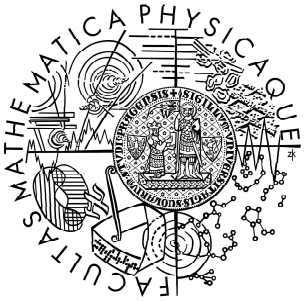
Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

BAKALÁŘSKÁ PRÁCE



Jan Voříšek

Vybrané metody optimalizace ve financích

Katedra pravděpodobnosti a matematické statistiky

Vedoucí bakalářské práce: Doc. RNDr. Jan Hurt, CSc.,
Studijní program: Matematika
Studijní obor: Finanční matematika

2006

Na tomto místě bych rád poděkoval Doc. RNDr. Janu Hurtovi, CSc. za trpělivost a cenné rady při vedení bakalářské práce.

Prohlašuji, že jsem svou bakalářskou práci napsal samostatně a výhradně s použitím citovaných pramenů. Souhlasím se zapůjčováním práce a jejím zveřejňováním.

V Praze dne 31. května 2006 Jan Voříšek

Obsah

Název práce: Vybrané metody optimalizace ve financích
Autor: Jan Voříšek
Katedra (ústav): Katedra pravděpodobnosti a matematické statistiky
Vedoucí bakalářské práce: Doc. RNDr. Jan Hurt, CSc.
e-mail vedoucího: hurt@karlin.mff.cuni.cz

Abstrakt: V předložené práci studujeme metody vytváření optimálního portfolia podle teorie H. Markowitze. V praktické ukázce konstrukce portfolia podle uvedené teorie z akcií báze indexu PX50 demonstrujeme možnosti využití výpočetního systému Mathematica. V první části je detailně popsána zabudovaná funkce systému Mathematica NMinimize (NMaximize). Druhá část uvádí obecnou teorii konstrukce portfolia s minimálním rozptylem a portfolia s maximálním Sharpeovým poměrem. Třetí část obsahuje praktické ukázky výpočtů optimálního portfolia pomocí zabudovaných funkcí systému Mathematica.

Klíčová slova: Portfolio, Markowitz, Mathematica

Title: Selected Optimization Methods in Finance
Author: Jan Voříšek
Department: Department of Probability and Mathematical Statistics
Supervisor: Doc. RNDr. Jan Hurt, CSc.
Supervisor's e-mail address: hurt@karlin.mff.cuni.cz

Abstract: In the present work we study construction methods of optimal portfolio according to theory of H. Markowitz. We illustrate potential of system Mathematica on construction of the optimal portfolio from stocks of index PX50. The built-in function NMinimize (NMaximize) is described in the first chapter. In second chapter we introduce the general theory of the minimum-variance portfolio and potfolio with maximum Sharpe's ratio. There are practical demonstrations of calculation of an optimal portfolio using the build-in functions in the third chapter.

Keywords: Portfolio, Markowitz, Mathematica

0.1 Úvod

Vzhledem k tomu, že je většina investic ovlivňována náhodnými faktory, nemůžeme dopředu přesně odhadnout, jak se v budoucnu bude měnit jejich výnosnost. Tím vzniká riziko finanční ztráty, které je třeba odhadnout a volit takové složení investic, aby riziko bylo co nejnižší. Staré pravidlo nabádalo k rozdělení disponibilních zdrojů na tři části: třetinu uložit, třetinu investovat do akcií a za zbytek nakoupit zlato. Po vzniku teorie optimálních portfolií H. Markowitze, se začalo riziko měřit pomocí směrodatné odchylky. Portfolio je soubor různých investic (peněžní hotovost, cenné papíry, nemovitosti atd.), který racionální investor vytváří se záměrem minimalizovat riziko spojené s investováním a současně maximalizovat výnos z investic. Konstrukce portfolia takového investora je tak hledání optimálního kompromisu mezi výší rizika a výnosu.

Cílem této práce je demonstrovat možnost využití systému Mathematica pro optimalizační úlohy ve financích. Ke konstrukci optimálního portfolia podle H. Markowitze použijeme zabudované funkce. Nejdůležitější z těchto funkcí NMinimize je detailně popsána v první části. Pak následuje obecná teorie konstrukce portfolia podle H. Markowitze a poslední část se věnuje konkrétním příkladům konstrukce portfolia z akcií báze indexu PX50.

Na přiloženém CD jsou uloženy soubory s daty a výpočty v programu mathematica.

Kapitola 1

Popis funkcí systému Mathematica

1.1 Funkce NMinimize

Funkce NMinimize (i NMaximize) je implementována v programu Mathematica verze 5.2. Jak uvádí [2] je funkce Minimize stejně jako její numerická obdoba NMinimize určena pro hledání globálního minima. Na rozdíl od Minimize funkce NMinimize nemůže obvykle zaručit nalezení *globálního* minima, přesto však podává dobré výsledky, pokud je minimalizovaná funkce poměrně hladká a nemá příliš mnoho lokálních extrémů. Na rozdíl od numerické funkce FindMinimum není potřeba do funkce NMinimize zadávat počáteční bod vyhledávání. NMinimize je třeba zadat pouze funkci a proměnou, vzhledem ke které chceme funkci minimalizovat. Navíc je možné zadat množinu omezení, ve které bude vyhledáváno globální minimum. Omezení mohou obsahovat seznam nebo logické kombinace rovností, nerovností a specifikací definičních oborů proměnných. Rovnosti a nerovnosti mohou být nelineární. Ostré nerovnosti budou převedeny na neostré díky limitám pracujícím s přibližnými čísly. Standardně jsou všechny proměnné považovány za reálné. Pokud požadujeme pouze celočíselné proměnné, specifikujeme je pomocí $x \in Integers$. V současné verzi jsou přípustné pouze obory reálných a celých čísel. Obor může být specifikován pouze pro proměnné. Splnění podmínek se kontroluje penalizací, pokud body opustí přípustný prostor. Pokud má NMinimize začít pracovat, potřebuje počáteční oblast, ve které má začít, podobně jako se u ostatních numerických metod zadává počáteční bod. Počáteční oblast můžeme specifikovat zadáním horní a dolní hranice pro

