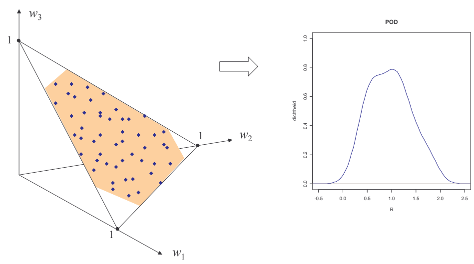
Portfolio Opportunity Distributions



Arno Weber

8 mei 2007

Stageopdracht

Portfolio Opportunity Distributions (PODs):

- Techniek voor Performance Evaluatie.
- Nog niet zo bekend.
- Wijkt af van traditionele methodes:
  - Benchmarks;
  - Peer groups.
- Gebruikt computersimulatie.
- R. Surz e.a. (1996).

Inhoud

- Knelpunten van benchmarks en peer groups
- PODs: globale werking
- Technische gedeelte: 3 algoritmes
- Voorbeelden
- Conclusies

Benchmark

Index die een onderliggende markt representeert.

- Marktindex.
  (AEX index, MSCI World, S&P 500, …)
- Customized index.

Excess return / Value added

$R_1^P, \ldots, R_n^P$: portefeuille rendementen

$R_1^M, \ldots, R_n^M$: benchmark rendementen

$$VA_i := R_i^P - R_i^M \quad \text{(aritmetisch)}$$

$$:= \frac{1+R_i^P}{1+R_i^M} - 1 \quad \text{(geometrisch)}$$

Is performance goed?

Statistische toets: $H_0: \quad \mathbb{E}VA \leq 0$

$H_1: \quad \mathbb{E}VA > 0$

Verwerp voor grote waarden van

$$T_n := \sqrt{n}\,\frac{\overline{VA}_n}{\sigma_{VA,n}}$$

- VA_i normaal verdeeld: t-toets.
- Zo niet: benader met CLS, of: tekentoets.

Problemen met benchmarks

- Statistische significantie moet over de tijd worden verkregen:

  1 waarneming per maand $\Rightarrow$
  meer dan 8 jaar wachten op 100 waarnemingen!

- Specifieke mandaatregels zijn niet weerspiegeld in benchmark:

Problemen met benchmarks

Mandaat A:	**Mandaat B**:
Aandelen Europa	Aandelen Europa
Tracking Error $\leq$ 10%	Tracking Error $\leq$ 15%
Max. exposure 15%	Max. exposure 20%

Benchmark: MSCI Europa

Beide managers dezelfde portefuille $\Rightarrow$
beide dezelfde performance.
Dit is niet terecht !

Peer group

Groep managers met een soortgelijk mandaat over dezelfde periode.

R^P : rendement portefeuille

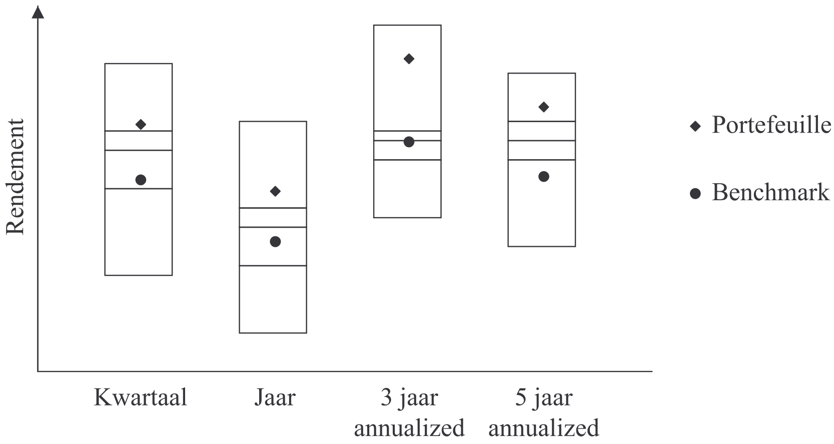
$R^{P_1},\ldots,R^{P_m}$: rendementen van peers

Idee: vergelijk R^P met $R^{P_1},\ldots,R^{P_m}$.

