Прогнозирование направления движения цены биржевых инструментов по новостному потоку.

Ахияров В., Борисов А., Говоров И., Родионов В.
akhiarov.va@phystech.edu, borisov.as@phystech.edu, govorov.is@phystech.edu, rodionov.vo@phystech.edu
МФТИ (ГУ)

Аннотация: В работе рассматривается задача классификации направления движения временных рядов. Классификация производится с помощью анализа признаков из отчётов 8-K, которые компании обязаны заполнять при значительных событиях, таких как банкротство, выбор совета директоров и пр. Рассматривается несколько моделей классификации. В одних используются только признаки из отчётов, 1-граммы которых встречающиеся более 10 раз. В других к предыдущему этапу применяется неотрицательная матричная факторизация (NMF). И в последней, ансамбле, объединяются предыдущие подходы путём голосования большинства. В качестве прикладной задачи рассматривается задача распознавания направления движения акций по новостям, выраженных отчётами 8-K. Модели классификации, исследованные в этой работе, сравнивается в точности и статистической значимости с простыми моделями, использующими только прогнозируемый показатель доход на акцию или использующую только финансовые показатели.

Ключевые слова: *метрическая классификация, анализ текстов, классификация временных рядов, новостной поток*

Введение

Прогнозирование направления движения цены биржевых инструментов по новостному потоку. Мотивируемое тем, что флуктуации цен на бирже, сильно зависящие от политической, географической и т.д. обстановок, интересные не только при скальпинге. Для среднесрочных торгов и инвестиций такие данные так же имеют большую роль, позволяя корректировать вложения. Как правило, крупные изменения в политике, природные катаклизмы и все события которые именяют распределение цен котировок, освещаются в прессе.

Исследование строится вокруг постоянных изменений цен биржевых котировок, новостей, и алгоритма NMF вектора.

Требуется на основе большого количество новой информации (предоставляемой в разрозненном текстовом виде) касающейся компаний, перечисленных на фондовом рынке, предсказать повышение, понижение либо стабилизацию цен на акции, ценные бумаги и т.д. Необходимо разработать модель, которая также учитывает недавнее движение акций, и так называемую «неожиданную прибыль» (отчет о прибылях и убытках компании, значительно отличающийся (в положительном или отрицательном направлении) от ожиданий аналитиков (согласованного прогноза)

Методы исследования. В работе приведены другие, которые как улучшают уже существующие, так и вводят новые методы обработки естественного языка. Так в Xie et al. (2013) вводится дерево представлений об информации в новостях, в Bollen et al. (2010) использованы данные из Twitter'a. Bar-Haim et al. (2011) распознают лучших экспертов-инвесторов, а Leinweber and Sisk (2011) исследуют влияние новостей и време-

ни усвоения новостей в событийной торговле. В Kogan et al. (2009) приводится предсказание риска по финансовым отчётам и в Engelberg (2008) - закономерность о том, что лингвистическая информация (возможно из-за когнитивной нагрузки при обработке) имеет более долгосрочную предсказуемость цен, нежели количественная информация.

Решаемая в данной работе задача. Построить и исследовать модель прогнозирования направления движения цены. Задано множество новостей S и множество временных меток T, соответствующих времени публикации новостей из S. 2. Временной ряд P, соответствующий значению цены биржевого инструмента, и временной ряд V, соответствующий объему продаж по данному инструменту, за период времени T'. 3. Множество T является подмножеством периода времени T'. 4. Временные отрезки $w = [w_0, w_1], l = [l_0, l_1], d = [d_0, d_1]$, где $w_0 < w_1 = l_0 < l_1 = d_0 < d_1$. Требуется спрогнозировать направление движения цены биржевого инструмента в момент времени $t = d_0$ по новостям, вышедшим в период w.