každou proměnnou. To buď zahrneme do podmínek $a \leq x \leq b$ nebo do proměnných {x,a,b}. Pokud je zadáno obojí, jako počáteční oblast se používají hranice uvedené v proměnných a podmínky se použijí pouze jako podmínky. Pokud není uvedena žádná počáteční oblast pro proměnnou x, použije se oblast $-1 \leq x \leq 1$. Různé proměnné mohou mít počáteční oblasti definované různým způsobem. NMinimize provádí několik algoritmů pro nalezení vázaného globálního minima. Využívané metody jsou dostatečně flexibilní, aby zvládly funkce, které nejsou diferencovatelné nebo spojité. Hledání globálního optima může být dosti složité, dokonce bez omezení, takže použité metody mohou selhat. Často je úspěšný postup optimalizovat funkci několikrát s různými počátečními podmínkami a volbami metody a vybrat nejlepší výsledek.

Funkci NMinimize v programu Mathematica voláme následujícím způsobem:
- NMinimize[*f*, {*x, y,...*}] - minimalizuje *f* numericky vzhledem k *x, y,...* .
- NMinimize[{*f, podm*}, {*x, y,...*}] - minimalizuje *f* numericky za podmínek *podm*.
NMinimize vrátí list ve formě $\{f_{min}, \{x \to x_{min}, y \to y_{min}, \dots\}\}$.
Pokud NMinimize zjistí, že podmínky nemohou být splněny, vrátí následující list {Infinity, {$x \to$ Indeterminate,...}}.

Možnosti nastavení NMinimize

jméno volby	natavená hodnota	
AccuracyGoal	Automatic	počet desetinných míst
EvaluationMonitor	None	sledování výpočtu po krocích
MaxIterations	100	maximální počet iterací
Method	Automatic	jaká metoda se použije při minimalizaci
PrecisionGoal	Automatic	počet cifer v jedné hodnotě
StepMonitor	None	vyhodnocovaný výraz v každém kroku výpočtu
WorkingPrecision	MachinePrecision	počet použitých cifer při vnitřním výpočtu

NMinimize a NMaximize používají několik optimalizačních metod: *Automatic, DifferentialEvolution, Nelder-Mead, RandomSearch* a *SimulatedAnnealing*. Optimalizační metoda je ovládána pomocí volby Method, která má

tvar pouze volby metody nebo seznamu, jehož první položkou je volba metody a další položky seznamu specifikují nastavení zvolené metody.

Pokud je nastaveno Method→*Automatic*, NMinimize na základě typu zadání vybere, kterou metodu použít při řešení. Pokud jsou podmínky a účelová funkce lineární, použije se při výpočtu zabudovaná funkce*LinearPrograming*. Pokud se v zadání vyskytují celočíselné proměnné nebo pokud hlava účelové funkce není numerickou funkcí, pro výpočet se použije *DifferentialEvolution*. Pro všechny další případy se používá *Nelder-Mead*. Pokud *Nelder-Mead* podává chabé výsledky, je vyzkoušen výpočet pomocí *DifferentialEvolution*.

DifferentialEvolution je vývojový algoritmus, který vytváří populaci vzorů, $x_1, \ldots, x_n$, reprezentovanou jako vektor reálných čísel ("geny"). Každá iterace vybere pro každé x_i náhodná celá čísla a,b,c a vytvoří dvojice x_i, $y_i = x_a + \gamma(x_b - x_c)$, kde γ je hodnota *ScalingFactor*. Potom je x_i spárováno s y_i podle hodnoty *CrossProbability*. Toto spojení dává potomka z_i, který soupeří s x_i o místo genu x_i v populaci. Standardní hodnota *SearchPoints* je Automatic, Min[10*d,50], kde d je počet proměnných. *DifferentialEvolution* je poměrně robustní , ale následkem poměrně velkého množství bodů, které vytváří, je obecně pomalejší než ostatní metody.

Specifické možnosti nastavení DifferentialEvolution

jméno volby	nastavená hodnota	
"CrossProbability"	0.5	pravděpodobnost, že gen vznikl z x_i
"InitialPoints"	Automatic	soubor výchozích bodů
"PenaltyFunction"	Automatic	funkce, která penalizací omezuje nepřípustné body
"PostProcess"	Automatic	jestli k následnému postupu použít lokální režim vyhledávání
"RandomSeed"	0	počáteční hodnota pro generátor náhodných čísel
"ScalingFactor"	0.6	váha rozdílu $(x_b - x_c)$ při vytváření páru
"SearchPoints"	Automatic	velikost populace použitá pro vývoj
"Tolerance"	0.001	tolerance při překročení omezení

