Threshold Accepting for Credit Risk Assessment and Validation

M. Lyra[1] A. Onwunta P. Winker

COMPSTAT 2010

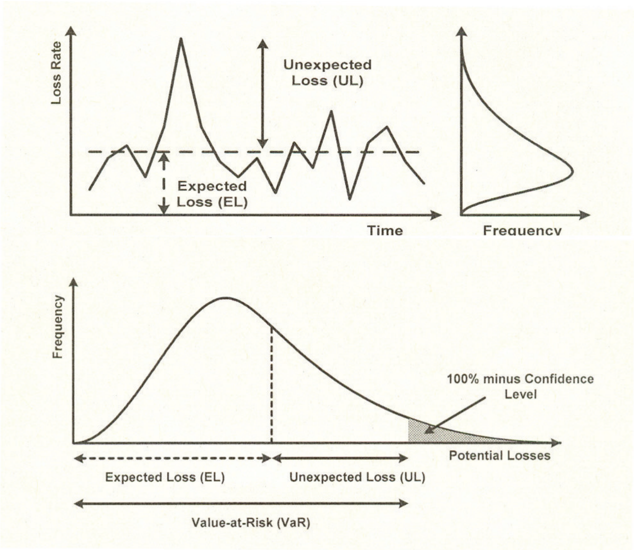
August 24, 2010

[1] Financial support from the EU Commission through COMISEF is gratefully acknowledged

1. **Introduction**
   - Basel II and credit risk clustering
   - Optimal size and number of clusters
2. **Ex-post validation**
   - Actual number of defaults
3. **Optimal buckets**
4. **Conclusion**
   - Summary - Outlook
   - For further reading

1. **Introduction**
   - Basel II and credit risk clustering
   - Optimal size and number of clusters
2. **Ex-post validation**
   - Actual number of defaults
3. **Optimal buckets**
4. **Conclusion**
   - Summary - Outlook
   - For further reading

Basel II and credit risk clustering



Regulatory Capital

Accurate regulatory capital calculation.

Credit Risk Bucketing

- Step 1: Compute borrowers' probability of default (p_k)

Regulatory Capital

Accurate regulatory capital calculation.

Credit Risk Bucketing

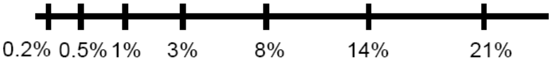
- Step 1: Compute borrowers' probability of default (p_k)

Regulatory Capital

Accurate regulatory capital calculation.

Credit Risk Bucketing

- Step 2: Assign borrowers to groups (grades)

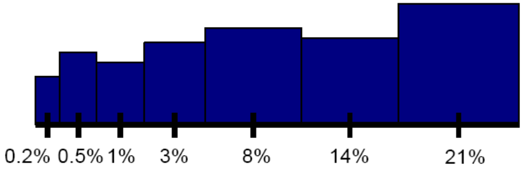


Regulatory Capital

Accurate regulatory capital calculation.

Credit Risk Bucketing

- Step 3: Compute MCR for each grade (based on its $\overline{p}_g$)



Regulatory Capital

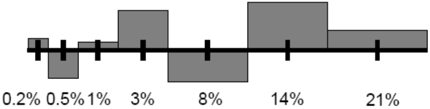
Accurate regulatory capital calculation.

Credit Risk Bucketing

- Step 1: Compute borrowers' probability of default (p_k)
- Step 2: Assign borrowers to groups (grades)
- Step 3: Compute MCR for each grade (based on its $\overline{p}_g$)
- Approximation Error

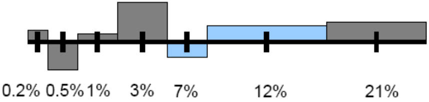
Approximation Error

Using $\overline{p}_g$ instead of individual $p_k$causes a loss in precision.



