

Financial Portfolio Optimization

Bernardo Gomes Kishan Rama Tomás Falcato

bernardo.n.gomes@ist.utl.pt kishan.rama@ist.utl.pt

tomas.falcato.costa@ist.utl.pt

Instituto Superior Técnico

Abstract—Para o problema de *Financial portfolio optimization*, dado um conjunto de *assets*, pretende-se obter uma forma de distribuir o dinheiro a investir por forma a maximizar o retorno. A cada *asset*, existe um risco associado, sendo que o risco de cada um poderá, ou não, influenciar o risco dos restantes. Se considerarmos todas as variáveis inerentes à otimização deste tipo de problemas, iremos obter funções não convexas, pelo que se considera uma aproximação facilmente computável através de otimização convexa e cujos resultados, obtidos a partir do *software CVX*, são bastante próximos dos ótimos. Posteriormente, o problema foi reformulado por forma a obter uma solução sub-ótima mas com menor tempo de computação.

Keywords. Portfolio Optimization •CVX •Barrier Method •Newton Method

I. INTRODUÇÃO

Este artigo pretende obter uma solução para o problema de *Portfolio Optimization*. Para tal, a nossa abordagem maximiza o retorno esperado e minimiza o risco de investimento tendo em conta a aversão ao risco colocada pela pessoa que investe.

Na vida real, o problema discutido neste artigo tem extrema importância na medida em que praticamente todas as pessoas fazem investimentos do seu dinheiro. A formulação terá em conta não só os casos em que a pessoa pretende rápidos retornos e pretende que estes sejam bastante elevados, independentemente de o risco a ele associado ser relativamente elevado, como também casos cujo objetivo seja obter um resultado de longo prazo em que a aversão ao risco é um fator preponderante face à quantia recebida.

Por forma a resolver, e melhor compreender o problema, numa primeira abordagem foi utilizado o *software CVX*. Posteriormente, implementa-se a mesma resolução do problema, mas com este reescrito por forma a utilizar o algoritmo *Barrier Method*.

O artigo está organizado da forma enunciada de seguida. No capítulo II, é introduzida a função a otimizar e as respetivas variáveis, capítulo III descreve a abordagem para a utilização do *software CVX*, sendo esta alterada para a utilização do *Barrier Method*, descrito no capítulo IV. Os resultados obtidos nas duas fases são discutidos e comparados em V. As conclusões são apresentadas posteriormente em VI.

II. FORMULAÇÃO DO PROBLEMA

Para a obtenção dos resultados do problema descrito anteriormente, a função utilizada foi a seguinte:

$$\begin{aligned} &\text{maximize} && \mu^\top \omega - \gamma \omega^\top \Sigma \omega \\ &\text{subject to} && 1^\top \omega = 1 \\ &&& \omega \in \mathbb{R}_+^n \end{aligned} \quad (1)$$

Apresenta-se de seguida a descrição das variáveis da função de otimização:

- $\omega \rightarrow$ vetor de dimensão n , onde cada posição i , contém a fração do dinheiro a investir no *asset* i para obter o máximo de lucro. Será esta a variável que se pretende maximizar;
- $\mu \rightarrow$ vetor de dimensão n , onde cada posição i , contém o retorno esperado para o *asset* i ;
- $\Sigma \rightarrow$ matriz das covariâncias, de dimensão $n \times n$ dos retornos dos *assets*. É através deste parâmetro que se representa a influência que cada *asset* tem nos restantes e vice-versa. Por exemplo, é de esperar que tendo um valor muito elevado na entrada ij desta matriz, no caso de haver uma quebra nos retornos do *asset* i , o *asset* j irá, também sofrer quebra nos retornos, ou um aumento dos mesmos, dependendo do valor dessa entrada. Um caso real desta formulação é a forte dependência entre as ações do petróleo e dos automóveis ou a relação de competição entre cadeias concorrentes;
- $\mu^\top \omega \rightarrow$ representa o retorno expectável do portfólio;
- $\gamma \rightarrow$ variável que controla o *tradeoff* entre o retorno e o risco. Será este parâmetro que irá controlar se se pretende investir dando preferência a maximizar o retorno, ou dando preferência à minimização do risco;
- $\gamma \omega^\top \Sigma \omega \rightarrow$ representa a variância do retorno do portfólio. É o parâmetro que mede o risco de investimento;

