



DOI: <https://doi.org/10.15407/eip2018.01.098>

УДК 336.7:519.688

JEL: C38, C88, G17

Чумаков А.Г., канд. фіз.-мат. наук

ORCID 0000-0001-6914-0392

e-mail: gefest@ua.fm

ВИКОРИСТАННЯ ФІЛЬТРІВ КАРУНЕНА – ЛОЕВА ДЛЯ ПРОГНОЗУВАННЯ ПОВЕДІНКИ ВАЛЮТНОГО РИНКУ

Одним із завдань трейдингу є аналіз часової поведінки ціни, в результаті якого можна дати рекомендацію трейдеру "увійти" в угоду. В роботі описано реалізацію зазначеної підказки для популярної платформи NinjaTrader. Для прогнозування використовується логіт-модель із поданням даних у базисі Карунена – Лоева. Описано оригінальну високоефективну процедуру синтезу фільтрів Карунена – Лоева для статистичних ансамблів, що характеризуються групуванням своїх членів поблизу обмеженої (до 103) кількості типових представників.

К л ю ч о в і с л о в а : трейдинг, прогнозування, базис Карунена – Лоева.

Вступ

Існує багато фінансових інструментів, які використовуються при трейдингу – діяльності на валютному ринку з купівлі-продажу. Об'єктом трейдингу зазвичай є пара, причому, залежно від майданчика здійснення трейдингу, це можуть виявитися як валюти, так і цінні папери, зобов'язання або натуральні товари (зерно, газ тощо). На поведінку ціни такої пари в часі впливають як макроекономічні чинники (політика, природні явища, зміни законодавства), так і активність учасників ринку [1]. У цьому дослідженні як приклад використовується пара долар США – британський фунт стерлінгів.

Трейдер у онлайн-режимі отримує від платформи масив часових відліків цін та обсягів угод за певний останній період часу у вигляді графіка та/або масивів даних [1–4]. Одиницю часу можна обирати, зазвичай ідеться про хвилину. Завданням трейдера є спрогнозувати подальшу поведінку ціни і прийняти рішення. Прогноз полягає у висновку: найближчим часом ціна інструменту (в нашому прикладі – фунта) зростатиме, спадатиме чи змінюватиметься несуттєво. Рішення полягає в укладанні угоди з купівлі чи продажу фунта або в утриманні від угоди. Трейдер-людина покладається на власний досвід та підказки програм-радників, трейдер-робот, уповноважений укладати угоди, приймає рішення виключно на ґрунті рекомендацій програм. Правильне рішення зумовлює отримання прибутку від угоди. Склад учасників та налаштування їх програм постійно змінюються. Зазвичай платформи також дають можливість отримати масиви часових відліків за минулі періоди, коли, на відміну від онлайн-режиму, вже відома подальша поведінка ціни й обсягів, і використати ці масиви для навчання й оцінки ефективності програм-радників, що й використано у нашому дослідженні.

© Чумаков А.Г., 2018



Найпростішим засобом аналізу/прогнозу поведінки ціни традиційно вважається технічний, зокрема графічний аналіз [5–7]. Однак сучасні роботи-трейдери [7–15] використовують досконаліші засоби аналізу. Хоча застосовані для прогнозування алгоритми і програми є суворю комерційною таємницею, відомо про широке застосування нейронних мереж [9–15]. Проблемою є той факт, що прогнозування з використанням нейронних мереж, хоча і є надзвичайно ресурсномістким рішенням, не дає достатньої надійності прогнозу. Тому актуальним залишається завдання пошуку алгоритмів для програм-радіників, що забезпечать досить надійний прогноз зі збереженням прийнятної для онлайн-режиму кількості обчислень.

У своєму дослідженні ми поставили мету виконати необхідні аналіз і прогноз у два етапи, з попереднім статистичним стисненням інформації. Такий підхід добре зарекомендував себе в розв'язанні задач аналізу зображень [16, 17].

Вхідні дані для навчання

Трейдингова платформа NinjaTrader [4] дозволяє експортувати частину часового ряду, що описує поведінку конкретного інструменту. Отримані звіди дані за січень 2013 р. були збережені на сервері як структуровані набори даних (див. рис. 1а). Крім часових відліків ціни угоди та іншої службової інформації, в наборі присутні відліки обсягу угоди (див. рис. 1б) і результат трейдингу зі знаком. Зрозуміло, що остання інформація доступна тільки для серій даних з "минулого" і може бути використана лише при навчанні програми. Обсяги угод зберігаються у продиференційованому поданні. Інформація про набори даних для зручності аналізу і перегляду зберігається в базі даних (див. рис. 1с) під керуванням СУБД SQLite [18].