Nelder-Mead je realizace Nelder-Meadova simplexového algoritmu. Ten byl navržen jako metoda pro minimalizování reálné funkce $f(\mathrm{x})$ pro $x \in R^n$. Podle [3] Nelder-Meadova metoda vyžaduje stanovení čtyř parametrů: koeficienty reflexe (ρ), expanze (χ), kontrakce (γ) a sražení (σ). Téměř univerzální volba používaná ve standardním Nelder-Meadově algoritmu je ρ=1 (*ReflectRatio*), χ=2 (*ExpandRatio*), $\gamma = \frac{1}{2}$ (*ContractRatio*) a $\sigma = \frac{1}{2}$ (*ShrinkRatio*). Na začátku k-té iterace je dán nedegenerovaný simplex Δ_k, spolu s $n+1$ vrcholy, kde každý z nich je dán bodem v R^n. Každá iterace začíná seřazením těchto vrcholů do do $x_1^{(k)}, \ldots, x_{n+1}^{(k)}$, podle $f_1^{(k)} \leq f_2^{(k)} \leq \cdots \leq f_{n+1}^{(k)}$, kde $f_i^{(k)}$ označuje $f(x_i^{(k)})$. K-tá iterace vygeneruje sadu $n+1$ vrcholů, které určují simplex pro další iteraci tak, že $\Delta_{k+1} \neq \Delta_k$. Protože chceme minimalizovat f, označujeme $x_1^{(k)}$ jako nejlepší a $x_{n+1}^{(k)}$ jako nejhorší bod nebo vrchol. Stejně tak označujeme $f(x_{n+1}^{(k)})$ jako nejhorší funkční hodnotu atd.

Průběh jedné iterace Nelder-Meadova algoritmu vypadá takto. Nejdřív je potřeba seřadit vrcholy podle $f_1^{(k)} \leq f_2^{(k)} \leq \cdots \leq f_{n+1}^{(k)}$. Potom algoritmus najde těžiště n nejlepších bodů $\bar{x}$, kde $\bar{x} = \sum_{i=1}^{n} \frac{x_i}{n}$. Dále spočítá bod *reflexe*

$$x_r = \bar{x} + \rho(\bar{x} - x_{n+1} = (1+\rho)\bar{x} - \rho x_{n+1}$$

a vyčíslí $f_r = f(x_r)$. Pokud $f_1 \leq f_r < f_n$, bod x_r splňuje podmínky a iterace je ukončena. Pokud $f_r < f_1$, spočítá bod *expanze*

$$x_e = \bar{x} + \chi(x_r - \bar{x}) = \bar{x} + \chi\rho(\bar{x} - x_{n+1}) = (1 + \chi\rho)\bar{x} - \chi\rho x_{n+1}$$

a vyčíslí $f_e = f(x_e)$. Pokud $f_e < f_r$, přijme x_e a ukončí iteraci. Naopak pokud $f_e \geq f_r$, přijme x_r a ukončí iteraci. Pokud $f_r \geq f_n$, provede se kontrakce mezi $\bar{x}$ a lepším z bodů x_{n+1} a x_r. *Vnější kontrakce* nastává v případě, že $f_n \leq f_r < f_{n+1}$. Spočítá bod

$$x_c = \bar{x} + \gamma(x_r - \bar{x}) = \bar{x} + \gamma\rho(\bar{x} - x_{n+1}) = (1 + \gamma\rho)\bar{x} - \gamma\rho x_{n+1}.$$

Vyčíslí se $f_c = f(x_c)$. Pokud $f_c \leq f_r$, přijme x_c a ukončí iteraci. Jinak přistoupí k poslednímu kroku sražení. *Vnitřní kontrakce* nastává v případě, že $f_r \geq f_{n+1}$. Spočítá bod

$$x_{cc} = \bar{x} - \gamma(\bar{x} - x_{n+1}) = (1 - \gamma)\bar{x} + \gamma x_{n+1}$$

a vyčíslí $f_{cc} = f(x_{cc})$. Pokud $f_{cc} < f_{n+1}$, přijme x_c a ukončí iteraci. Jinak přistoupí k poslednímu kroku *sražení*. *Sražení* spočívá ve vypočtení f v n bodech $v_i = x_1 + \sigma(x_i - x_1)$, pro i=2,...,n+1. Neuspořádané vrcholy simplexu následující iterace jsou $x_1, v_2, \ldots, v_{n+1}$.

Specifické možnosti nastavení NelderMead

jméno volby	nastavená hodnota	
"ContractRatio"	0.5	koeficient použitý pro kontrakci
"ExpandRatio"	2.0	koeficient použitý pro expanzi
"InitialPoints"	Automatic	soubor výchozích bodů
"PenaltyFunction"	Automatic	funkce, která penalizací omezuje nepřípustné body
"PostProcess"	Automatic	jestli k následnému postupu použít lokální režim vyhledávání
"RandomSeed"	0	počáteční hodnota pro generátor náhodných čísel
"ReflectRatio"	1.0	koeficient použitý pro reflexi
"ShrinkRatio"	0.5	koeficient použitý pro sražení
"Tolerance"	0.001	tolerance při překročení omezení

RandomSearch je jednoduchá metoda, která generuje velký počet bodů a každý z nich použije jako počáteční bod pro lokální vyhledávání. Možné metody lokálního vyhledávání jsou *Automatic*, *FindMinimum* a *InteriorPoint*. Standardní hodnota Method je *Automatic*, která používá *FindMinimum*. Když *FindMinimum* nemůže mít omezení jako argumenty, používá se modifikovaná účelová funkce. Pokud je zadáno Method → *InteriorPoint*, je použita nelineární metoda vnitřního bodu. Základní nastavení hodnoty pro počet *SearchPoints* je Automatic, což znamená Min[10*d,100], kde d je počet proměnných. *RandomSearch* je rychlá metoda, ale jsou problémy s dimenzí prostoru. Také má hodně stejných omezení jako *FindMinimum*. Není příliš vhodná pro řešení diskrétních úloh a úloh, kde derivace nepodávají dobrou představu o řešeném problému.