A *constraint* $1^\top \omega = 1$ indica que a soma dos investimentos no portfólio deve corresponder ao capital inicial que se pretende investir. Note-se que os investimentos são uma fração do capital inicial e não o seu valor absoluto. Ao forçar que $\omega \in \mathbb{R}_+^n$ indica que não podem existir investimentos negativos. Na prática esta situação poderia existir, no caso de termos capital num dado *asset* e a melhor solução ser retirar o capital deste para investir noutro mais lucrativo.

Ao analisar a função que queremos otimizar, verifica-se que é quadrática, ou seja, do tipo $x^\top A x + b^\top x + c$. Para o contexto em questão, tem-se que $\text{maximize } \mu^\top \omega - \gamma \omega^\top \Sigma \omega \equiv \text{minimize } \gamma \omega^\top \Sigma \omega + (-\mu^\top) \omega \equiv \text{minimize } \omega^\top \gamma \Sigma \omega + (-\mu^\top) \omega$.

Sendo a matriz das covariâncias semi-definida positiva e simétrica, verifica-se que a nossa função é realmente quadrática sendo $A = \gamma\Sigma$, $b = -\mu$ e $c = 0$. Sendo a função quadrática, podemos também concluir, que é convexa. Desta forma, a otimização será feita de acordo com esta propriedade.

III. CVX

Por forma a implementar a solução apresentada na equação 1, e sendo esta convexa, utilizou-se o *software* CVX[1]. De seguida apresenta-se a parte do código de *Matlab* onde se utiliza o CVX.

```
cvx_begin
    variable w(number_of_entries);

    maximize(miu'*w - gama*w'*cov*w);

    subject to
        (vec_ones')*w == 1; w>=0;
cvx_end;
```

IV. BARRIER METHOD

Por forma a melhorar o tempo de otimização, utilizou-se um *Interior Point Method*, visto que o problema pertence à categoria dos *Inequality and equality-constrained smooth problems*, categoria esta que é reduzida para uma sequência de *Equality-constrained smooth problems*, a partir da utilização de *Interior Point Methods*. Iremos descrever como fizemos esta transformação ao longo do capítulo.

Para esta abordagem, pretende-se reescrever o problema sob a forma de problema de otimização da seguinte forma:

$$\begin{aligned} &\text{minimize} && f_0(x) \\ &\text{subject to} && f_i(x) \leq 0 \\ &&& Ax = b \end{aligned} \quad (2)$$

As *constraints* de otimização devem portanto ser lineares ou inequações do tipo acima descrito em que os f_i são convexos e duas vezes diferenciáveis.

Por forma a traduzir a formulação de (2) numa formulação de um *Equality-constrained smooth problem*, e utilizando *Unconstrained Minimization Technique*, obtem-se (4), que é uma aproximação de (3).

$$\begin{aligned} &\text{minimize} && f_0(x) + \sum_{i=1}^m I_-(f_i(x)) \\ &\text{subject to} && Ax = b \end{aligned} \quad (3)$$

$$\begin{aligned} &\text{minimize} && f_0(x) + (1/t) \sum_{i=1}^m \log(f_i(x)) \\ &\text{subject to} && Ax = b \end{aligned} \quad (4)$$

Em (3), $I_{-}f_i(x)$ é a função indicadora e representa uma barreira para a função de custo, ou seja, para $f_i(x) \leq 0$, tem valor nulo sendo infinito para os restantes. Isto traduz-se na penalização infinita caso as *constraints* se aproximem da violação das condições impostas pelas inequações. Na equação (4), $(1/t)\log(f_i(x))$ representa a *log barrier*, ou seja, a aproximação da barreira à função indicadora. À medida que t aumenta, a *log barrier* aproxima melhor a função indicadora,

obtendo-se assim resultados cada vez mais semelhantes à solução ótima. Está ilustrado na figura 1 o aumento de t e a sua consequência na *log barrier*.