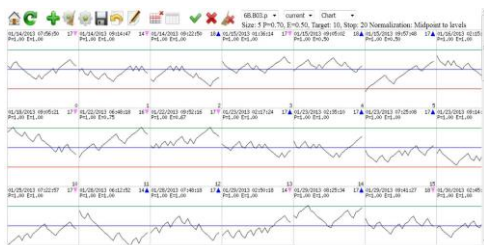


Рис. 1а. Вхідні набори даних для навчання. Часові відліки ціни угоди¹

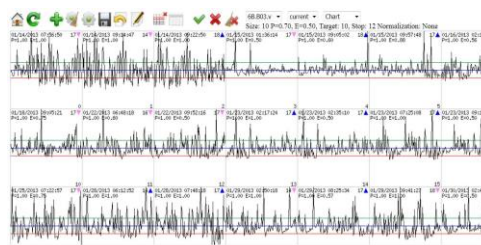


Рис. 1б. Вхідні набори даних для навчання. Часові відліки обсягу угоди

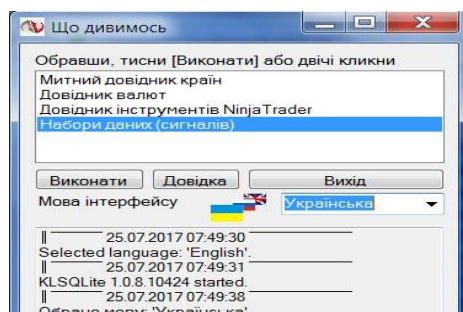


Рис. 1с. Меню обрання даних для перегляду з БД

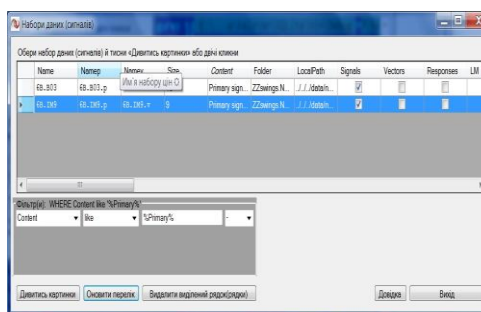


Рис. 1д. Інформація про вхідні набори даних у БД

¹ Тут і далі в цій роботі джерелом є авторські розрахунки.

Для пошуку певного набору досить накласти фільтр з вимогою показати тільки набори вхідних даних і натиснути кнопку перегляду (див. рис. 1d). Як мову і середовище програмування були використані C# та Microsoft Visual Studio 2013 [19], оскільки NinjaTrader передбачає взаємодію саме з цими інструментами.

Побудова базису власних векторів і стиснення даних

Обсяг вхідних даних у наборі занадто великий для аналізу і, тим більше, прогнозування в реальному часі. Очевидно, набір містить випадкову інформацію, яку необхідно відсіяти, і повторювану, яку необхідно зберегти і ранжувати в порядку значимості. В таких умовах першим етапом розпізнавання/передбачення є перехід від спостережуваних величин (у нашому випадку – від часових відліків цін і обсягів угод) до значень ознак, уже після якого проводиться одна або кілька процедур аналізу та прогнозу. Означений перехід повинен забезпечити зниження розмірності вектора, яким поданий сигнал, при збереженні суттєвої для його аналізу інформації [20–22]. Одним із найпоширеніших прийомів такого роду є перехід до величин відгуків на заздалегідь розраховані фільтри, тобто до скалярних добутків оброблюваного сигналу на спеціально сконструйовані еталонні сигнали. Математично ця процедура описується проектуванням простору всіх сигналів $\{u_{k,i}\}$ на простір фільтрованих сигналів $\{v_{k,l}\}$ меншої розмірності з переходом до подання сигналів по базису, в якому базисними векторами є ці еталонні сигнали $f_{l,i}$:

$$v_{k,l} = \sum_{i=1}^N (f_{l,i} u_{k,i}), \quad (1)$$

де: i, l – радіус-вектори у просторі, відповідно, спостережуваних величин і ознак;

i – в нашому випадку є номером часового відліку в сигналі, що містить дані про ціну (а також, можливо, обсяг) угод;

l – номер фільтра (ознаки);

k – номер сигналу.

Існує багато способів конструювання набору фільтрів $\{f_{l,i}\}$. Вибір серед них у кожному конкретному випадку визначається комплексом вимог до властивостей одержуваних фільтрів, можливостями їх обчислення, а також прийнятним способом реалізації операції проектування (1). З точки зору максимальної інформативності ознак при мінімумі їх кількості оптимальними є фільтри Карунена – Лоева $f_{l,i}^0$ [21, с. 234–241], що є власними векторами матриці C_{ij}^0 кореляції членів статистичного ансамблю сигналів:

$$\sum_{j=1}^N (C_{ij}^0 f_{l,j}^0) = \lambda_l f_{l,i}^0, \quad (2)$$

де

$$C_{ij}^0 = \text{ave}_k (u_{k,i} u_{k,j}). \quad (3)$$

Альтернативною популярною назвою для аналізу сигналів у базисі Карунена – Лоева є аналіз за методом головних компонент.

Вектори (фільтри) $f_{l,j}^0$ мають три такі суттєві властивості:

- 1) повноту в просторі сигналів, що становлять статистичний ансамбль;
- 2) ортонормованість;



3) некорельованість коефіцієнтів C_{kl}^0 подання по них ансамбля $\{u_{k,i}^0\}$:

$$C_{kl}^0 = \text{Diag}_{kl} \lambda_l, \quad (4)$$

де

$$C_{lm}^0 = \text{ave}_k [\sum_{i=1}^N (u_{k,i} f_{l,i}^0) \sum_{j=1}^N (u_{k,j} f_{m,j}^0)] . \quad (5)$$

Остання властивість є наслідком визначення (2) – (3) і зумовлює мінімальність помилки заміни сигналу $u_{k,i}$ на його усічене подання по базису Карунена – Лоева, порівняно з усіченим поданням по будь-якому іншому базису. Ця властивість і є підставою для обрання відгуків на перші – найзначущі з фільтрів Карунена – Лоева, як найінформативніших ознак для процедур аналізу і прогнозування.

Особливо ефективно відбувається стиснення корисної інформації при застосуванні перетворення (1) з фільтрами Карунена – Лоева як $f_{l,i}$ у тому випадку, коли $u_{k,i}$ є сигналами високої інформаційної ємності, тобто межа підсумовування по дискретному i сягає $10^3 \div 10^8$. При цьому кількість ознак, що представляють сигнал, стає досить малою для виконання над ним різних операцій в оперативній пам'яті комп'ютера за прийнятний час. Однак саме в цьому випадку унеможлилювався сам синтез фільтрів Карунена – Лоева, наприклад, за допомогою стандартної процедури сингулярного розкладання матриці SVD [22, 23], адже для цього треба було би розв'язувати спектральну задачу (2) для матриці C_{ij}^0 (3) загального вигляду з числом рядків $N=10^3 \div 10^8$, що є непосильною для більшості сучасних комп'ютерів. Нижче буде розглянута процедура [16–17] синтезу фільтрів Карунена – Лоева, що дозволяє обійти цю обчислювальну складність для конкретного, досить загального, типу ансамблю $\{u_{k,i}\}$.