Предлагаемое решение. 8K - отчеты компаний об их внутренних событиях. Данная отчетность выходит строго в период между закрытием торгов в один день и их открытием на следующий день. Из отчета 8K убираются все HTML-теги, таблицы и прочее. Используется метод NMF вектора. Вычитается из цен сегодняшнего открытия торгов вчерашние цены закрытия торгов с поправкой на индекс. Берется текст отчета 8K и на выходе нейронной сети функция, принимающая три значения :

1. UP – цена открытия следующего дня больше на 1+ % от предыдущего дня – «изменение индекса»
2. DOWN – цена открытия следующего дня меньше на 1+ % от предыдущего дня – «изменение индекса»
3. STAY – цена открытия следующего дня в пределах ± 1 % от предыдущего дня – «изменение индекса»

Плюсы метода: Большой объем данных Он более доступен небольшим инвесторам, чем real-time trading tools, которыми пользуются большие трейдинговые компании Он показывает accuracy на 10% больше, baseline, который использует только финансовые фичи (см. статью [3]) смотрят «изменение цены» - «изменение индекса» => чистое влияние все дивидендные гэпы убирали.

Минусы: Исследование проведено на рынке США, где отчеты выходят не в торговое время => вся информация отражается мгновенно в цене акции от открытии результаты не имеют значения на практике => невозможно извлечь финансовую прибыль Метод не улавливает такие эффекты, как: slippage, transaction costs, borrowing costs

Эксперимент будет проведен на финансовых данных: данные о котировках (с интервалом в один тик) нескольких финансовых инструментов (GAZP, SBER, VTBR, LKOH) за 2 квартал 2017 года с сайта Finam.ru; для каждой точки ряда известны дата, время, цена и объем. И на текстовых данных: экономические новости за 2 квартал 2017 года от компании Форексис; каждая новость является отдельным html файлом.

Постановка задачи

Поставим задачу построения признакового пространства, описывающего тексты (отчёты) с целью их классификации. Даны тексты с меткой времени их появления. Выборка $\mathfrak{D}$ представляет собой векторные описания текстов $\mathbf{x}(t) = [x_1, \ldots, x_m]^\top$ в моменты

времени $\mathbf{t} = [t_1, \ldots, t_m]^\top$. Вектор текста — бинарный вектор наличия отобранных признаков: слов, обладающих наибольшей релевантностью. Задана выборка $\mathfrak{D} = \{(\mathbf{x}_i, y_i)\}$, где $y \in \{0, 1\}$, 0 — stay, 1 — move. Рассматриваются модели-претенденты $\mathfrak{F} = \{f(\mathbf{w}, \mathbf{x})\}$: логистическая регрессия, линейный вектор опорных векторов, случайный лес и градиентный бустинг. Где модель — параметрическое семейство функций $f(\mathbf{w}, \mathbf{x}) = \mu(\mathbf{w}^\top \mathbf{x})$, где в общем случае задач классификации $\mu = \frac{1}{1+exp(-\mathbf{w}^\top \mathbf{x})}$.

Рассмотрим задачу логистической регрессии. Предполагается, что вектор ответов $\mathbf{y} = [y_1, \ldots, y_m]^\top$ — бернуллиевский случайный вектор с независимыми компонентами $y_i \sim \mathfrak{B}(p_i, 1 - p_i)$ и плотностью

$$p(\mathbf{y}|\mathbf{w}) = \prod_{i=1}^{m} p_i^{y_i}(1 - p_i)^{1-y_i} \tag{1}$$

Определим функцию ошибки следующим образом:

$$E(\mathbf{w}) = -\ln p(\mathbf{y}|\mathbf{w}) = -\sum_{i=1}^{m} y_i \ln p_i + (1 - y_i) \ln(1 - p_i) \tag{2}$$

Другими словами, функция ошибки есть логарифм плотности, или функции правдоподобия, со знаком минус. Требуется оценить вектор параметров $\hat{\mathbf{w}}$, доставляющий минимум функции ошибки:

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w} \in \mathbb{R}^n} E(\mathbf{w}) \tag{3}$$