Is performance goed?

$$\textbf{Kwartiel-ranking} := \begin{cases} 1 & R^P \text{ ligt in top 25\%} \\ 2 & R^P \text{ ligt in top 50\%} \\ 3 & R^P \text{ ligt in top 75\%} \\ 4 & R^P \text{ anders} \end{cases}$$

Floating bar chart



Problemen bij peer groups

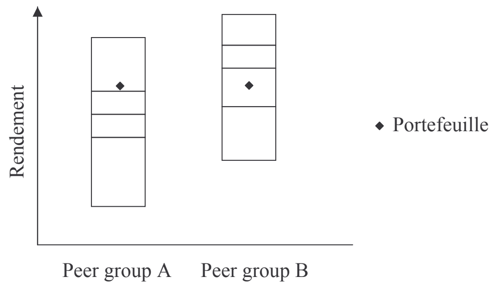
- **Survivorship bias**:
  Portefeuilles worden opgeheven door onderperformance en verdwijnen uit de database van de peer group provider.

  Langere meetperiode $\Rightarrow$ data onzuiverder.

  Marathon analogie (R. Surz, 1996):
  1000 atleten,
  100 halen de finish,
  is de 100-ste de laatste? Of in de top 10%?

Problemen bij peer groups

- **Classification bias**:
  Wanneer zijn mandaten vergelijkbaar?
- **Composition bias**:
  Database levert te weinig waarnemingen:



Beide “biassen” werken elkaar tegen.

Inhoud

- Knelpunten van benchmarks en peer groups
- PODs: globale werking
- Technische gedeelte: 3 algoritmes
- Voorbeelden
- Conclusies

Portfolio Opportunity Distributions

- Een alternatief voor benchmarks en peer groups:
- Elimineert alle zwakke punten.
- Idee: vergelijk het behaalde rendement met alle rendementen die behaald *hadden kunnen* worden.

Stap 1

Specificeer een **meetperiode** en het **mandaat** van de manager:

Voorbeelden van mandaatregels:

- Universum van beleggingsobjecten.
- Grenzen op exposure gewichten.
- Grenzen op aantal verschillende stukken.
- Grenzen op gewicht in een klasse (regio, land, sector,…)
- Grenzen op ex-ante risicomaten (tracking error).

Stap 2

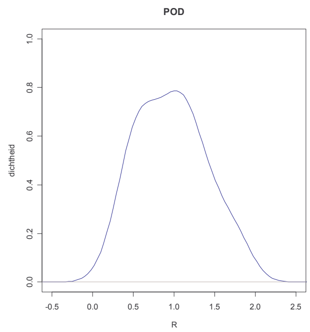
Genereer een groot aantal **random portefeuilles** die voldoen aan het mandaat.

Wall Street Journal, 1988:

Beleggers nemen het op tegen een fictieve aap (**beursgorilla**), die belegt volgens een random strategie.

Stap 3

Bereken de rendementen van de getrokken portefeuilles: $R_1, \ldots, R_n$
Hiermee wordt een kansverdeling geschat:



Monte Carlo Simulatie

Portfolio Opportunity Distrubution (POD)

Stap 4

Pas statistiek toe:

R^P : gerealiseerd rendement
$R_1, \ldots, R_n$: POD schatting
R : POD exact (stochast)

POD ranking $\theta := \mathbb{P}(R > R^P)$

"kans dat de aap de belegger verslaat"

POD ranking

Voer in: $Y_i := \begin{cases} 1, & R_i > R^P; \\ 0, & R_i \le R^P. \end{cases}$

Schatter: $\hat{\theta}_n := \overline{Y}_n = \frac{1}{n}\sum_{i=1}^{n} Y_i$

Betrouwbaarheidsinterval:

$$\theta = \overline{Y}_n \pm \sqrt{\frac{\overline{Y}_n(1-\overline{Y}_n)}{n-1}}\Phi^{-1}(1-\alpha/2)$$

Is de aap verslagen?