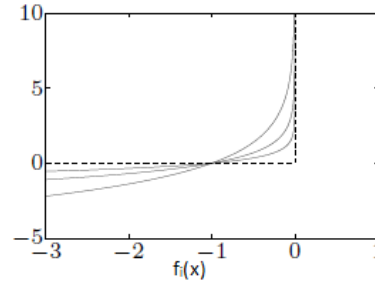


Fig. 1: Aproximação da função indicadora por vários valores de t

Não sendo possível a implementação de funções não diferenciáveis, faz-se uma aproximação logarítmica, representada em (4), da barreira onde é possível aplicar o método de *Newton*, pelo facto de a nova formulação ser *smooth*. Desta forma, escolhendo uma tolerância e um valor para t , será possível obter uma solução sub-ótima para o problema em apenas uma iteração. No entanto, não tendo forma de escolher o valor de t mais apropriado, utiliza-se o *Barrier Method* que permite efetuar várias otimizações da função $f(z)$, *homotopy method*. À medida que t tende para um valor elevado, espera-se que o valor sub ótimo, $x^*(t)$, se aproxime do valor ideal, x^* . Quando t tende para infinito espera-se que $x^*(t)$ tenda para x^* , e que este valor seja a solução do nosso problema. As soluções de cada iteração estão situadas ao longo do *central path* que corresponde ao conjunto de soluções $x^*(t)$. Estas soluções são caracterizadas pelas *KKT Conditions* e são, usualmente, denominadas de *central points*. O *Barrier Method* é iterativo, pelo que, a cada iteração, chamada *outer iteration*, é computada uma solução sub ótima, esta solução é depois utilizada com ponto inicial de próxima iteração, estende-se esta computação até que o *stopping criterion* seja verificado. A cada iteração o valor de t é aumentado por um fator μ , este fator é denominado de *barrier parameter* e controla o valor que a *log barrier* toma. O *stopping criterion* é obtido através do *duality gap* e é dado por $f(x^*(t)) - f^* \leq m/t$, em que m é o número de *constraints* existentes no problema e $f(x^*(t)) - f^*$ o *duality gap*, ou seja, a diferença entre a solução sub ótima e a ótima, e irá ser substituído por ϵ , ou nível de precisão, como se observa na descrição do método. Como referido acima, espera-se que, quando t tenda para infinito, $f(x^*(t)) - f^*$ tenda para zero.

No *Barrier Method* a escolha dos parâmetros t e μ é de grande importância. Caso μ seja muito pequeno irão ser necessárias muitas *outer iterations*, visto que o aumento de t será reduzido, por outro lado, caso μ seja muito elevado, irão ser necessárias muitas iterações do método de *Newton*, já que as soluções sub ótimas de iterações consecutivas irão estar muito afastadas entre si. Quanto ao valor de t inicial, se este for muito pequeno, irão ser também necessárias muitas *outer iterations*, a adaptação à função indicadora será feita lentamente, caso t inicial seja muito elevado, a primeira iteração do método de *Newton* (*first centering step*) irá requerir muitas iterações para calcular x^* inicial.

Pela impossibilidade de aplicação deste método com a formulação anterior devido à *constraint* $1^\top \omega = 1$, foi necessário recorrer a uma mudança de variável.

Deste modo, define-se ω pela equação (5).

$$\omega = Dz + b \quad (5)$$

As matrizes serão as seguintes:

$$D_{n,n-1} = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -1 & -1 & -1 & \cdots & -1 \end{pmatrix}$$

$$z_{n-1,1} = \begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_{n-1} \end{pmatrix} \quad b_{n,1} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix}$$

Com esta mudança de variável, a nova variável de otimização z , tem menos um grau de liberdade que a anterior, ω . O último investimento será assim a percentagem em falta do capital inicial.