Specifické možnosti nastavení RandomSearch

jméno volby	nastavená hodnota	
"InitialPoints"	Automatic	sada počátečních bodů
"Method"	Automatic	která metoda bude použita při minimalizaci
"PenaltyFunction"	Automatic	funkce, která penalizací omezuje nepřípustné body
"PostProcess"	Automatic	jestli k následnému postupu použít lokální režim vyhledávání
"RandomSeed"	0	počáteční hodnota pro generátor náhodných čísel
"SearchPoints"	Automatic	velikost populace použitá pro vývoj
"Tolerance"	0.001	tolerance při překročení omezení

SimulatedAnnealing je použití hledání pomocí zkreslené náhodné procházky. Generuje náhodný počet počátečních bodů a pro každý počáteční bod vybere náhodný směr, kterým se vydá. Pokud se pohybem dostane do lepšího bodu, přijme ho. Pokud se nedostane do lepšího bodu, vypočítá pravděpodobnost ρ a porovná jí s náhodnou hodnotou $r \in (0,1)$ a pokud $r < \rho$, nový bod přijme i přes to, že neznamená zlepšení. Pravděpodobnost ρ je dána vztahem $\rho = e^{b[i,\Delta f,f_0]}$, kde b je funkce definovaná *BoltzmannovýmExponentem*, i je současná iterace, Δf je změna hodnoty účelové funkce a f_0 je hodnota účelové funkce z předcházející iterace. Tento postup je opakován pro každý počáteční bod, dokud není dosaženo maximálního počtu iterací, není nalezen bod konvergence nebo nezůstane nepřetržitě v jednom bodě po dobu (počet kroků) zadanou v *LevelIterations*. Pokud je *BoltzmannExponent* $\rightarrow$ Automatic, použitá funkce je Function[{i,df,f0},- df * Log[i+1]/10]. Standardní hodnota *SearchPoints* je Automatic, což znamená Min[2*d,50], kde d je počet proměnných. *SimulatedAnnealing* je obecně o trochu rychlejší než *DifferentialEvolution*, ale neřeší tak přesně některé úlohy.

Specifické možnosti nastavení SimulatedAnnealing

jméno volby	nastavená hodnota	
"BoltzmannExponent"	Automatic	exponent pravděpodobnostní funkce
"InitialPoints"	Automatic	sada počátečních bodů
"LevelIterations"	50	maximální počet iterací, kdy může proces zůstat v jednom bodě
"PenaltyFunction"	Automatic	funkce, která penalizací omezuje nepřípustné body
"PerturbationScale"	1.0	stupnice pro náhodný skok
"PostProcess"	Automatic	jestli k následnému postupu použít lokální režim vyhledávání
"RandomSeed"	0	počáteční hodnota pro generátor náhodných čísel
"SearchPoints"	Automatic	velikost populace použitá pro vývoj
"Tolerance"	0.001	tolerance při překročení omezení

Kapitola 2

Markowitzovo portfolio

2.1 Předpoklady

Podle [1], je ke konstrukci optimálního portfolia podle teorie H.Markowitze zapotřebí několika rozumných a několika umělých omezujících ekonomických předpokladů. Eficientní trh musí splňovat následující požadavky (v závorce je uveden komentář ke každému požadavku):
1. Investoři rozhodují o složení svého portfolia výhradně na základě očekávaných výnosů a kovarianční struktuře výnosu (realistické)
2. Investoři z portfolií se stejným rizikem preferují portfolio s nejvyšším očekávaným výnosem (realistické; racionální chování)
3. Investoři z portfolií se stejným očekávaným výnosem preferují portfolio s nejnižším rizikem (realistické; odpor k riziku)
4. Aktiva jsou absolutně dělitelná (omezující; obchodování na burze se většinou provádí v lotech (sto akcií), při obchodování s méně akciemi se připočítávají extra náklady)
5. Investiční horizont je jedna časová perioda (realistické)
6. Neexistence transakčních nákladů a daní (omezující; daně a transakční náklady mohou být částečně začleněny do výnosů pokud jsou lineárně závislé na velikosti transakce)
7. Existuje pouze jediná bezriziková úroková míra pro všechny investory, kteří mohou poskytnout nebo si mohou půjčit libovolně velký obnos za tuto bezrizikovou úrokovou míru (nerealistické)
8. Všechna uvažovaná aktiva jsou obchodovatelná (realistické)
9. Jsou povoleny prodeje nakrátko (omezující; koupě k aktiv k financování jiného projektu, omezeno legislativou)

10. Žádný investor nemůže podstatně ovlivnit výnosy příslušného aktiva (omezující; neexistuje žádný investor disponující výrazně většími prostředky než ostatní investoři)
11. Všechny informace (o střední hodnotě a kovarianci) jsou stejně přístupné všem investorům ve stejném čase (omezující)

Za splnění těchto předpokladů vzniká tržní rovnováha (pokud investoři mají přesné informace o trhu a chovají se racionálně).

2.2 Obecná konstrukce

Při konstrukci optimálního portfolia uvažujeme N aktiv a disponibilní zdroje výše 1. Potom portfolio je vektor $\mathbf{x}=(x_1,\ldots,x_N)^T$, kde x_n představuje část bohatství investovanou do n-tého aktiva, $n=1,\ldots,N$, proto platí $\mathbf{1}^T\boldsymbol{x}=1$. Obecně nepředpokládáme, že $x_n \geq 0$. Pokud připustíme $x_n < 0$ hovoříme o prodejích nakrátko (investor může prodat záruku, kterou nevlastní. Investor si půjčuje příslušné aktivum). Dále budeme předpokládat, že výnos (míra výnosnosti) z N zmíněných aktiv je náhodná proměnná $\boldsymbol{\rho}=(\rho_1,\ldots,\rho_N)^T$ s očekávanými výnosy $\boldsymbol{r}=\mathrm{E}\boldsymbol{\rho}=(r_1,\ldots,r_N)^T$ a kovarianční maticí $\boldsymbol{V}=(\sigma_{ij})$, kde $\sigma_{ij}=\mathrm{cov}(\sigma_i,\sigma_j)$, i, j=1,...,N. Prvky na diagonále matice $\boldsymbol{V}$ můžeme označit $\sigma_i^2:=\sigma_{ii}$. Směrodatná odchylka výnosů je $\sigma_i:=\sqrt{\sigma_{ii}}$. Očekávaný výnos portfolia p, reprezentovaného vektorem $\boldsymbol{x}$ je $r_p=\boldsymbol{r}^T\boldsymbol{x}$ a rozptyl výnosu portfolia p je $\sigma_p^2=\boldsymbol{x}^T\boldsymbol{V}\boldsymbol{x}$. Potom riziko portfolia p je směrodatná odchylka $\sigma_p=\sqrt{\sigma_p^2}$.