У багатьох ансамблях можна виділити підансамбль, що містить кінцеве, невелике число найтипівіших представників кожного класу, усереднення за яким дозволяє отримати незміщену конзистентну оцінку матриці C_{ij}^0 . Більшість власних чисел матриці C_{ij}^0 такого ансамблю близькі до нуля. Стійкість рішень задачі (2), що відповідають великим l , зумовлює те, що фільтри, побудовані за таким підансамблем, повинні бути близькі за властивостями до фільтрів, побудованих за повним ансамблем. Оскільки $f_{l,i}^0$ є власними векторами ермітової матриці C_{ij}^0 (3), їх можна уточнювати методом теорії збурень, добре розробленим у рамках квантової механіки.

Нехай маємо ансамбль сигналів $\{u_{k,i}\}$, кількість членів якого M значно менша за розмірність N самого сигналу. Матриця C_{ij}^0 такого ансамблю матиме щонайменше $N-M$ нульових власних чисел. Тоді можна побудувати базис Карунена – Лоева, не розв'язуючи одразу задачі отримання властивостей I–III) шляхом обчислення матриці кореляції C_{ij}^0 та розв'язання для неї спектральної задачі (4), а послідовно конструюючи базиси I–III, що мають спочатку тільки властивість I), потім – I) та II), і, нарешті, I)–III), на кожному етапі використовуючи вже набуті корисні властивості базису, що будується. Для цього (етап I) як вектор базису I) оберемо власне сигнали $u_{k,i}$, користуючись повнотою їх самоподання:

$$f_{k,i}^I \equiv u_{k,i} . \quad (6)$$

Конструювання базису I полягає в обчисленні елементів його метричного тензору:

$$g_{kl} = \sum_{i=1}^N (f_{k,i}^I f_{l,i}^I) . \quad (7)$$

Надалі обраховується матриця C_{ls}^I кореляції коефіцієнтів подання ансамблю $\{u_{l,i}\}$ по базису I шляхом підстановки $f_{k,i}^I$ до (5). При цьому можна, завдяки (6), скористатися вже обчисленими скалярними добутками (7). В результаті для C_{ls}^I отримаємо:

$$C_{ls}^I = ave_k (g_{kl} g_{ks}) . \quad (8)$$

Етап II полягає в ортогоналізації базису I із застосуванням процесу Грама – Шміда. Його результатом є обчислення, по-перше, матриці B_{nl} переходу від базису I до ортогоналізованого базису II, і, по-друге, матриці C_{nm}^{II} кореляції коефіцієнтів подання ансамблю $\{u_{l,i}\}$ у ортонормованому базисі II. Оскільки перехід від базису I до базису II відбувається шляхом домноження на B_{kl} , матрицю C_{nm}^{II} можна отримати з уже відомої C_{ls}^I таким чином:

$$C_{nm}^{II} = \sum_{l=1}^M [B_{nl} \sum_{s=1}^M (C_{ls}^I B_{sm})] . \quad (9)$$

Етап III полягає у знаходженні унітарного перетворення (повороту) T_{pn} , що зведе C_{nm}^{II} до діагонального вигляду (4):

$$\sum_{n=1}^M [T_{pn} \sum_{m=1}^M (C_{nm}^{II} T_{mq})] = C_{pq}^{III} , \quad (10)$$

де C_{pq}^{III} задовольняє умові (4). Порівняно з пошуком такого перетворення для C_{ij}^0 ця операція виконується в $(N/M)^3$ раз швидше завдяки меншій розмірності C_{nm}^{II} . Знаючи матрицю цього перетворення, T_{pn} , можна подати вектори базису III через вектори базису II, а із застосуванням матриці B_{nl} – і через базис I, тобто, з урахуванням визначення (7), через вхідні сигнали. Таким чином, остаточно для базисних фільтрів Карунена – Лоева маємо вираз:

$$f_{l,i}^0 = \sum_{k=1}^M (H_{lk} u_{k,i}) , \quad (11)$$

де

$$H_{lk} = \sum_{n=1}^M (T_{ln} B_{nk}) . \quad (12)$$

Описана процедура синтезу фільтрів включає етапи з числом необхідних арифметичних операцій, пропорційним M^2N (7), (11) та M^3 (8)-(10), (12), тоді як класична процедура (2)–(3) потребує кількості операцій, пропорційної N^3 [23]. З цього випливає, що для ансамблю сигналів, для яких характерні значення $N=10^3 \div 10^8$, запропонована процедура забезпечує зменшення обчислювальних витрат на синтез фільтрів Карунена – Лоева в $10^2 \div 10^{12}$ разів і тим самим робить такий синтез практично здійсненим [16–17].

Програма аналізу даних для подальшого прогнозування (див. рис. 2а) може брати вхідні дані (й зберігати побудований базис) як з локального диску, так і з сервера (див. рис. 2б). Для аналізу можуть використовуватися як виключно часові відліки ціни угоди, так і "склеєні" набори відліків цін і обсягів угод (див. рис. 2с). Параметри для створення базису беруться з json-файлу, який можна редагувати звичайним текстовим редактором. У результаті побудови базису (див. рис. 2д) стають можливими суттєве стиснення корисної інформації (див. рис. 2е) та відсіювання випадкової.

