IPOPT	22 - 29	29 - 53	73 - 161	
CM	CPLEX	4 - 10	11 - 14	57 - 76	
MC(1)	CPLEX	9 - 13	20 - 28	58 - 66	
MC(20)	CPLEX	138 - 179	270 - 315	437 - 582	

		Variance			
Model	Solver	10 grp	50 grp	300 grp	3000 grp
Uncond	MOSEK	< 1	< 1	1	

Elapsed time (sec)

Server : 8 x Opteron 885 CPU, 16 cores (jobs run on 1 core), 64 Gb RAM

* VaR optimization for MC(20) was run in parallel mode on 4 threads

Algorithmics

Detailed Performance Data

Credit-Risk Model with Credit-State Migrations

3000 Groups, Wide Budget, 10000 Scenarios, 99.9% Quantile

Problem dimension: 3000 groups - 6000 variables, 6003 constraints

Minimizing Value-at-Risk or Expected Shortfall

The Hessian Matrix is Computed or Approximated

Solver / Model	Solution status	Solution time (seconds)	Relative difference in optimal solution	Number of iterations	Number of function evaluations	Number of gradient evaluations	Number of Hessian evaluations
MOSEK Objective: VaR Hessian: computed	The Optimization Problem is Nonconvex	2185		-	-	-	-
IPOPT Objective: VaR Hessian: computed	Optimal Solution Found (Overall /max solution error: 8.0e-09)	11484		64	65	65	64
IPOPT Objective: VaR Hessian: approximation	Solved To Acceptable Level (Overall /max solution error: 9.1e-07)	1408		438	1197	441	0
MOSEK Objective: Expected Shortfall Hessian: computed	Optimal (Overall /max solution error: 1.6e-08)	8058	-0.00037% (vs. IPOPT Hes)	36	39	75	37
MOSEK Objective: Expected Shortfall Hessian: computed *Parallel – 8 CPUs*	Optimal (Overall /max solution error: 1.6e-08)	1672		36	39	75	37
IPOPT Objective: Expected Shortfall Hessian: computed	Optimal Solution Found (Overall /max solution error: 2.5e-09)	11554	0.00037% (vs. MOSEK Hes)	65	66	66	65
IPOPT Objective: Expected Shortfall Hessian: approximation	Optimal Solution Found (Overall /max solution error: 1.5e-09)	979	0.00076% (vs. MOSEK Hes)	260	465	261	0

Algorithmics

Industry Practice (March 2009)

Typical portfolio size: 5,000 counterparties

Typical no. credit drivers per counterparty: 1

Typical beta: 0.4 – 0.5

Typical no. systemic samples: 10,000

Typical no. specific samples: 1,000 (for risk measurement, not optimization)