Esta condição é conseguida através da equação matricial (5), onde a matriz D é uma matriz identidade nas primeiras $n-1$ linhas e a última linha desta matriz conter apenas "-1"s; a matriz b é uma matriz coluna de "0"s com apenas a ultima posição "1", sendo a *constraint* referida anteriormente assim respeitada.

Relativamente à segunda *constraint*, $\omega \in \mathbb{R}_+^n$, com a mudança de variável efetuada, tem-se $Dz + b \geq 0$. Como tal, o problema descrito pela equação (2) adaptado à situação em causa ficará escrito da seguinte forma:

$$\begin{aligned} &\text{minimize} && -\mu^\top (Dz + b) + \gamma (Dz + b)^\top \Sigma (Dz + b) \\ &\text{subject to} && -Dz - b \leq 0 \end{aligned} \quad (6)$$

Os $f_i(x)$ são assim $f_i(x) = -\delta_i^\top z - b_i$ onde δ_i^\top representa a linha i da matriz D .

Neste algoritmo, faz-se num primeiro passo a minimização com recurso ao método de *Newton*, sendo a função utilizada a apresentada na equação (7),

$$f(z) = t f_0(z) + \phi(z) \quad (7)$$

onde

$$f_0(z) = -\mu^\top (Dz + b) + \gamma (Dz + b)^\top \Sigma (Dz + b) \quad (8)$$

e

$$\phi = - \sum_{i=1}^m \log(-f_i(x)) \quad (9)$$

O termo ϕ evita que a minimização com o método de *Newton* vá para soluções não permitidas, uma vez que a função de otimização assume o valor infinito (*penalty*), obrigando o

método de *Newton* a calcular outro *step* para encontrar uma solução viável.

Para a aplicação do método de *Newton*, é ainda necessário o cálculo do gradiente e da hessiana da função $f(z)$. Como tal, o gradiente é obtido da seguinte forma:

$$\nabla f(z) = t(\nabla f_0(z))^\top + \nabla \phi(z) \quad (10)$$

onde

$$\nabla f_0(z) = -\mu^\top D + \gamma (Dz + b)^\top 2\Sigma D \quad (11)$$

e

$$\nabla \phi(z) = \sum_{i=1}^m \frac{1}{-f_i(z)} \nabla f_i(z) \quad (12)$$

A expressão da hessiana será obtida de forma idêntica pelas equações (13), (14) e (15).

$$\nabla^2 f(z) = t(\nabla^2 f_0(z))^\top + \nabla^2 \phi(z) \quad (13)$$

$$\nabla^2 f_0(z) = \gamma 2(D^\top \Sigma D) \quad (14)$$

$$\nabla^2 \phi(z) = \sum_{i=1}^m \frac{1}{-f_i^2(z)} \nabla f_i(z) (\nabla f_i(z))^\top \quad (15)$$

Por último, o *Barrier Method* converge num número finito de iterações pelo indicado pelo ponto três do algoritmo. O número de *centering steps* requerido ao *Barrier Method* para atingir um nível de precisão ϵ são:

$$\frac{\log(\frac{m}{t(0)\epsilon})}{\log(\mu)} + 1 \quad (16)$$

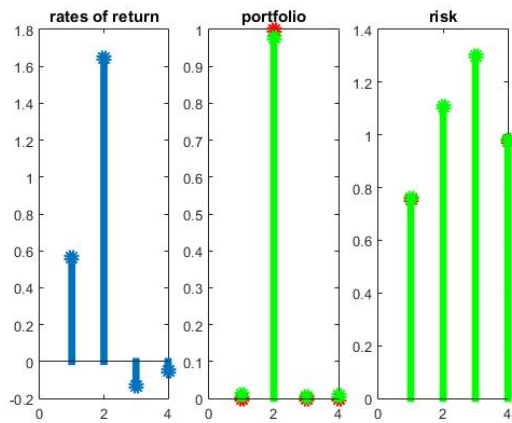
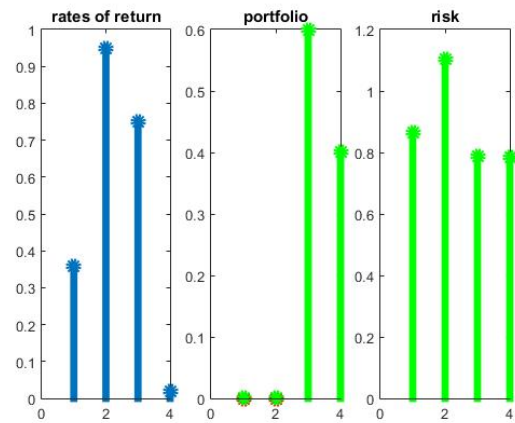
V. RESULTADOS NUMÉRICOS