Konstrukce optimálního portfolia pouze na základě znalosti očekávaných výnosů a kovarianční matice výnosů je známa jako Markowitzův přístup. Optimalizaci je možné provést dvěma způsoby.

1. Minimalizovat $\frac{1}{2}\boldsymbol{x}^T\boldsymbol{V}\boldsymbol{x}$ za podmínek $\mathbf{1}^T\boldsymbol{x}=1$ a $\boldsymbol{r}^T\boldsymbol{x}=\mu$, kde μ je požadovaný očekávaný výnos. Jinak řečeno investor hledá očekávaný výnos μ s co nejmenším rizikem. Odpovídajícímu portfoliu se říká portfolio s minimálním rozptylem. Minimalizování rizika je ekvivalentní minimalizování rozptylu.

2. Maximalizovat Sharpeův poměr (Sharpeovu míru portfolia) $\frac{\boldsymbol{r}^T\boldsymbol{x}}{\sqrt{\boldsymbol{x}^T\boldsymbol{V}\boldsymbol{x}}}$, což je očekávaný výnos portfolia dělený rizikem portfolia za podmínky $\mathbf{1}^T\boldsymbol{x}=1$. Někdy se maximalizuje modifikovaná Sharpeova míra portfolia, tj. poměr nadvýnosu ku riziku $\frac{\boldsymbol{r}^T\boldsymbol{x}-r_0}{\sqrt{\boldsymbol{x}^T\boldsymbol{V}\boldsymbol{x}}}$, kde r_0 je bezriziková úroková míra.

U obou způsobů je nutné uvažovat definiční obor složek vektoru $\boldsymbol{x}$ a vlastnosti kovarianční matice $\boldsymbol{V}$. Pokud $x_i \geq 0$ pro i=1,...,N, prodeje nakrátko jsou zakázány. V případě povolení prodejů nakrátko mohou být složky vektoru $\boldsymbol{x}$ i záporné. U kovarianční matice $\boldsymbol{V}$ rozlišujeme, jestli je pozitivně

definitní nebo je pozitivně semidefinitní, což naznačuje, že existují bezrizikové aktiva nebo dvě aktiva, která mají korelaci mezi výnosy v absolutní hodnotě rovnou jedné, tj. jsou perfektně korelovaná. V takovém případě je matice $\boldsymbol{V}$ singulární a nemá smysl maximalizovat Sharpeův poměr.

Kapitola 3

Konstrukce optimálního portfolia

3.1 Příprava dat

Obecnou teorii budeme ilustrovat na konstrukci optimálního portfolia složeného z akcií zařazených do báze indexu PX 50 společností ČEZ, ERSTE BANK, ČESKÝ TELECOM, KOMERČNÍ BANKA, UNIPETROL, PHILIP MORRIS ČR, ZENTIVA, ORCO PROPERTY GR a CENT EURO MEDIA podle teorie H. Markowitze pomocí funkcí implementovaných v programu Mathematica. Tento systém umožňuje zadání jednoho příkazu několika různými způsoby. Proto je možné dostat se ke stejnému výsledku při použití rozdílných příkazů, než které jsou v této práci uvedeny. Data cen akcií jsou z druhého pololetí roku 2005 pro každý den, kdy byly obchodovány, což představuje dostatečnou databázi 126 výnosů pro každý titul. Před začátkem samotné optimalizace je užitečné zadat vektory cen akcií jednotlivých firem do funkce

vynos[spol_] := (Drop[spol, -1] - Drop[spol, 1]) / Drop[spol, 1],

která z nich vytvoří vektory výnosů jednotlivých firem. Funkce *vynos* počítá výnosy podle vzorce $R_t = \frac{P_{t+1}-P_t}{P_t}$ tak, že pracuje s vektorem cen akcie, ze kterého vypustí buď první nebo poslední prvek. Výsledek je opět vektor. Z výnosových vektorů vytvoříme matici, kde v každém sloupci jsou výnosy jedné společnosti a v řádku jsou výnosy za stejné období. Tato výnosová matice představuje základ pro následující praktické výpočty v programu Mathematica, neboť z ní lze snadno zabudovanými funkcemi získat odhad

vektoru očekávaných výnosů označený r a kovarianční matici V. Matici uložíme příkazem

Export[”umístění a název souboru”, název matice,”Table”]

Načtení matice výnosů, kterou nazveme data, proběhne následujícím způsobem

data = Import[”umístění a název souboru”, ”Table”]

Dále je třeba zavolat standardní balík popisné statistiky příkazem

Needs[”Statistics‘MultiDescriptiveStatistics‘”]

Tento balík obsahuje funkce Mean a CovarianceMatrix. Sérií příkazů

r = Mean[data];
V = CovarianceMatrix[data];
n = Length[r]; X = Table[x_i, {i, n}];

získáme vektor očekávaných výnosů r, kovarianční matici výnosů V, a vektor aktiv X. Protože konstruujeme portfolio z výše uvedených akcií, můžeme předpokládat, že kovarianční matice V je pozitivně definitní, neboť případ bezrizikového aktiva nebo perfektně korelovaných výnosů dvou aktiv je velmi nepravděpodobný. Protože matice V je určitě pozitivně semidefinitní, stačí nám pro ověření pozitivní definitnosti spočítat determinant zabudovanou funkcí