Meaningful assignment of borrowers to clusters

Choose appropriate size and number of clusters to minimize over/understatement of MCR and allow statistical ex-post validation



Optimal size and number of clusters

Optimal Credit Risk Rating System

Choose appropriate size and number of grades

(ex post)

- *Predicts defaults correctly*

Optimal size and number of clusters

Optimal Credit Risk Rating System

Choose appropriate size and number of grades

(ex post)

- *Predicts defaults correctly*

Optimal Credit Risk Rating System

Choose appropriate size and number of grades

(ex post)

- *Predicts defaults correctly*

1. Introduction
   - Basel II and credit risk clustering
   - Optimal size and number of clusters
2. **Ex-post validation**
   - Actual number of defaults
3. Optimal buckets
4. Conclusion
   - Summary - Outlook
   - For further reading

Validate Actual Number of Defaults

Predicted correctly if $D^a_g \in [D^f_{g,l}; D^f_{g,u}]$ with confidence 1-α

- $D^f_{g,l} = n_g \cdot max(\overline{p}_g - \varepsilon, 0)$
- $D^f_{g,u} = n_g \cdot min(\overline{p}_g + \varepsilon, 1)$

Validate Actual Number of Defaults

Predicted correctly if $D_g^a \in [D_{g,l}^f; D_{g,u}^f]$ with confidence 1-α

- $D_{g,l}^f = n_g \cdot max(\bar{p}_g - \varepsilon, 0)$
- $D_{g,u}^f = n_g \cdot min(\bar{p}_g + \varepsilon, 1)$

Model actual defaults as binary variable

$$\mathbb{P}_{int} = \mathbb{P}\left(D_{g,l}^f \leq D_g^a \leq D_{g,u}^f\right)$$

Validate Actual Number of Defaults

Predicted correctly if $D_g^a \in [D_{g,l}^f; D_{g,u}^f]$ with confidence 1-α

- $D_{g,l}^f = n_g \cdot max(\overline{p}_g - \varepsilon, 0)$
- $D_{g,u}^f = n_g \cdot min(\overline{p}_g + \varepsilon, 1)$

Binomial distribution

$$\mathbb{P}_{int} = \sum_{k=D_{g,l}^f}^{D_{g,u}^f} \binom{n_g}{k} \overline{p}_g^k \left(1 - \overline{p}_g\right)^{n_g - k} \geq 1 - \alpha\,.$$

1. Introduction
   - Basel II and credit risk clustering
   - Optimal size and number of clusters
2. Ex-post validation
   - Actual number of defaults
3. **Optimal buckets**
4. Conclusion
   - Summary - Outlook
   - For further reading

Objective functions

- Objective function for minimizing within grades variance

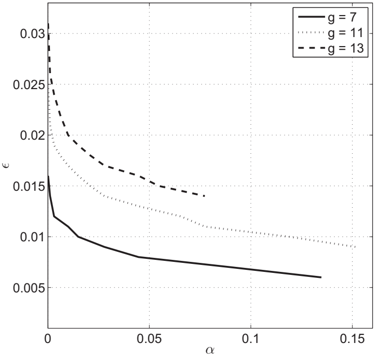
$$min \sum_{g} \sum_{k \in g} \left(\overline{p}_{c,g} - p_{c,k} \right)^2 \tag{1}$$

- Objective function for minimizing regulatory capital

$$min \sum_{g} \sum_{k \in g} 1.06 \cdot \left| UL\left(\overline{p}_g\right) - UL\left(p_k\right) \right| \tag{2}$$

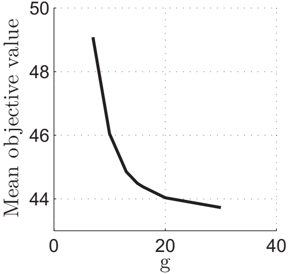
Feasible region

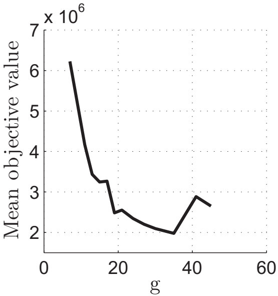
Minimizing regulatory capital using the validation technique ($\alpha = 1.5\%$, $\varepsilon = 1\%$)



Optimum backet setting

Within grades variace (left), Regulatory capital (right)