Вероятность принадлежности объекта к одному из двух классов определим как

$$p_i = \frac{1}{1 + \exp(-\mathbf{x}_i^T \mathbf{w})} = \sigma(\mathbf{x}_i^T \mathbf{w}) \equiv \sigma_i \tag{4}$$

Для оценки параметров, воспользовавшись тождеством

$$\frac{d\sigma(\theta)}{d\theta} = \sigma(1 - \sigma)$$

вычислим градиент функции $E(\mathbf{w})$:

$$\nabla E(\mathbf{w}) = -\sum_{i=1}^{m} (y_i(1 - \sigma_i) - (1 - y_i)\sigma_i)\mathbf{x}_i = \sum_{i=1}^{m} (\sigma_i - y_i)\mathbf{x}_i = \mathbf{X}^T(\boldsymbol{\sigma} - \mathbf{y})$$

где вектор $\boldsymbol{\sigma} = [\sigma_1, \ldots, \sigma_m]^\top$ и матрица $\mathbf{X} = [\mathbf{x}_1^T, \ldots, \mathbf{x}_m^T]^\top$ состоит из векторов-описаний объектов.

Оценка параметров осуществляется по схеме Ньютона–Рафсона. Введем обозначение Σ — диагональная матрица с элементами $\Sigma_i i = \sigma_i(1 - \sigma_i), i = 1, \ldots, m$. В качестве начального приближения $\mathbf{w} = [w_1, \ldots, w_n]^\top$ вектора $\hat{\mathbf{w}}$ возьмём

$$w_j = \sum_{i=1}^{m} y_i(1 - y_i), \quad j = 1, \ldots, n$$

.

Оценка параметров $\mathbf{w}_{k+1}$ логистической регрессии (4) на $k+1$-м шаге итеративного приближения имеет вид

$$\mathbf{w}_{k+1} = \mathbf{w}_k - (\mathbf{X}^T\boldsymbol{\Sigma}\mathbf{X})^{-1}\mathbf{X}^T(\boldsymbol{\sigma} - \mathbf{y}) = (\mathbf{X}^T\boldsymbol{\Sigma}\mathbf{X})^{-1}\mathbf{X}^T\boldsymbol{\Sigma}(\mathbf{X}\mathbf{w}_k - \boldsymbol{\Sigma}^{-1}(\boldsymbol{\sigma} - \mathbf{y})) \quad (5)$$

Процедура оценки параметров повторяется, пока норма разности $\|\mathbf{w}_{k+1} - \mathbf{w}_k\|_2$ не станет достаточно мала.

Алгоритм классификации имеет вид:

$$a(\mathbf{x}) = \text{sign}(\sigma(\mathbf{x}, \mathbf{w}) - \sigma_0) \quad (6)$$

где σ_0 — задаваемое пороговое значение функции регрессии (4).

В качестве критерия качества классификации будем использовать AUC (площадь под ROC-кривой). Введём долю верно принятых объектов $TPR = \frac{TP}{TP+FN}$ и долю неверно принятых объектов $FPR = \frac{FP}{FP+TN}$, где TP — истоно-положительное решение, TN — истино-отрицательное решение, FP — ложно-положительное решение, FN — ложно-отрицательное решение (из задачи бинарной классификации с классами {-1} и {1}). Вторым критерием качества классификации выберем меру $F_1 = 2 \cdot \frac{Precision \times Recall}{Precision + Recall}$, где $Precision = \frac{TP}{TP+FP}$ — точность, а $Recall = \frac{TP}{TP+FN}$ — полнота. Тогда чем выше значения AUC и F_1, тем лучше классификатор.

Выборка $\mathfrak{D}$ разбивается на 3 части: тестовую $\mathfrak{D}_t$ (данные с 2002 по 2009 год), на которой происходит обучение, дополнительную $\mathfrak{D}_a$ (данные с 2009 по 2011 год), на которой донастраиваются параметры и контрольной $\mathfrak{D}_c$ (данные с 2011 по 2013 год), на которой производится контроль качества построенных моделей.