Det[V],

který vychází $3.2514308516475683 * 10^{-34}$, takže matice V je sice formálně pozitivně definitní, ale numerické výpočty na ní založené jsou nestabilní, a proto je musíme brát s rezervou. Korelační matice však ukazuje, že nejvyšší korelace je 0.54 mezi společnostmi ČEZ a KB.

$$\begin{pmatrix}
1. & 0.2 & 0.32 & 0.54 & 0.41 & 0.17 & 0.49 & 0.5 & 0.18 \\
0.20 & 1. & 0.13 & 0.23 & 0.12 & 0.07 & 0.21 & 0.11 & 0.09 \\
0.32 & 0.13 & 1. & 0.33 & 0.36 & 0.13 & 0.36 & 0.35 & 0.08 \\
0.54 & 0.23 & 0.33 & 1. & 0.49 & 0.21 & 0.39 & 0.51 & 0.07 \\
0.41 & 0.12 & 0.36 & 0.49 & 1. & 0.12 & 0.32 & 0.44 & 0.09 \\
0.17 & 0.07 & 0.13 & 0.21 & 0.12 & 1. & 0.14 & 0.16 & 0.25 \\
0.49 & 0.21 & 0.36 & 0.39 & 0.32 & 0.14 & 1. & 0.49 & 0.20 \\
0.50 & 0.11 & 0.35 & 0.51 & 0.44 & 0.16 & 0.49 & 1. & 0.03 \\
0.18 & 0.09 & 0.08 & 0.07 & 0.09 & 0.25 & 0.20 & 0.03 & 1.
\end{pmatrix}$$

Do listu *jmena* uložíme názvy společností ve stejném pořadí v jakém jsme umístili vektory výnosů do matice *data*. List *jmena* vypadá následovně

{"ČEZ", "ERSTE BANK", "ČESKÝ TELECOM", "KOMERČNÍ BANKA", "UNIPETROL", "PHILIP MORRIS ČR", "ZENTIVA", "ORCO PROPERTY GR", "CENT EURO MEDIA"}

3.2 Minimalizace rozptylu

Minimalizaci rozptylu provedeme hledáním minima funkce $\frac{1}{2}\boldsymbol{x}^T\boldsymbol{V}\boldsymbol{x}$ za podmínek $\mathbf{1}^T\boldsymbol{x}$=1 a $\boldsymbol{r}^T\boldsymbol{x}$=$\mu$, kde μ je požadovaný očekávaný výnos. V konkrétním příkladu je zvoleno μ=0.004.

1. Minimalizace metodou Lagrangeových multiplikátorů. Předpokládáme povolení prodejů nakrátko a neexistenci bezrizikových aktiv. Lagrangeova funkce je dána vztahem

Lagrange = $\frac{1}{2}$X.V.X+λ_1(1-ones. X)+λ_2(μ - r.X),

kde ones je vektor jedniček o délce n. Lagrangeovu funkci derivujeme podle všech proměnných X a Lagrangeových multiplikátorů λ_1,λ_2,

derivace = D[Lagrange,{{X,λ_1,λ_2}}// Flatten]

Dostaneme list n+2 výrazů, který se rovná nule. Abychom dostali soustavu rovnic, použijeme

soustava = Thread[derivace == 0].

V posledním kroku vyřešíme soustavu

OptimalniPortfolio = (X/.Solve[soustava,{X,λ_1,λ_2}//Flatten])//Flatten

a dostaneme list řešení $x_1, \ldots, x_n$ uložený v proměnné OptimalniPortfolio. Konkrétně v našem případě se po příkazu

Print["Optimální portfolio: ",{jmena,OptimalniPortfolio} //Transpose //MatrixForm, "Očekávaný výnos = ", OptimalniPortfolio.r, " Riziko = ", Sqrt [OptimalniPortfolio. V .OptimalniPortfolio]]

objeví

Optimální portfolio:

ČEZ	0.523788
ERSTE BANK	0.151766
ČESKÝ TELECOM	0.0967847
KOMERČNÍ BANKA	−0.621273
UNIPETROL	0.260128
PHILIP MORRIS ČR	−0.054671
ZENTIVA	0.374391
ORCO PROPERTY GR	0.133806
CENT EURO MEDIA	0.135281

Očekávaný výnos = 0.004
Riziko = 0.015912

Stejný výsledek nám dá funkce *Portfolio*, do které jako proměnné zadáme řetězec s úplnou cestou a názvem souboru s maticí výnosů a požadovaný výnos.

Portfolio[soubor_ , μ_]:=
Module[{data, r, V, n, ones, X, λ1, λ2, lagr, der, rov, OP},
Needs["Statistics 'MultiDescriptiveStatistics' "];
data = Import[soubor, "Table"];
r = Mean[data];
V = CovarianceMatrix[data];
n = Length [r];
ones = Table [1,{n}];
X = Table [Unique[ ],{n}];
lagr = $\frac{1}{2}$ X.V.X + λ1*(1- ones. X)+λ2*(μ - r.X);
der = D[lagr,{{X,λ1,λ2}//Flatten}]==0;

rov = Thread[der];
OP = (X/.Solve[rov,X,λ1,λ2}//Flatten])//Flatten;
Print["Optimální portfolio : ",{jmena,OptimalniPortfolio} //Transpose //MatrixForm, "Očekávaný výnos = ", {OP. r}, "Riziko = ", Sqrt[OP.V.OP]]]

2. Minimalizace užitím zabudované funkce NMinimize. Předpokládáme, že prodeje nakrátko jsou zakázány. Funkce NMinimize minimalizuje přímo zadanou funkci $\frac{1}{2}\,\boldsymbol{x}^T\boldsymbol{V}\boldsymbol{x}$ s ohledem na podmínky $\mathbf{1}^T\boldsymbol{x}$=1, $\boldsymbol{r}^T\boldsymbol{x}$=$\mu$ a $x_i \geq 0$ pro i=1,...,n. Do funkce *PortfolioNMin* zadáme řetězec s úplnou cestou a názvem souboru s maticí výnosů jako první proměnnou a požadovaný výnos jako druhou.