1. Introduction
   - Basel II and credit risk clustering
   - Optimal size and number of clusters
2. Ex-post validation
   - Actual number of defaults
3. Optimal buckets
4. **Conclusion**
   - Summary - Outlook
   - For further reading

Summary

- Minimum capital requirements to cover unexpected losses
- Threshold Accepting to cluster loans with real-world constraints
- Optimal size and number of buckets based on ex-post validation

Outlook

- Relax default risk independence constraint
- Alternative assumptions for actual default distributions

For further reading

P. Winker.
Onptimization Heuristics in Econometrics: Applications of Threshold Accepting.
Wiley, New York, 2001.

Basel Committee on Banking Supervision.
Capital Standards a Revised Framework.
Bank for International Settlements, 2006.

M. Lyra and J. Paha and S. Paterlini and P. Winker.
Optimization Heuristics for Determining Internal Rating Grading Scales.
Computational Statistics & Data Analysis, Article in Press.

M. Kalkbrener and A. Onwunta.
Validation Structural Credit Portfolio Models.
In:Model Risk in Finance, forthcoming.

P. Winker.
Onptimization Heuristics in Econometrics: Applications of Threshold Accepting.
Wiley, New York, 2001.

Basel Committee on Banking Supervision.
Capital Standards a Revised Framework.
Bank for International Settlements, 2006.

M. Lyra and J. Paha and S. Paterlini and P. Winker.
Optimization Heuristics for Determining Internal Rating Grading Scales.
Computational Statistics & Data Analysis, Article in Press.

M. Kalkbrener and A. Onwunta.
Validation Structural Credit Portfolio Models.
In:Model Risk in Finance, forthcoming.

For further reading

P. Winker.
Onptimization Heuristics in Econometrics: Applications of Threshold Accepting.
Wiley, New York, 2001.

Basel Committee on Banking Supervision.
Capital Standards a Revised Framework.
Bank for International Settlements, 2006.

M. Lyra and J. Paha and S. Paterlini and P. Winker.
Optimization Heuristics for Determining Internal Rating Grading Scales.
Computational Statistics & Data Analysis, Article in Press.

M. Kalkbrener and A. Onwunta.
Validation Structural Credit Portfolio Models.
In:Model Risk in Finance, forthcoming.

P. Winker.
Onptimization Heuristics in Econometrics: Applications of Threshold Accepting.
Wiley, New York, 2001.

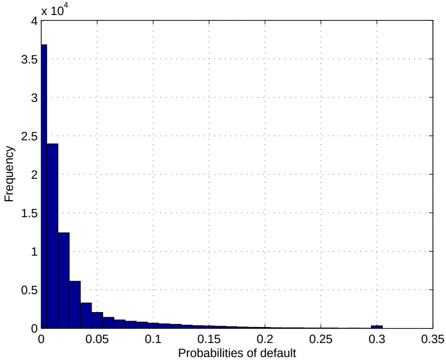
Basel Committee on Banking Supervision.
Capital Standards a Revised Framework.
Bank for International Settlements, 2006.

M. Lyra and J. Paha and S. Paterlini and P. Winker.
Optimization Heuristics for Determining Internal Rating Grading Scales.
Computational Statistics & Data Analysis, Article in Press.

M. Kalkbrener and A. Onwunta.
Validation Structural Credit Portfolio Models.
In:Model Risk in Finance, forthcoming.

Data description

- portfolio of 93 580 retail borrowers.
- LGDs range between 0.17 and 1.
- p_k vary from 0.000001% to 30%.