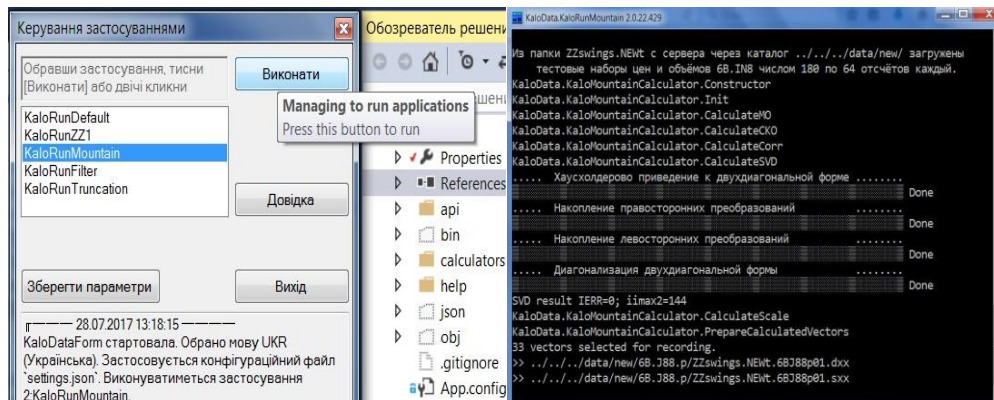


Рис. 2а. Меню запуску програми побудови бази

Рис. 2б. Побудова бази за набором відліків з серверу

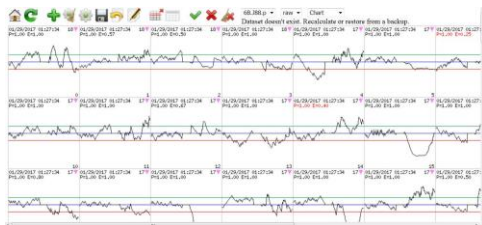


Рис. 2с. "Склесні" набори відліків цін і обсягів угод

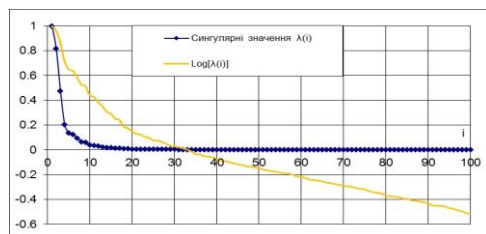


Рис. 2е. Розподіл енергії (статистичної вартості) набору в просторі головних компонент у звичайному і логарифмічному масштабах

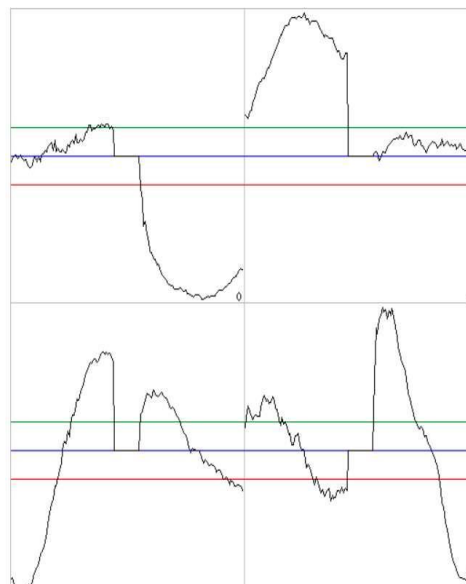


Рис. 2д. Перші декілька з побудованих власних векторів (головних компонент)

Створення і навчання моделі для прогнозування

Навчальні та іспитові набори подаються у вигляді головних компонент, тобто відгуків на побудовані вектори-фільтри. Оскільки відліки, що близькі за часом до моменту прийняття рішення, вважаються важливішими за далекі за часом, для обліку значимості відліку застосовується "вага-гірка", приклади графіків якої наведені на рис. 3. Для наборів, "склесних" з цін та обсягів, вага застосовується і для тих, і для інших відліків. За бажання можна задати дуже полого "гірку", і тоді відліки будуть враховані незалежно від їх близькості до моменту прийняття рішення.

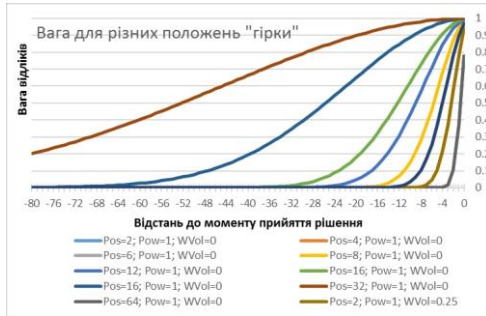


Рис. 3а. Приклади графіків "ваги-гірки" для регулювання вартості часового відліку залежно від його близькості до моменту прийняття рішення

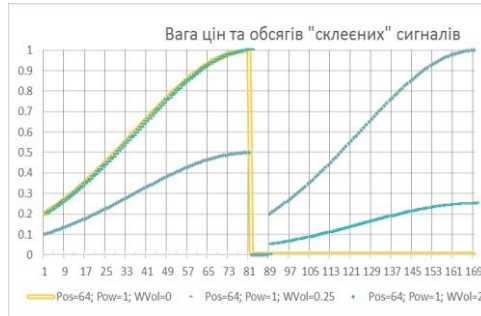


Рис. 3б. Приклади графіків "ваги-гірки" для "склеєних" сигналів

Гістограми розподілу відгуків у просторі головних компонент представлені на рис. 4, а приклади розташування наборів у просторі перших трьох векторів – на рис. 5. Завданням розпізнавання є досить впевнено спрогнозувати, чи буде в подальшому курс помітно зростати (такі набори позначені на рис. 5 значком ##), спадати (значок ==) чи змінюватися несуттєво (значки ++ та --). Видно, що чіткої сегрегації на зростаючі, спадаючі та нейтральні в рамках лінійної моделі немає. Були виконані кількісні оцінки згідно з [24], з урахуванням поправок на невеликий обсяг вибірки. Для фільтрації промахів і визначення відстаней між центрами наборів розподіл координат зростаючих, спадаючих і нейтральних у просторі приймався узагальненим нормальним,