Statistische toets: $H_0: \quad \text{mediaan}(R) \geq R^P$

$H_1: \quad \text{mediaan}(R) < R^P$

- R_i normaal verdeeld: t-toets:

$$T_n := \sqrt{n}\,\frac{\overline{R}_n - R^P}{\sigma_{R,n}}$$

- Zo niet: tekentoets:

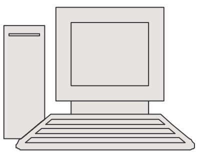
$$T_n := \#\{i : R_i < R^P\}$$

Verwerp voor grote waarden van T_n.

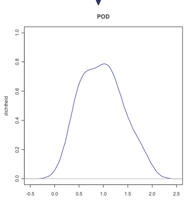
Samenvatting

Mandaat

Marktgegevens
(o.a. stukrendementen)



MCS



POD

R^P

Statistiek

Performance

Voordelen t.o.v. benchmarks

- Per periode een kansverdeling i.p.v. 1 getal.
- Alle regels van het mandaat wegen mee, niet alleen het universum.

Idee: POD gemiddelde vormt een custom benchmark. Benchmark analyses zijn dus toepasbaar op PODs.

Voordelen t.o.v. peer groups

- Géén classification bias.
- Géén composition bias.
- Géén survivorship bias.

POD universum vormt een “ideale” peer group, **maar:** bestaat uit random portefeuilles, niet uit echte portefeuilles.

Kritische kanttekeningen

- Vergelijking met “aap” en niet met echte concurrenten.
- Mandaat moet goed gedefinieerd zijn.
- Geen transacties (voorlopig niet).
- Mogelijk lange rekentijd door computersimulatie.

Inhoud

- Knelpunten van benchmarks en peer groups
- PODs: globale werking
- Technische gedeelte: 3 algoritmes
- Voorbeelden
- Conclusies

Technische gedeelte

Hoe kunnen we random portefeuilletrekking uitvoeren?

- Model
- 3 algoritmes

Model

$N :=$ # objecten in het universum

Allocatie: # segmenten
Selectie: # stukken

$W := (W^1, \ldots, W^N)$

vector met getrokken gewichten, dus

$W^j \in [0\%, 100\%]$ en $\sum_{j=1}^{N} W^j = 100\%$

Model

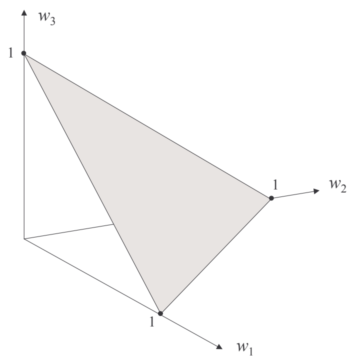
$R^j :=$ rendement van object j

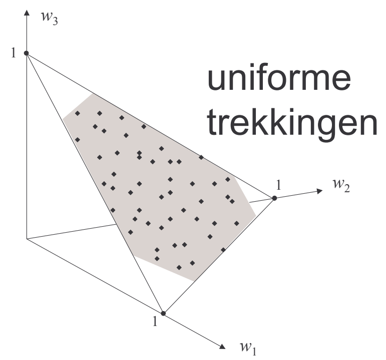
Allocatie: benchmark rendement
Selectie: stukrendement

$R(W) = \sum_{j=1}^{N} W^j R^j$ rendement van de random portefeuille

Model

Eenheidssimplex ($N = 3$)



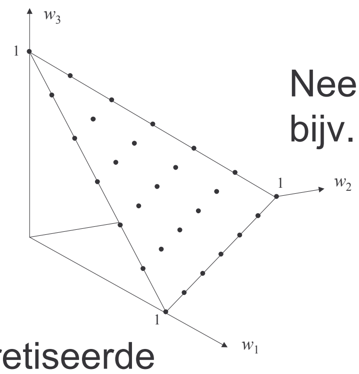


volledige simplex

toegelaten verzameling corr. met mandaat

Algoritme 1

Geen random trekkingen, maar benadering van de exacte POD:



Neem discretisatiestap d
bijv. $d = 0.01\%$

gediscretiseerde eenheidssimplex

Algoritme 1

1. Loop alle vectoren w af met gewichten een veelvoud van d.
2. **Acceptatie/rejectie**:
   bepaal of w voldoet aan alle mandaatregels.
   Ja: Bereken het rendement van w en voeg die toe aan de dataset.
   Nee: Ga naar de volgende w.
3. De empirische verdeling, bepaald door de rendementen, benadert de exacte POD.