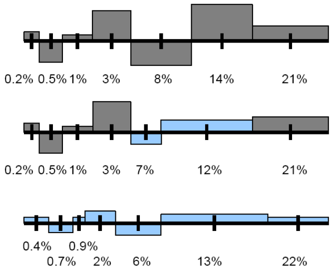
Credit Risk Assignment - Side Constraints

- Enforced by constraint handling techniques
  - $\overline{p}_g$ in bucket $\nleq$ 0.03%
  - Each bucket $\ngeq$ 35% of total bank exposure
- Considered in the structure of the algorithm
  - No bucket overlapping
  - Buckets correspond to all borrowers

Optimization Heuristics

Optimal partition of k bank clients in g clusters

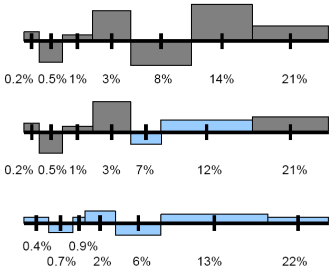
1. Generate random starting thresholds (candidate solution)
2. Alter current candidate solution
3. Accept or reject new candidate solution
4. Repeat until a very good solution is found



Optimization Heuristics

Optimal partition of k bank clients in g clusters

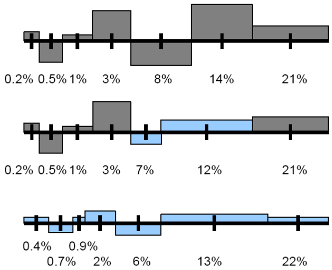
1. Generate random starting thresholds (candidate solution)
2. Alter current candidate solution
3. Accept or reject new candidate solution
4. Repeat until a very good solution is found



Optimization Heuristics

Optimal partition of k bank clients in g clusters

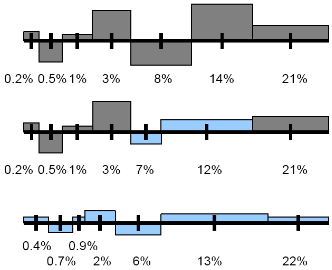
1. Generate random starting thresholds (candidate solution)
2. Alter current candidate solution
3. Accept or reject new candidate solution
4. Repeat until a very good solution is found



Optimization Heuristics

Optimal partition of k bank clients in g clusters

1. Generate random starting thresholds (candidate solution)
2. Alter current candidate solution
3. Accept or reject new candidate solution
4. Repeat until a very good solution is found



Threshold Accepting - The Basic Idea

- Generate a random candidate solution and determine its objective function value
- Repeat a predefined number of iterations
  - Modify candidate solution and determine its objective function value
  - Replace current solution with modified solution if new solutions yields
    - An improved objective function value or
    - A deterioration that is smaller than some threshold (predefined by a threshold sequence)

Algorithm 1 Threshold Accepting Algorithm.

Initialize n_R, n_{S_τ} , and $\tau_r, r = 1, 2, \ldots, n_R$
Generate at random a solution $x^0 \in [\alpha_l \alpha_u] \times [\beta_l \beta_u]$
for $r = 1$ to n_R **do**
  for $i = 1$ to n_{S_τ} **do**
    Generate neighbor at random, $x^1 \in \mathcal{N}(x^0)$
    if $f(x^1) - f(x^0) < \tau_r$ **then**
      $x^0 = x^1$
    end if
  end for
end for

Threshold Accepting - Candidate Solutions

- Starting Candidate Solution
  - For g buckets, select g-1 upper bucket thresholds from actual pds
  - Discrete search $\Rightarrow$ Each solution constitutes a new partition
- New Candidate Solution
  - Determine some bucket threshold of current solution randomly
  - Replace with new pd from interval [next lower threshold; next higher threshold]
  - Shrink interval linearly in the number of iterations; $[(I+1)-i]/I$

Threshold Accepting - Updating Objective Function Values

- Alter only one bucket threshold per iteration
- New objective function differs from that of the current solution only in contribution of two buckets
- Only compute those two buckets' fitness and update objective function value of current solution
- Consequence: Tremendous increase in search speed

Threshold Accepting - Threshold Sequence

- Idea: Use mean of last 100 weighted fitness differences (in absolute values) as threshold T
- If last fitness differences were mainly
  - improvements, T shrinks $\Rightarrow$ Stay on path to (local) optimum
  - deteriorations, T increases $\Rightarrow$ Overcome (local) optimum and search for a new one
- Weights (w_1, w_2) for restrictive threshold sequence
  - Fitness improvement (frequent and high at the beginning of the search) $\Rightarrow w_1 = i/I$
  - Fitness deterioration (frequent and high at the end of the search) $\Rightarrow w_2 = 1 - i/I$
- Scale above means with (1-i/I) for further restrictiveness

Algorithm 2 Pseudocode for TA with data driven generation of threshold sequence.