PortfolioNMin[misto_ , μ_] :=
Module[{X, n, f, OP, data, r, V, podm, vahy, ciste, riz},
Needs["Statistics 'MultiDescriptiveStatistics' "];
data = Import[misto, "Table"];
r = Mean[data];
V = CovarianceMatrix[data];
n = Length[r];
X = Table[Unique[ ], { n}];
f = $\frac{1}{2}$ X . V . X;
podm = Total[X] == 1 && X . r == μ && Apply[And, Thread[X≥0]];
OP = (NMinimize[{ f, podm }, X]) //Flatten;
vahy = X /. Drop[OP // Flatten, 1];
ciste = Map[Chop[#, 10^{-6}] &, vahy];
Print ["Optimální portfolio: ", ({jmena, ciste}//Transpose)// MatrixForm,
"Očekávaný výnos = ", ciste . r , "Riziko = ", Sqrt[ciste.V.ciste]]]

Funkce *PortfolioNMin* provede následující operace:
Funkce Module[{proměnné}, výraz] zavede proměnné použité ve výrazu jako lokální v průběhu funkce *PortfolioNMin*. Po jejím skončení v nich opět bude uložena hodnota, kterou měly před zavoláním funkce v globálním kontextu. V dalších krocích proběhne natažení dat z uložené databáze, vypočtení vektoru očekávaných výnosů a kovarianční matice. Do proměnné *podm* se uloží podmínky na velikost disponibilních zdrojů výše jedna, výši požadovaného výnosu a nezápornost složek vektoru X. Po numerickém výpočtu se ještě odstraní zanedbatelně malé hodnoty výsledku. Výsledek se zobrazí stejně jako u funkce *Portfolio*.

Konkrétní výsledky pro zadaná data jsou:

$$\text{Optimální portfolio:} \quad \begin{pmatrix} \text{ČEZ} & 0.580866 \\ \text{ERSTE BANK} & 0 \\ \text{ČESKÝ TELECOM} & 0 \\ \text{KOMERČNÍ BANKA} & 0 \\ \text{UNIPETROL} & 0.419134 \\ \text{PHILIP MORRIS ČR} & 0 \\ \text{ZENTIVA} & 0 \\ \text{ORCO PROPERTY GR} & 0 \\ \text{CENT EURO MEDIA} & 0 \end{pmatrix}$$

Očekávaný výnos = 0.004
Riziko = 0.0195202

$$\text{Optimální portfolio:} \quad \begin{pmatrix} \text{ČEZ} & 0.379376 \\ \text{ERSTE BANK} & 0.014892 \\ \text{ČESKÝ TELECOM} & 0 \\ \text{KOMERČNÍ BANKA} & 0 \\ \text{UNIPETROL} & 0.180726 \\ \text{PHILIP MORRIS ČR} & 0 \\ \text{ZENTIVA} & 0.316392 \\ \text{ORCO PROPERTY GR} & 0 \\ \text{CENT EURO MEDIA} & 0.108614 \end{pmatrix}$$

Očekávaný výnos = 0.003
Riziko = 0.0134527

$$\text{Optimální portfolio:} \quad \begin{pmatrix} \text{ČEZ} & 0.166539 \\ \text{ERSTE BANK} & 0.149207 \\ \text{ČESKÝ TELECOM} & 0.190854 \\ \text{KOMERČNÍ BANKA} & 0 \\ \text{UNIPETROL} & 0.0697307 \\ \text{PHILIP MORRIS ČR} & 0 \\ \text{ZENTIVA} & 0.218779 \\ \text{ORCO PROPERTY GR} & 0.047373 \\ \text{CENT EURO MEDIA} & 0.157517 \end{pmatrix}$$

Očekávaný výnos = 0.002
Riziko = 0.00911854

Optimální portfolio:

ČEZ	0
ERSTE BANK	0.275107
ČESKÝ TELECOM	0.372497
KOMERČNÍ BANKA	0.0287675
UNIPETROL	0
PHILIP MORRIS ČR	0.12474
ZENTIVA	0.0120844
ORCO PROPERTY GR	0.027232
CENT EURO MEDIA	0.159571

Očekávaný výnos = 0.001
Riziko = 0.00721113

Pro každou volbu metody vycházejí výsledky stejně. Rozdíly jsou pouze v časech výpočtu.

Differential Evolution	0.731 sec
Nelder-Mead	0.15 sec
Random Search	0.831 sec
Simulated Annealing	0.17 sec

3.3 Maximalizace Sharpeova poměru

Maximalizaci Sharpeova poměru provedeme hledáním maxima funkce $\frac{\boldsymbol{r}^T\boldsymbol{x}}{\sqrt{\boldsymbol{x}^T\boldsymbol{V}\boldsymbol{x}}}$ za podmínky $\mathbf{1}^T\boldsymbol{x}$=1.