Algoritme 1

Groot nadeel: Complexiteit!
Als $k := 100\% / d$ dan is het aantal langs te lopen vectoren

$$\binom{N+k-1}{k} = \frac{(N+k-1)!}{(N-1)!k!}$$

(exponentieel in N)

Daardoor alleen bruikbaar als $N \leq 5$.

Algoritme 2

1. Trek uniform een vector W uit de N-dimensionale eenheidssimplex.
2. **Acceptatie/rejectie**:
   bepaal of W voldoet aan alle mandaatregels.
   Ja: Bereken het rendement van en voeg die toe aan de dataset.
   Nee: Ga naar 1.
3. Dataset groot genoeg? Stop.
   Anders: Ga naar 1.

Algoritme 2

Procedure voor uniforme trekking W uit de N-simplex:

1. Doe N trekkingen $E^1,\ldots,E^N$ uit de Exponentieel(1) verdeling.
2. Bereken de som $$S := \sum_{j=1}^{N} E^j$$
3. $W := (W^1,\ldots,W^N)$ met $W^j := E^j / S$

(**Stelling**: W is uniform verdeeld op de N-simplex.)

Algoritme 2

Voordeel: rekentijd lineair in dimensie N.

Nadeel: niet bruikbaar als acceptatie/rejectie vrijwel alles weigert.
Preciezer: als kans op weigeren $\geq 99.999\%$ dan kost 1 trekking minuten rekentijd.

Voorbeeld: $N = 100$, gewichten max 2.6%.

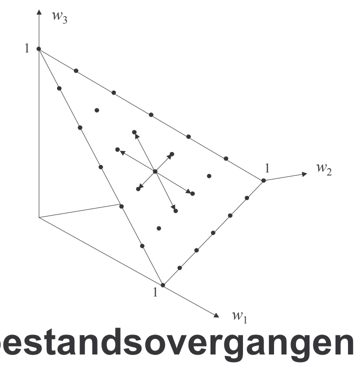
Algoritme 3

Markov-keten Monte Carlo (**MCMC**):

Idee: Simuleer een stochastisch proces (Markov-keten) op de toegelaten verzameling zodat de limietverdeling de uniforme verdeling is.

Algoritme 3

Toestandsruimte: toegelaten punten in de gediscretiseerde eenheidssimplex.



Neem discretisatiestap d
bijv. $d = 0.01\%$

Toestandsovergangen: alle mogelijke overgangen naar een naaste buur (1 gewicht $+d$, 1 gewicht $-d$), mits toegelaten.

Algoritme 3

Hoe garanderen dat de limietverdeling uniform is?

$(M_n)_n$: Naaste buren keten
Limietverdeling: ???

$(X_n)_n$: Afgeleide keten
Limietverdeling: uniform!

Algoritme 3

Hastings-Metropolis algoritme:

- Kies een start toestand $X_0 = M_0$
- Gegeven een toestand X_n:
  Genereer kandidaat-toestand K volgens de keten $(M_n)_n$: $\mathbb{P}(K = k) = m_{X_n,k}$

  Met kans $\min\left(\frac{m_{K,X_n}}{m_{X_n,K}}, 1\right)$ accepteer K: $X_{n+1} := K$

  Anders: $X_{n+1} := X_n$

(**Stelling**: limietverdeling van $(X_n)_n$ is uniform.)

Algoritme 3

Voordeel: trekking rechtstreeks uit toeg. gebied. Werkt dus ook voor zeer klein toeg. gebied.

Nadeel: rekentijd: # iteraties per trekking kan hoog oplopen.
Voorbeeld: $N \approx 100$, 16 uur voor 100 trekkingen.
Oplossing: indelen van stukken in groepen, maar dan kan de POD er afwijkend uit zien.