Initialize I, $Ls = (0, \ldots, 0)$ of length 100
Generate at random an initial solution x^c, set $\tau = f(x^c)$
for $i = 1$ to I **do**
    Generate at random $x^n \in \mathcal{N}(x^c)$
    Delete first element of Ls
    if $f(x^n) - f(x^c) < 0$ **then**
        add $|f(x^n) - f(x^c)| \cdot (i/I)$ as last element to Ls
    else
        add $|f(x^n) - f(x^c)| \cdot (1 - i/I)$ as last element to Ls
    end if
    $\tau = \overline{Ls} \cdot (1 - i/I)$
    if $f(x^n) - f(x^c) < \tau$ **then**
        $x^c = x^n$
    end if
end for

Constraint Handling - Rejection Technique in TA

- Both candidate solutions are feasible
  - TA: Select the new candidate if $f(g_n) + T \leq f(g_c)$
- One solution is feasible, select the feasible
- No feasible solution
  - Select fewer violations
  - Select with regard to fitness
    - TA: Select the new candidate if $f(g_n) + T \leq f(g_c)$

Constraint Handling - Penalty Technique in TA

- Penalize candidate solutions' objective value by a factor $A \in [1; 3.7183] \Rightarrow f_c(g) = f_u(g) \cdot A$
- A rises in the number of iterations i and the degree of constraint violation $a \in [0; 1] \Rightarrow A = \left(1 + exp(\frac{i}{I})\right)^a$
- $a = 1$, if
  - all buckets besides one are empty, and
  - EAD is concentrated in one bucket.
- Select the new candidate if $f_c(g_n) + T \leq f_c(g_c)$

Table: Objective function for minimizing within grades variance(1)

	Best	Mean	Worst	s.d.	q90%	Freq
			g = 7			
TA[a]	18.6836	18.6836	18.6836	$3.6731 \cdot 10^{-8}$	18.6836	8/10
TA[b]	18.6552	24.4809	46.2984	8.2478	24.8221	1/10
			g = 10			
TA[a]	9.7293	9.7293	9.7293	$5.3490 \cdot 10^{-7}$	9.7293	1/10
TA[b]	9.1118	10.3545	10.9233	0.8520	10.9108	1/10
			g = 13			
TA[a]	6.6716	6.6716	6.6716	$2.9353 \cdot 10^{-6}$	6.6716	1/10
TA[b]	6.5974	10.0515	14.5469	2.7151	12.4890	1/6
			$g = 16$			
TA[a]	5.2454	5.2454	5.2454	$1.9032 \cdot 10^{-6}$	5.2454	1/10
TA[b]	10.3647	10.3647	10.3647	0.0000	10.3647	1/1

[a] Actual number of defaults constraint
[b] Unexpected loss constraint

Table: Objective function for minimizing unexpected losses (2)

	Best	Mean	Worst	s.d.	q90%	Freq
			$g = 7$			
TA[a]	6,228,874	6,228,874	6,228,874	$9.8170 \cdot 10^{-10}$	6,228,874	10/10
TA[b]	6,419,727	6,423,788	6,426,403	2,053	6,420,826	1/10
			$g = 11$			
TA[a]	4,165,257	4,167,952	4,182,902	5,999	4,165,257	7/10
TA[b]	5,534,072	5,636,388	5,814,094	101,283	5,538,839	1/10
			$g = 13$			
TA[a]	3,425,092	3,435,627	3,436,798	3,701.71	3,436,798	1/10
TA[b]	5,192,945	5,608,280	5,929,156	230,630	5,846,709	1/9
			$g = 15$			
TA[a]	3,245,441	3,245,636	3,247,260	571.05	3,245,445	1/10
TA[b]	5,627,306	6,285,472	7,166,148	647,632	6,945,510	1/3

[a] Actual number of defaults constraint

[b] Unexpected loss constraint