Отображение f является бинарной классификацией и отображает признаковое описание текста $\mathbf{x} \in \mathbf{x}(t)$ в метку класса $\{0, 1\}$:

$$f : (\mathbf{w}, \mathbf{x}) \mapsto y$$

Назовём вектор $\mathbf{w}$ вектором параметров классификатора.
Требуется найти оптимальный классификатор $f(\mathbf{x}_i)$ при $\mathbf{x}_i \in \mathfrak{D}_t$ из условия:

$$\hat{f} = \text{argmin}_f S$$

где $S = \{S_1, S_2\}$.

Вычислительный эксперимент

Будем рассматривать модели:

1. Random Forest (RF)
2. Logistic Regression (LR)
3. Linear SVM (LSVC)
4. XGBoost (XGB)

С критериями качества:

1. F1-score
2. AUC-ROC

Используем следующие представления данных для каждой модели:

1. Unigram
2. NMF 50
3. NMF 100
4. NMF 200
5. Ensemble

					Average	
Classifier	Features	Data Set	F1 Score	AUC ROC	F1 Score	AUC ROC
RF	Unigrams	1	0.7811	0.7181		
	Unigrams	2	0.8061	0.4973	**0.8093**	0.5565
	Unigrams	3	0.8408	0.4541		
	NMF 50	1	0.7397	0.6080		
	NMF 50	2	0.8087	0.5000	0.7647	0.5394
	NMF 50	3	0.7458	0.5102		
	NMF 100	1	0.7602	0.5841		
	NMF 100	2	0.8061	0.4973	0.7984	0.5487
	NMF 100	3	0.8288	0.5648		
	NMF 200	1	0.7720	0.7235		
	NMF 200	2	0.8087	0.5000	0.7996	**0.5838**
	NMF 200	3	0.8180	0.5278		
	Ensemble	1	0.7907	0.7198		
	Ensemble	2	0.8018	0.5006	0.8045	0.5617
	Ensemble	3	0.821	0.4648		

Таблица 1. RandomForestClassifier on 3 data sets

					Average	
Classifier	Features	Data Set	F1 Score	AUC ROC	F1 Score	AUC ROC
XGB	Unigrams	1	0.8371	0.7623	**0.835**	**0.5805**
	Unigrams	2	0.8035	0.4946		
	Unigrams	3	0.8643	0.4846		
	NMF 50	1	0.8239	0.7508	0.8284	0.5716
	NMF 50	2	0.8035	0.4946		
	NMF 50	3	0.8577	0.4693		
	NMF 100	1	0.7989	0.7054	0.8257	0.5586
	NMF 100	2	0.8035	0.4946		
	NMF 100	3	0.8747	0.4759		
	NMF 200	1	0.7923	0.6815	0.8221	0.5617
	NMF 200	2	0.8061	0.4973		
	NMF 200	3	0.8679	0.5063		
	Ensemble	1	0.8046	0.7314	0.8217	0.5680
	Ensemble	2	0.8035	0.4946		
	Ensemble	3	0.8571	0.4780		
LR	Unigrams	1	0.8217	0.6873	0.8464	**0.5624**
	Unigrams	2	0.8087	0.5000		
	Unigrams	3	0.9087	0.5000		
	NMF 50	1	0.8235	0.6831	0.8470	0.5610
	NMF 50	2	0.8087	0.5000		
	NMF 50	3	0.9087	0.5000		
	NMF 100	1	0.8154	0.6724	0.8443	0.5575
	NMF 100	2	0.8087	0.5000		
	NMF 100	3	0.9087	0.5000		
	NMF 200	1	0.8244	0.6811	**0.8473**	0.5604
	NMF 200	2	0.8087	0.5000		
	NMF 200	3	0.9087	0.5000		
	Ensemble	1	0.8214	0.6782	0.8463	0.5594
	Ensemble	2	0.8087	0.5000		
	Ensemble	3	0.9087	0.5000		