How does it really work in practice? Provide simulated performance metrics. If possible, compare with other well-known methods.

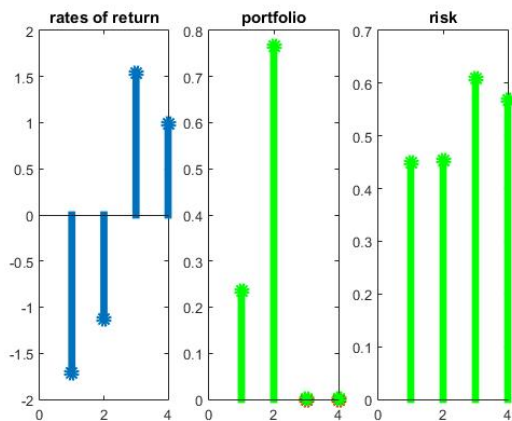
Nas figuras que irão ser expostas tem-se a verde os resultados do *Barrier Method* e a vermelho do *CVX*, tanto os retornos como os riscos dos *assets* são iguais para ambos os programas. Para verificar o correto funcionamento de ambos os programas realizaram-se testes em que os resultados podem ser facilmente previstos, tais como:

- Igualar γ a zero, ou seja, caso o utilizador não tenha quaisquer preocupações com o risco associado a cada investimento. Espera-se, portanto, que o portfólio obtido seja investir todo o capital no *asset* com maior retorno, independentemente do risco.

Na figura 2 observa-se que, de facto, o capital foi investido no *asset* de maior retorno. Existem pequenas diferenças entre o portfólio apresentado pelos dois programas, pelo que se pode observar que no resultado do *CVX* o investimento é total no *asset* de maior retorno, enquanto que no resultado do *Barrier Method* existem pequenas quantias investidas pelos outros *assets*.

Fig. 2: Portfólio com γ igual a 0Fig. 4: Portfólio com γ igual a 100 e retornos positivos

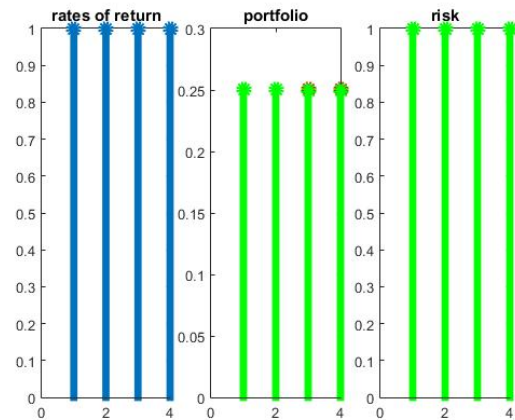
- Igualar γ a 100 (100 é um valor arbitrário), pretende-se ilustrar uma aversão ao risco muito elevada por parte do utilizador. Espera-se um portfólio bastante conservador, ou seja, que seja investido capital apenas no(s) *asset(s)* com menor risco associado. Quando γ tende para infinito, o portfólio será investido somente nos *assets* com menor risco associado, ao contrário do caso em que γ é igual a 100, onde existem pequenos investimentos nos outros *assets*.