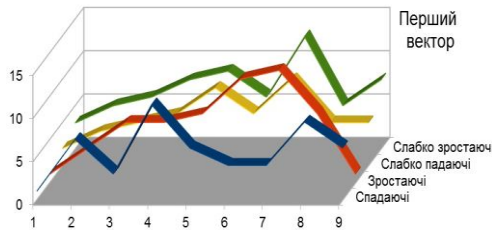


Рис. 4а. Гістограма відгуків на 1-й вектор

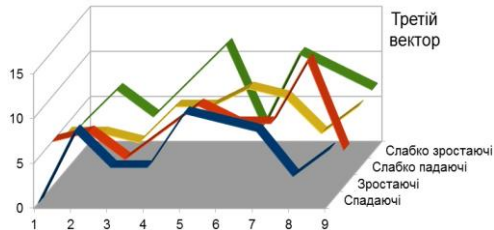


Рис. 4б. Гістограма відгуків на 3-й вектор

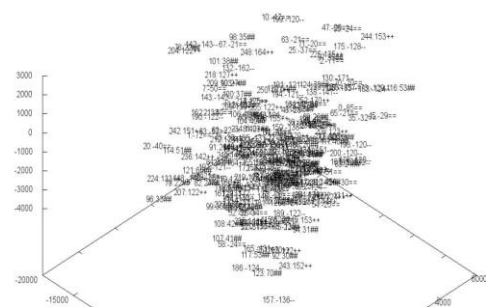


Рис. 5. Приклад розташування наборів у просторі перших трьох векторів

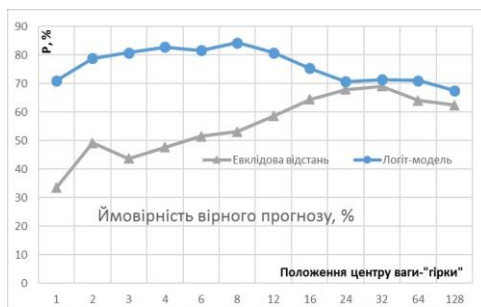


Рис. 6. Ймовірність вірного прогнозування для різного положення центру "ваги-гірки"



що узгоджується з критерієм Жака – Бера (значення критерію перебувало в межах $0.03 \div 0.5$, що відповідає ймовірності правильності гіпотези про вид розподілу не гірше 98%). На рис. 4–5 добре видно, що середньоквадратичне відхилення зростаючих, падаючих і нейтральних наборів у разі перевищує відстань між центрами цих наборів. У результаті ймовірність правильного розпізнавання не перевищує 55%, що надзвичайно мало для практичних потреб. У деяких випадках дещо поліпшив розпізнавання підрахунок відстані лише по найближчих сусідах, але незначно (до 60%).

Надалі будувалася і навчалася нелінійна логіт-модель [25]. Коефіцієнти логіт-моделі знаходяться шляхом мінімізації функції помилки на навчальній множині сигналів. Як оцінка помилки використовується крос-ентропія (плюс невеликий регуляризуючий додаток, що поліпшує збіжність). Логіт-модель є, з одного боку, окремим випадком узагальнених лінійних моделей, а з іншого – окремим випадком нейронної мережі – мережею з одним лінійним шаром і SOFTMAX-нормалізацією виходів. Перевага розпізнавання за логіт-моделлю порівняно з лінійною моделлю демонструється на рис. 6. Імпортовані з Ninja-Trader набори відліків діляться на три групи. Перша використовується для створення базису головних компонент. Друга (навчальна) – для навчання моделі, третя (випробувальна) – для оцінки якості передбачення. Таким чином забезпечується об'єктивна оцінка якості зробленого прогнозу, оскільки третя група дозволяє обрахувати ймовірність вірного прогнозування.

Для навчання моделі їй пред'являються набори трьох класів: зростаючі, спадаючі та нейтральні. Параметри задаються в json-файлі. Програма (приклад виклику – див. рис. 7, детальніший опис – див. [26]) має кілька параметрів, що налаштовуються:

- положення центру "ваги-гірки";
- крутизна "ваги-гірки";
- кількість головних компонент, що враховуються;
- ступінь урахування головних компонент залежно від величини сингулярного числа;
- відносна статистична вага відліків обсягів порівняно з відліками цін.

Ці параметри можуть бути як жорстко задані в конфігураційному json-файлі, так і знайдені шляхом підбору. У цьому ж файлі задається, де розташовані вхідні дані та результат (локальний диск або сервер), номер позиції при запуску розрахунку після переривання або збою. Серіалізовані параметри навченої моделі зберігаються разом із відгуками. Оскільки перебігом навчання модель підраховує середньоквадратичне відхилення (СКВ) оцінки ймовірності приналежності набору навчальної вибірки зазначеному класу, остаточний результат приналежності набору відліків до класу зростаючих або спадаючих у ціні приймався тільки в разі перевищення зазначеної ймовірності над ймовірностями приналежності до інших класів понад СКВ. Досягнуті результати розпізнавання, в тому числі оцінка ймовірності вірного передбачення, що підрахована за незалежною від навчальної тестовою вибіркою, записуються:

- ✓ у текстовий файл-протокол;
- ✓ у базу даних для подальшого аналізу (див. рис. 8);

- ✓ у HTML-звіт (див. рис. 9);
- ✓ виводяться на графіки (див. рис. 10).