Таблица 2. XGBClassifier & LogisticRegression on 3 data sets

					Average	
Classifier	Features	Data Set	F1 Score	AUC ROC	F1 Score	AUC ROC
LSVC	Unigrams	1	0.7952	0.5957	0.8309	0.5406
	Unigrams	2	0.8087	0.5000		
	Unigrams	3	0.8889	0.5260		
	NMF 50	1	0.8049	0.6204	**0.8349**	**0.5495**
	NMF 50	2	0.8087	0.5000		
	NMF 50	3	0.8912	0.5281		
	NMF 100	1	0.7933	0.5907	0.8310	0.5396
	NMF 100	2	0.8087	0.5000		
	NMF 100	3	0.8912	0.5281		
	NMF 200	1	0.7962	0.5936	0.8312	0.5399
	NMF 200	2	0.8087	0.5000		
	NMF 200	3	0.8889	0.5260		
	Ensemble	1	0.8029	0.6155	0.8343	0.5479
	Ensemble	2	0.8087	0.5000		
	Ensemble	3	0.8912	0.5281		

Таблица 3. LinearSVC on 3 data sets

Найдём для каждой модели лучшее представление данных по критерию качества F1-score.

Сравнение классификаторов на разных признаках

Будем обозначать модели Random Forest Classifier, XGB Classifier, Logistic Regression и Linear SVC как RF, XGB, LR и LSVC, соответственно. Сравнив модели (RF, XGB, LR, LSVC) на разных признаках (Unigrams, NMF 50, NMF 100, NMF 200, Ensemble) выберем признаки, на которых модели давали лучший результат по F1-score. Такими оказались: Unigrams (для моделей RF и XGB), NMF 50 (для модели LSVC) и NMF 200 (для модели LR).

Оптимизация гиперпараметров

В этом разделе для каждой модели с выбранными представлениями данных для найдём оптимальные гиперпараметры.

max_depth	min_samples_leaf	min_samples_split	n_estimators	Data Set	F1 Score	AUC ROC	Average F1 Score	Average AUC ROC
None	3	5	2000	1	0.8677	0.7710	0.8615	0.5961
				2	0.8087	0.5000		
				3	0.9080	0.5174		
None	3	2	2000	1	0.8663	0.7751	0.8611	0.5946
				2	0.8087	0.5000		
				3	0.9084	0.5087		
10	3	2	1000	1	0.8661	0.7640	0.8611	0.5909
				2	0.8087	0.5000		
				3	0.9084	0.5087		
None	3	2	1000	1	0.8647	0.7681	0.8605	0.5952
				2	0.8087	0.5000		
				3	0.9080	0.5174		
50	3	5	2000	1	0.8640	0.7702	0.8604	0.5930
				2	0.8087	0.5000		
				3	0.9084	0.5087		
10	3	5	2000	1	0.8639	0.7591	0.8603	0.5892
				2	0.8087	0.5000		
				3	0.9084	0.5087		
50	3	2	2000	1	0.8647	0.7681	0.8599	0.5915
				2	0.8087	0.5000		
				3	0.9062	0.5065		
50	3	2	1000	1	0.8624	0.7632	0.8597	0.5935
				2	0.8087	0.5000		
				3	0.9080	0.5174		
None	3	5	1000	1	0.8624	0.7632	0.8597	0.5935
				2	0.8087	0.5000		
				3	0.9080	0.5174		