Conclusie: Algoritme 3 alleen gebruiken als er geen alternatief is.

Wanneer welk algoritme?

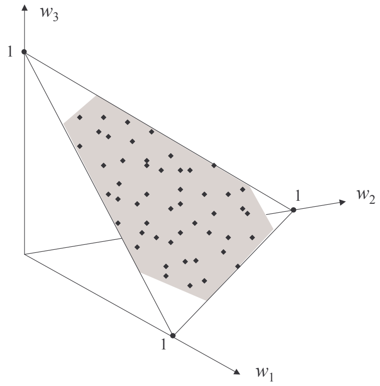
- $N \leq 5$: Alg 1 of 2.
- $N > 5$, groot toeg. gebied: Alg 2.
- $N > 5$, klein toeg. gebied: Alg 3.

Openstaande problemen

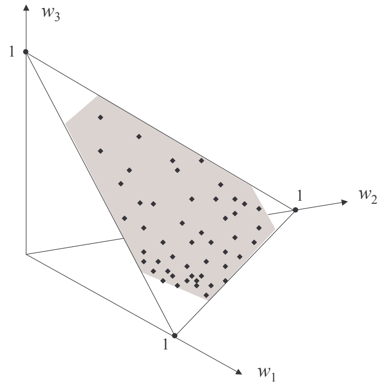
- Schatting ex-ante tracking error.
- Simuleren van transacties.
- Kansverdelingen.

Kansverdelingen

Welke verdeling op het toeg. gebied?



Uniforme verdeling



Marktverdeling (kapitalisatie gewogen)

Kansverdelingen

Definitieprobleem marktverdeling. We willen o.a.:

- Dichtheid groter naarmate de kapitalisatie

$$\sum_{j=1}^{N} cap^j w^j$$

groter is.
Voorbeeld: lineaire dichtheid

$$f_W(w) \sim \sum_{j=1}^{N} cap^j w^j$$

- Gemiddelde = benchmark:

$$\mathbb{E}W \sim (cap^1, \ldots, cap^N)$$

Inhoud

- Knelpunten van benchmarks en peer groups
- PODs: globale werking
- Technische gedeelte: 3 algoritmes
- Voorbeelden
- Conclusies

Voorbeelden

Illustratie van de POD techniek op 2 voorbeeld mandaten:

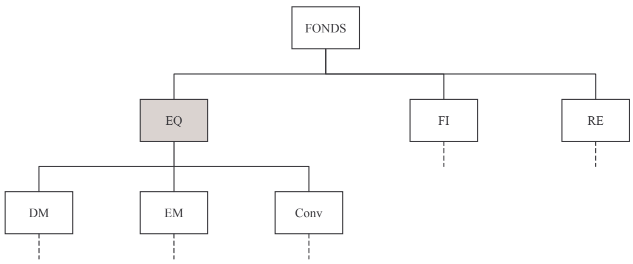
- Allocatie naar 3 equity segmenten.
- Selectie naar 314 aandelen Japan.

Voorbeeld 1

Equity segment van een VBI.

Allocatie naar 3 onderliggende segmenten:

- Developed Market (DM)
- Emerging Market (EM)
- Convertible bonds (Conv)



Voorbeeld 1

Mandaatregels:

- Ex-ante tracking error ten hoogste 5.5%.
- Exposure gewichten binnen de intervallen

  DM: [77.8%, 91.6%];

  EM: [5.5%, 13.9%];

  Conv: [2.8%, 8.4%].