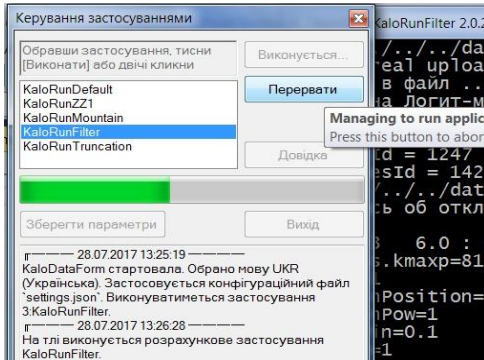


Рис. 7. Виклик програми побудови й навчання логіт-моделі

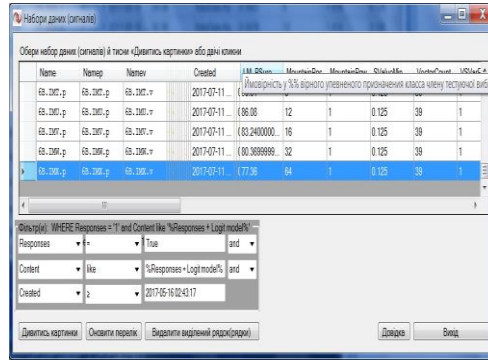


Рис. 8. Результати сеансу побудови й навчання логіт-моделі у таблиці БД

Results of the search of parameters

i	Pos	Pos	Pos	W	SValueMin	SValue	N of vectors	i	Value	Euclid	Myopia	Legiti	RMS	PSure	Line	SetName	A sign of the best	
0	1	2	0	1	0	0.125	39	0	1	0	29.3	17.97	78.91	0.2143	84.21	0	68...ZDK.p	
1	2	4	0	1	0	0.125	39	0	1	0	46.83	17.97	83.90	0.2859	77.84	1	68...ZDK.p	
2	3	6	0	1	0	0.125	39	0	1	0	54.3	15.23	83.59	0.2843	87.22	2	68...ZDK.p	
3	4	8	0	1	0	0.125	39	0	1	0	54.49	23.05	83.59	0.2807	87.95	3	68...ZDK.p	
4	5	12	0	1	0	0.125	39	0	1	0	70.7	23.44	83.16	0.279	88.29	4	68...ZDK.p	
5	6	16	0	1	0	0.125	39	0	1	0	72.66	19.14	84.77	0.2866	87.98	5	68...ZDK.p	
6	7	32	0	1	0	0.125	39	0	1	0	67.38	19.53	73.83	0.2498	80.75	6	68...ZDK.p	
7	8	64	0	1	0	0.125	39	0	1	0	68.75	18.75	67.97	0.3611	80.82	7	68...ZDK.p	
8	1	2	0	1	0	0.125	39	0	1	2	0.25	52.34	11.33	79.69	0.3089	87.69	8	68...ZDK.p
9	2	4	0	1	0	0.125	39	0	1	2	0.25	45.7	16.02	83.59	0.2815	87.44	9	68...ZDK.p
10	3	6	0	1	0	0.125	39	0	1	2	0.25	53.12	14.45	82.81	0.2807	88.29	10	68...ZDK.p
11	4	8	0	1	0	0.125	39	0	1	2	0.25	54.69	19.14	85.55	0.2763	88.34	11	68...ZDK.p
12	5	12	0	1	0	0.125	39	0	1	2	0.25	66.02	17.58	81.64	0.29	88.1	12	68...ZDK.p
13	6	16	0	1	0	0.125	39	0	1	2	0.25	67.19	21.88	78.91	0.3115	87.56	13	68...ZDK.p
14	7	32	0	1	0	0.125	39	0	1	2	0.25	71.48	21.88	69.14	0.2507	83.33	14	68...ZDK.p
15	8	64	0	1	0	0.125	39	0	1	2	0.25	71.48	19.92	69.92	0.254	81.65	15	68...ZDK.p
16	1	2	0	1	0	0.125	39	0	1	3	1	53.52	10.94	78.91	0.3169	86.81	16	68...ZDK.p
17	2	4	0	1	0	0.125	39	0	1	3	1	44.53	16.41	82.81	0.2886	87.38	17	68...ZDK.p
18	3	6	0	1	0	0.125	39	0	1	3	1	51.56	14.45	81.64	0.2881	87.74	18	68...ZDK.p
19	4	8	0	1	0	0.125	39	0	1	3	1	53.12	15.62	84.38	0.2896	88.73	19	68...ZDK.p
20	5	12	0	1	0	0.125	39	0	1	3	1	58.59	21.09	80.86	0.3044	87.37	20	68...ZDK.p
21	6	16	0	1	0	0.125	39	0	1	3	1	64.45	21.09	75.39	0.328	83.96	21	68...ZDK.p
22	7	32	0	1	0	0.125	39	0	1	3	1	71.48	18.36	71.48	0.3542	78.5	22	68...ZDK.p
23	8	64	0	1	0	0.125	39	0	1	3	1	64.06	17.19	71.09	0.3613	77.27	23	68...ZDK.p

Рис. 9. Звіт про результати сеансу побудови й навчання логіт-моделі

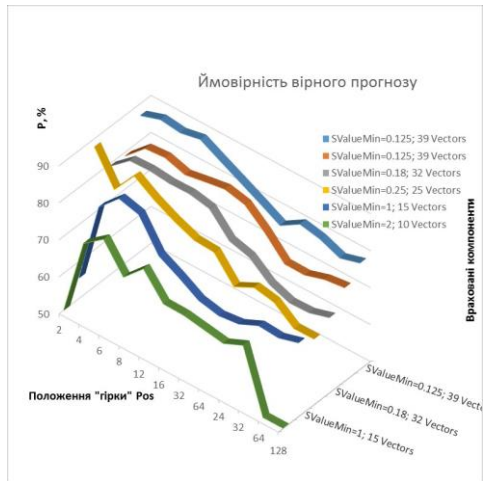


Рис. 10. Графіки ймовірності вірного прогнозування у сеансі побудови й навчання логіт-моделі

У результаті перебору параметрів, що налаштовуються, можна зробити такі висновки:

- спосіб управління "вагою-гіркою": положенням центру або крутизною – в невеликих межах принципового значення не має. Оптимальне прогнозування досягалося в межах положення центру $6 \div 16$ і крутизни 1;
- ступінь урахування головних компонент та їх кількість не вимагають точного регулювання, оскільки логіт-модель у змозі самостійно досить розумно розставити ваги пред'явленим їй ознакам-відгукам.

Для обраної пари фінансових інструментів (долар-фунт) при параметрах:

- положення центру "ваги-гірки" 8;
- крутизна "ваги-гірки" 1;



- кількість головних компонентів, що враховуються, 39;
- ступінь врахування головних компонент залежно від величини сингулярного числа 1;
- відносна статистична вага відліків обсягів порівняно з відліками цін 1;

було досягнуто ймовірність вірного прогнозування 88%, що вже прийнятно для практичних потреб.