Fig. 3: Portfólio com γ igual a 100 e alguns retornos negativos

Na figura 3 a solução dada por ambos os programas é em tudo semelhante, note-se que, apesar das *rates of return* dos *assets* de menor risco serem negativas, a aversão ao risco do utilizador irá gerar um portfólio em que a importância dada ao risco é bastante superior à dos retornos. A divisão do portfólio pelos dois *assets* de menor risco é a esperada, ambos os programas escolheram investir mais capital no *asset* com maior retorno.

No caso da figura 4, o portfólio segue o mesmo padrão da figura 3, aposta-se o capital nos *assets* de menor risco e entre eles, investe-se a grande parte no *asset* que apresenta maior retorno.

- Igualar todos o *returns* e riscos de todos os *assets*. O portfólio deverá dividir o capital igualmente por todos os *assets*.

Fig. 5: Portfólio com riscos e retornos iguais para todos os *assets*

Na figura 5 os resultados do portfólio refletem o esperado, visto que ambos os programas repartiram igualmente o capital por todos os *assets*.

- Igualar apenas os *returns* de todos os *assets*. O portfólio deverá investir a grande parte do seu capital no *asset* de menor risco. Quando se aumenta γ nesta situação é esperado que a quantidade investida no *asset* de menor risco aumente.

No caso enunciado acima, a solução de portfólio ótima é intuitiva, os programas deverão investir todo o capital no *asset* de menor risco, visto que todos retornam o mesmo valor do investimento inicial. É isto que se comprova na figura 6.

The commentary of the plots should not just repeat the graphically obvious such as "the solution is different from the original signal", but explain, for example, how this difference relates to quick changes on the signal. Is the solution too slow to follow the signal variations? What is the magnitude of the

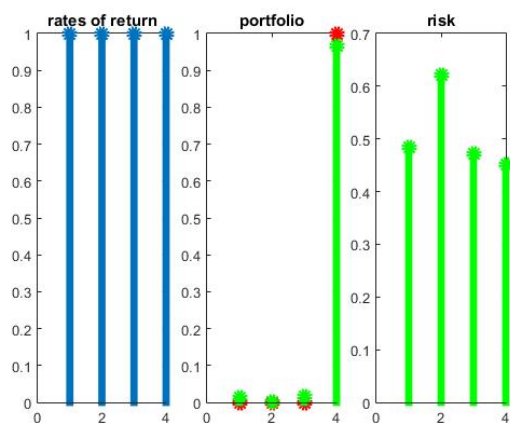


Fig. 6: Portfólio com retornos iguais para todos os *assets*

error?

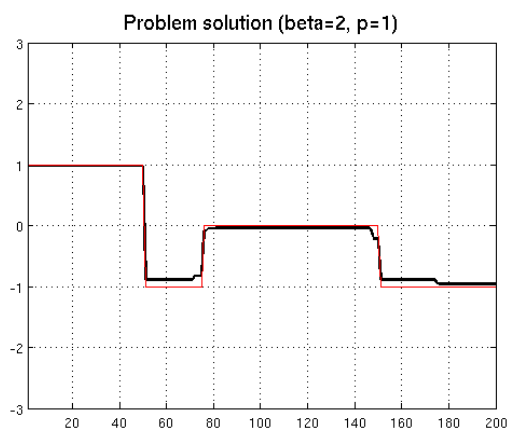


Fig. 7: Solution for the L1 norm cost function.

You might also want to vary the problem's parameters and see how the solution evolves. Don't forget to explain why is the choice of parameters in the solution depicted in Figure 7 better than the others.

Figures should be chosen wisely. You can never lay out the whole parameter space, so provide insight into which parameters are significant over what range and which ones are less important.

VI. CONCLUSÕES

In general a short summarizing paragraph will be sufficient. It should not simply repeat material from the Abstract or Introduction. In some cases it's possible to now make the original claims more concrete, *e.g.*, by referring to quantitative results.

REFERENCES

- [1] [CVX15] <http://cvxr.com/cvx>, Junho 2015
- [2] [BV_CVXSLIDES] https://web.stanford.edu/boyd/cvxbook/bv_cvxslides.pdf