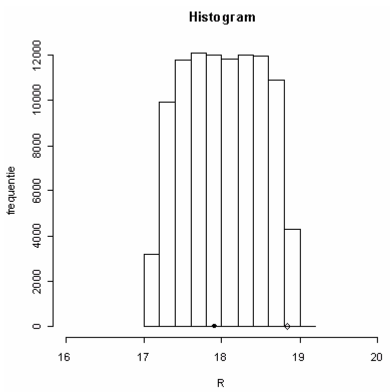
Meetperiode: 2005-2006

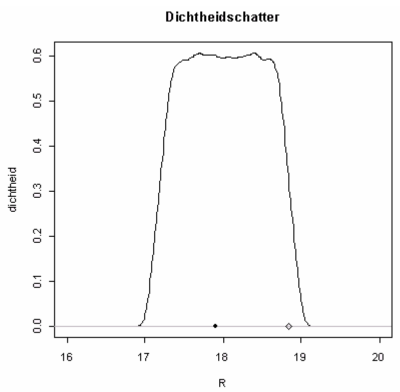
Gerealiseerde rendementen (ann.):

- PF: 18.84%
- BM: 17.90%
- DM: 16.40%
- EM: 35.11%
- Conv: 10.29%

Voorbeeld 1

Portfolio Opportunity Distribution





Voorbeeld 1

- POD ranking:

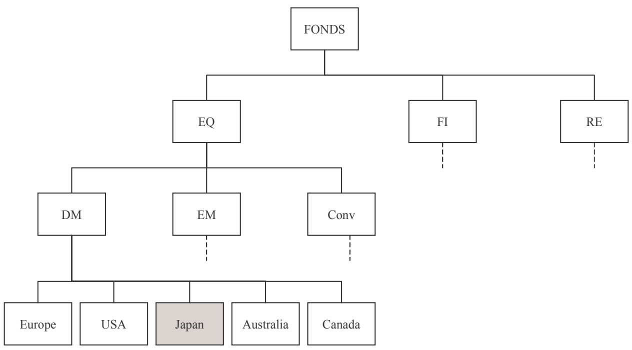
$\hat{\theta}_n = 0.0291$, met 95% b.t.i. $\theta \in [0.0281, 0.0302]$

- Tekentoets:

$p = 0$

Voorbeeld 2

Selectiemandaat Japanse aandelen.
Benadering: 314 MSCI Japan aandelen.



Voorbeeld 2

Mandaatregels:

- Ex-ante tracking error ten hoogste 1.7%.
- Exposure gewichten binnen het interval [0%, 5%].
- Aantal positieve gewichten tussen 50 en 80.

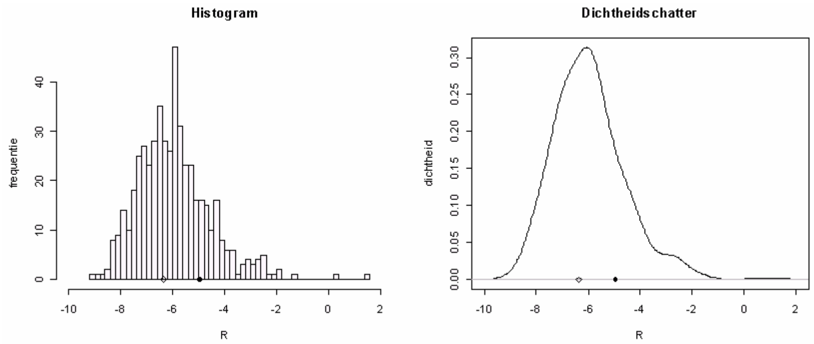
Meetperiode: 2006

Gerealiseerde rendementen:

- PF: -6.359%
- BM: -4.959%

Voorbeeld 2

Portfolio Opportunity Distribution



POD ranking: $\hat{\theta}_n = 0.584$, met 95% b.t.i. $\theta \in [0.541, 0.627]$

Inhoud

- Knelpunten van benchmarks en peer groups
- PODs: globale werking
- Technische gedeelte: 3 algoritmes
- Voorbeelden
- Conclusies

Conclusies

- PODs elimineren de knelpunten van benchmarks en peer groups.
  Reden: alle mandaatregels zijn deel van de input, en simulatie geeft in korte tijd veel data.
- Interpretatie: gekozen strategie wordt vergeleken met alle mogelijke “passieve” strategieën.
- Simuleren van transacties is soms noodzakelijk, maar dit leidt tot een keuzeprobleem.
  Voor een evt. nieuw ORTEC product zal aanvullend onderzoek nodig zijn.