Імплементация отриманих результатів до платформи NinjaTrading

Для вирішення означеного завдання створений програмний модуль (клас) Advisor. Крім іншого, він містить доступні для виклику ззовні (публічні) функції (методи):

- AdvisorLoad, що вантажить в оперативну пам'ять з локального каталогу або з серверу набір базисних векторів і навчену модель;
- AdvisorAnswer, що отримує масиви часових відліків цін та обсягів угод за найближчий до моменту прийняття рішень період і видає вердикт про подальшу поведінку ціни:
 - 0 – спадатиме;
 - 2 – зростатиме;
 - 1 – не впевнений;
 - <0 та діагностику – у випадку помилки.

Цей модуль може бути підключений як "радник" до програми-індикатора (для підказки трейдеру-людині) або до робота, уповноваженого самостійно реалізовувати стратегію трейдингу. Модуль реалізовує запропоновану нову методику роботи програми-радника. Таким чином реалізується перспективна можливість запропонованого інструментарію для прогнозів економічних показників (зокрема курсу валютного маркету).

Висновки. Запропонована методика аналізу та передбачення поведінки валютного маркету з побудовою логіт-моделі в поданні Карунена – Лоева дозволяє досягнути прийнятних для практичного застосування результатів прогнозування. Подальша перспектива дослідження вбачається в його імплементации до комерційного продукту автоматичного (роботизованого) трейдингу.

Список використаних джерел

1. Кияница А.С. Фундаментальный анализ финансовых рынков. Санкт-Петербург: "Питер", 2005. 288 с.
2. Петров Г., Шилов Б., Ведихин А. Форекс от первого лица . Москва: "Смартбук", 2012. 408 с.
3. Forex.ua. URL: <http://forex.ua>
4. Ninjatrader / Ninjatrader Group. URL: <http://ninjatrader.com/>
5. Графический анализ Форекс / Телетрейд Групп. URL: <https://www.teletrade.ru/novice/graficheskiy-analiz-forex>



6. Анализ графиков в техническом анализе рынка Форекс / Форекс отзывы. URL: <http://reviewforex.ru/page/analiz-grafikov>
7. Академия Форекс / bforex. URL: <https://ru.bforex.com/>
8. Сильвани А. Переиграть дилера на рынке Forex: Взгляд инсайдера. Москва: "Альпина Паблишер", 2012. 259с.
9. Бабак Геннадий. Автоматизация торговых стратегий. URL: http://profitmania.org/chto_luchshe_avtomat_ili_poluavtomaticheskie_strategii
10. Смирнов Михаил. Прогнозирование фондового рынка с использованием нейронных сетей. URL: <https://geektimes.ru/post/279170/>
11. Нейронные сети в трейдинге на Форекс / Академия Форекса. URL: <http://academy-fx.ru/article/blogi/2436-nejronnye-seti-v-trejdinge-na-foreks>
12. Кац Д.Э., МакКормик Д.Л. Энциклопедия торговых стратегий. Москва: "Альпина Паблишер", 2002. 400с.
13. Ширяев В.И. Финансовые рынки. Нейронные сети, хаос и нелинейная динамика. Москва: "Либроком", 2009. 232 с.
14. Babu A.S., Reddy S.K. Exchange Rate Forecasting using ARIMA, Neural Network and Fuzzy Neuron. *J Stock Forex Trading* 2015. Vol.4, Is. 3. P. 155. doi: <https://doi.org/10.4172/2168-9458.1000155>
15. Kumar Chandar S., Sumathi M., Sivanandam S. N. Forecasting of Foreign Currency Exchange Rate Using Neural Network. *International Journal of Engineering and Technology (IJET)*. 2015. Feb.-Mar. Vol. 7, No. 1. P. 99–108.
16. Chumakov A.G. Image processing in Karhunen-Loeve representation. *20th Annual European Meeting on Atmospheric Studies by Optical Methods: Book of abstracts*. 14–18 Sep. 1993. Apatity, Russia. P. 56.
17. Chumakov A.G., Kisil, A.V., Kovalenko, A.V. et al. Optoelectronic system of digital holographic image processing. *Proc. SPIE 2108, International Conference on Holography, Correlation Optics, and Recording Materials*. 10 December 1993. P. 332–337. doi: <https://doi.org/10.1117/12.165403>
18. SQLite. URL: <https://www.sqlite.org/>
19. Welcome to Visual Studio 2013 / Microsoft. URL: [https://msdn.microsoft.com/en-us/library/dd831853\(v=vs.120\).aspx](https://msdn.microsoft.com/en-us/library/dd831853(v=vs.120).aspx)
20. Фомин Я.А., Тарловский Г.Р. Статистическая теория распознавания образов. Москва: Радио и связь, 1986. 264 с.
21. Фукунага К. Введение в статистическую теорию распознавания образов. Москва: Наука, 1979. 368 с.
22. Айвазян С.А. и др. Прикладная статистика. Классификация и снижение размерности. Москва: "Финансы и статистика", 1989. 607 с.
23. QR and LQ decompositions / ALGLIB. URL: <http://www.alglib.net/matrixops/qr.php>
24. Булашев С.В. Статистика для трейдеров. Москва: "Компания Спутник+", 2003. 245 с.
25. Multinomial logit regression / ALGLIB. URL: <http://www.alglib.net/dataanalysis/logit.php>
26. Чумаков А.Г. Програма KaloData побудови й навчання логіт-моделі для NinjaTrader. URL: <http://gefest.ho.ua/Pgm/KLSQLite.UKR.htm>

Надійшла до редакції 08.11.2017 р.