1. Maximalizace čtverce Sharpeova poměru. Protože nejde explicitně maximalizovat Sharpeův poměr metodou Lagrangeových multiplikátorů, maximalizujeme jeho čtverec $\frac{(\boldsymbol{r}^T\boldsymbol{x})^2}{\boldsymbol{x}^T\boldsymbol{V}\boldsymbol{x}}$. Výsledkem může být portfolio se záporným očekávaným výnosem, tj. ztrátou, nicméně na eficientním trhu se s touto situací setkáme vzácně. Předpokládáme povolení prodejů nakrátko a neexistenci bezrizikových aktiv. Protože $\boldsymbol{V}$ je pozitivně definitní, existuje symetrická matice druhých odmocnin $\boldsymbol{V}^{1/2}$. Ze Schwarzovy nerovnosti

$$(\boldsymbol{r}^T\boldsymbol{V}^{-1/2}\boldsymbol{V}^{1/2}\boldsymbol{x})^2 \leq (\boldsymbol{r}^T\boldsymbol{V}^{-1}\boldsymbol{r})(\boldsymbol{x}^T\boldsymbol{V}\boldsymbol{x})$$

vyplývá, že maximum čtverce Sharpeova poměru je $\boldsymbol{r}^T\boldsymbol{V}^{-1}\boldsymbol{r}$ a rovnost platí pouze pro $\boldsymbol{x}$= c $\boldsymbol{V}^{-1}\boldsymbol{r}$ pro libovolnou konstantu c. Proto budeme řešit tuto funkci s podmínkou, že $\boldsymbol{x}$ má být portfolio.

Xopt = X /. Solve[X == c Inverse[V].r, 1 == X.ones, X, c // Flatten] // Flatten;

nám dá list řešení $x_1, \ldots, x_n$ uložený v proměnné Xopt. Konkrétní výsledek ze souboru data, který zobrazíme příkazem

Print["Optimální portfolio: ", jmena, Xopt // Transpose // MatrixForm, " Očekávaný výnos = ", Xopt.r, " Riziko = ", Sqrt[Xopt.V.Xopt], " Sharpeův poměr = ", (Xopt.r)/(Sqrt[Xopt.V.Xopt])]

je

$$\text{Optimální portfolio:} \quad \begin{pmatrix} \text{ČEZ} & 0.434885 \\ \text{ERSTE BANK} & 0.166459 \\ \text{ČESKÝ TELECOM} & 0.137031 \\ \text{KOMERČNÍ BANKA} & -0.52005 \\ \text{UNIPETROL} & 0.213249 \\ \text{PHILIP MORRIS ČR} & -0.0320638 \\ \text{ZENTIVA} & 0.33177 \\ \text{ORCO PROPERTY GR} & 0.128459 \\ \text{CENT EURO MEDIA} & 0.140262 \end{pmatrix}$$

Očekávaný výnos = 0.00352082
Riziko = 0.0139726
Sharpeův poměr = 0.251981

2. Maximalizace Sharpeova poměru pomocí zabudované funkce NMaximize. Na rozdíl od prvního řešení není při použití funkce NMaximize potřeba maximalizovat čtverec Sharpeova poměru, ale funkce NMaximize maximalizuje přímo Sharpeův poměr $\frac{\boldsymbol{r}^T\boldsymbol{x}}{\sqrt{\boldsymbol{x}^T\boldsymbol{V}\boldsymbol{x}}}$ za podmínky $\mathbf{1}^T\boldsymbol{x}$=1. Zadáním funkce

NMaximize[SharpRatio, Total[X] == 1, X]

si ověříme, že výpočet pomocí funkce NMaximize při povolení prodejů prodejů nakrátko a neexistenci bezrizikových aktiv dává shodný výsledek jako výpočet přes čtverec Sharpeova poměru.

Při zákazu prodejů nakrátko přidáme podmínku $x_i \geq 0$ pro i=1,...,n. Pro získání výsledku stačí do funkce *PortfolioNMax* zadat úplnou cestu a název souboru s maticí výnosů.

PortfolioNMax[misto_] := Module[{X, n, SharpRatio, OP, data, r, V, podm, vahy, ciste},
Needs["Statistics` MultiDescriptiveStatistics"];
data = Import[misto, "Table"];
r = Mean[data];
V = CovarianceMatrix[data];
n = Length[r];
X = Table[Unique[], n];
SharpRatio = $\frac{r^T x}{\sqrt{x^T V x}}$;
podm = Total[X] == 1 && Apply[And, Thread[X ≥ 0]];
OP = (NMaximize[{SharpRatio, podm}, X]) // Flatten;
vahy = X /. Drop[OP // Flatten, 1];
ciste = Map[Chop[#, 10^{-6}] &, vahy];
Print["Optimální portfolio: ", {jmena, ciste} // Transpose // MatrixForm,
"Očekávaný výnos = ", ciste . r,
" Riziko = ", Sqrt[ciste . V . ciste],
" Sharpeův poměr = ",(ciste . r)/(Sqrt[ciste . V . ciste])]]

Funkce PortfolioNMax pracuje podobně jako funkce PortfolioNMin. Rozdíl je pochopitelně v optimalizované funkci a jako důsledek chybějící podmínka na požadovaný očekávaný výnos.

Konkrétní výsledek pro zadaná data je

Optimální portfolio:

ČEZ	0.284556
ERSTE BANK	0.082379
ČESKÝ TELECOM	0.0794738
KOMERČNÍ BANKA	0
UNIPETROL	0.132291
PHILIP MORRIS ČR	0
ZENTIVA	0.283736
ORCO PROPERTY GR	0.00419616
CENT EURO MEDIA	0.133368

Očekávaný výnos = 0.00255163
Riziko = 0.0113521
Sharpeův poměr = 0.22477

Literatura

[1] Dupačová J., Hurt J., Štěpán J.: *Stochastic Modeling in Economocs and Finance*. Kluwer Academic Publishers. Dordrecht 2002.

[2] Wolfram S.: *Mathematica Book*. Wolfram Media. Champaign (IL), 2003.

[3] Lagarias J. C., Reeds J. A., Wright M. H., Wright P. E.: *Convergence properties of the Nelder-Mead simplex method in low dimensions*. http://www.siam.org/journals/siopt/9-1/30347.html, 1998.