Чумаков А.Г., канд. физ.-мат. наук

ИСПОЛЬЗОВАНИЕ ФИЛЬТРОВ КАРУНЕНА – ЛОЭВА ДЛЯ ПРОГНОЗИРОВАНИЯ ПОВЕДЕНИЯ ВАЛЮТНОГО РЫНКА

Одной из задач трейдинга является анализ временного поведения цены, в результате которого можно дать рекомендацию трейдеру "войти" в сделку. В работе представлен программный проект, реализующий указанные подсказки для популярной платформы NinjaTrader. Для прогнозирования использовалась логит-модель с представлением данных в базисе Карунена – Лоева. Описана оригинальная высокоэффективная процедура синтеза фильтров Карунена – Лоева для статистических ансамблей, характеризующихся группированием своих членов вблизи ограниченного (до 10^3) числа типичных представителей.

Ключевые слова: трейдинг, прогнозирование, базис Карунена – Лоева

A. Chumakov, PhD in Physics and Mathematics

THE USE OF KARHUNEN-LOEVE FILTERS TO PREDICT THE BEHAVIOR OF THE CURRENCY MARKET

One of the tasks of trading is to predict the behavior of the price of currency market transactions in order to timely enter into an agreement. The purpose of the article is to describe a proposed forecasting technique adapted for the implementation in the popular platform NinjaTrader. The forecast is based on the analysis of data provided in the Karhunen-Loeve representation using the logit model. The article describes an original and highly effective procedure for the synthesis of Karhunen-Loeve filters for statistical ensembles characterized by grouping its members around a limited (up to 10^3) number of typical representatives.

The proposed procedure for constructing a basis provides a $10^2 \div 10^{12}$ fold reduction in the computational cost for the synthesis of Karhunen-Loeve filters for a specified type of statistical signal ensembles, which makes such a synthesis feasible. Described are programs that implement the construction of the basis, the creation and training of a logit model and the forecasting. It is demonstrated how the application of data analysis in the constructed basis with the help of a logit model can provide an acceptable level of reliability for a forecast intended for practical application.

Key words: trading, prediction, Karhunen-Loeve basis

References

1. Kijanica, A.S. (2005). Fundamental analysis of financial markets. Russia, Sankt-Peterburg: Piter [in Russian].
2. Petrov, G., Shilov, B., Vedikhin, A. (2012). Forex from the first person. Moscow: Smartbook [in Russian].
3. Forex.ua (2017). Retrieved from <http://forex.ua> [in Ukrainian].
4. Ninjatrader (2017). Retrieved from <http://ninjatrader.com>



5. Graphical analysis of Forex (2016). Retrieved from <https://www.teletrade.ru/novice/graficheskij-analiz-forex> [in Russian].
6. Analysis of charts in the technical analysis of the Forex market (2013). Retrieved from <http://reviewforex.ru/page/analiz-grafikov> [in Russian].
7. Academy of Forex (2010). Retrieved from <https://www.bforex.com/>
8. Silvani, A. (2012). Replay the dealer in the Forex market: Insider's view. Moscow: Alpina Publisher [in Russian].
9. Automating trading strategies (2015). Retrieved from http://profitmania.org/chto_luchshe_avtomat_ili_poluavtomaticheskie_strategii [in Russian].
10. Forecasting the stock market using neural networks (2016). Retrieved from <https://geektimes.ru/post/279170/> [in Russian].
11. Neural networks in Forex trading (2017). Retrieved from <http://academyfx.ru/article/blogi/2436-nejronnye-seti-v-trejdinje-na-foreks> [in Russian].
12. Katz, J.O., McCormick, D.L. (2002). The Encyclopedia of Trading Strategies. Moscow: Alpina Publisher [in Russian].
13. Shyryaev, V.I. (2009). Financial markets. Neural networks, chaos and nonlinear dynamics. Moscow: Librocom [in Russian].
14. Babu, A.S., Reddy, S.K. (2015). Exchange Rate Forecasting using ARIMA, Neural Network and Fuzzy Neuron. *J Stock Forex Trading*, 4: 3, 155. doi: <https://doi.org/10.4172/2168-9458.1000155>
15. Kumar, Chandar S., Sumathi, M., Sivanandam, S.N. (2015, Feb.-Mar.). Forecasting of Foreign Currency Exchange Rate Using Neural Network. *International Journal of Engineering and Technology (IJET)*, 7: 1, 99-108.
16. Chumakov, A.G. Image processing in Karhunen-Loeve representation (1993, Sep. 14-18). *20th Annual European Meeting on Atmospheric Studies by Optical Methods: Book of abstracts*. Apatity, Russia [in Russian].
17. Chumakov, A.G., Kisil, A.V., Kovalenko, A.V. et al. (1993, December 10). Optoelectronic system of digital holographic image processing. *Proc. SPIE 2108, International Conference on Holography, Correlation Optics, and Recording Materials*, 332-337. doi: <https://doi.org/10.1117/12.165403>
18. SQLite (2017). Retrieved from <https://www.sqlite.org>
19. Welcome to Visual Studio 2013 (2017). Retrieved from [https://msdn.microsoft.com/en-us/library/dd831853\(v=vs.120\).aspx](https://msdn.microsoft.com/en-us/library/dd831853(v=vs.120).aspx)
20. Fomin, Ja.A., Tarlovskij, G.R. (1986). Statistical Theory of Pattern Recognition. Moscow: Radio i svjaz' [in Russian].
21. Fukunaga, K. (1979). Introduction to the statistical theory of pattern recognition. Moscow: Nauka [in Russian].
22. Ajvazjan, S.A. et al. (1989). Applied statistics. Classification and dimension reduction. Moscow: Finansy i statistika [in Russian].
23. ALGLIB (2017). QR and LQ decompositions. Retrieved from <http://www.alglib.net>
24. Bulashev, S.V. (2003). Statistics for traders. Moscow: Kompanija Sputnik+ [in Russian].
25. ALGLIB (2017). Multinomial logit regression. Retrieved from <http://www.alglib.net/dataanalysis/logit.php>
26. Chumakov, A.G. (2017). Program KaloData of construction and training of the logit model for NinjaTrader. Retrieved from <http://gefest.ho.ua/Pgm/KLSQLite.ENG.htm>