Таблица 4. RandomForestClassifier with Unigrams

gamma	learning_rate	max_depth	n_estimators	Data Set	F1 Score	AUC ROC	Average F1 Score	Average AUC ROC
0	0.1	3	50	1	0.8291	0.7495	0.8419	0.5824
				2				
				3	0.8880	0.4977		
0.2	0.1	9	50	1	0.8391	0.7269	0.8407	0.5705
				2	0.8087	0.5000		
				3	0.8742	0.4846		
0.2	0.1	9	100	1	0.8457	0.7417	0.8398	0.5725
				2	0.8087	0.5000		
				3	0.8649	0.4759		
0.2	0.01	9	1000	1	0.8387	0.7380	0.8396	0.5764
				2	0.8087	0.5000		
				3	0.8714	0.4911		
0	0.1	9	500	1	0.8415	0.7520	0.8389	0.5794
				2	0.8061	0.4973		
				3	0.8690	0.4889		
0	0.1	9	1000	1	0.8343	0.7483	0.8388	0.5804
				2	0.8061	0.4973		
				3	0.8760	0.4955		
0.2	0.01	3	500	1	0.8219	0.7256	0.8388	0.5737
				2	0.8087	0.5000		
				3	0.8857	0.4955		
0.2	0.1	9	500	1	0.8427	0.7388	0.8387	0.5716
				2	0.8087	0.5000		
				3	0.8649	0.4759		
0.2	0.1	9	1000	1	0.8427	0.7388	0.8387	0.5716
				2	0.8087	0.5000		
				3	0.8649	0.4759		

Таблица 5. XGBClassifier with Unigrams

C	loss	max_iter	multi_class	Data Set	F1 Score	AUC ROC	Average F1 Score	Average AUC ROC
0.1	squared_hinge	1000	ovr	1 2 3	0.8152 0.8087 0.9087	0.6633 0.5000 0.5000	0.8442	0.5544
0.1	squared_hinge	1500	ovr	1 2 3	0.8152 0.8087 0.9087	0.6633 0.5000 0.5000	0.8442	0.5544
0.1	hinge	1500	crammer_singer	1 2 3	0.7852 0.8087 0.9087	0.5982 0.5000 0.5000	0.8342	0.5327
0.1	squared_hinge	1500	crammer_singer	1 2 3	0.7852 0.8087 0.9087	0.5982 0.5000 0.5000	0.8342	0.5327
0.1	squared_hinge	1000	crammer_singer	1 2 3	0.7852 0.8087 0.9087	0.5982 0.5000 0.5000	0.8342	0.5327
0.1	hinge	1000	crammer_singer	1 2 3	0.7852 0.8087 0.9087	0.5982 0.5000 0.5000	0.8342	0.5327
1	hinge	1000	ovr	1 2 3	0.7841 0.8087 0.9087	0.6002 0.5000 0.5000	0.8338	0.5334
1	hinge	1500	ovr	1 2 3	0.7841 0.8087 0.9087	0.6002 0.5000 0.5000	0.8338	0.5334
0.1	hinge	1000	ovr	1 2 3	0.7786 0.8087 0.9087	0.5763 0.5000 0.5000	0.832	0.5254

Таблица 6. LinearSVC with NMF 50

C	max_iter	solver	Data Set	F1 Score	AUC ROC	Average F1 Score	Average AUC ROC
1	150	liblinear	1	0.8235	0.6831	0.847	0.561
			2	0.8087	0.5000		
			3	0.9087	0.5000		
1	100	liblinear	1	0.8235	0.6831	0.847	0.561
			2	0.8087	0.5000		
			3	0.9087	0.5000		
1	150	newton-cg	1	0.8173	0.6683	0.8449	0.5561
			2	0.8087	0.5000		
			3	0.9087	0.5000		
1	100	lbfgs	1	0.8173	0.6683	0.8449	0.5561
			2	0.8087	0.5000		
			3	0.9087	0.5000		
1	100	newton-cg	1	0.8173	0.6683	0.8449	0.5561
			2	0.8087	0.5000		
			3	0.9087	0.5000		
1	150	lbfgs	1	0.8173	0.6683	0.8449	0.5561
			2	0.8087	0.5000		
			3	0.9087	0.5000		
0.1	100	liblinear	1	0.7960	0.6208	0.8378	0.5403
			2	0.8087	0.5000		
			3	0.9087	0.5000		
0.1	150	liblinear	1	0.7960	0.6208	0.8378	0.5403
			2	0.8087	0.5000		
			3	0.9087	0.5000		
0.1	100	lbfgs	1	0.7904	0.5878	0.8359	0.5293
			2	0.8087	0.5000		
			3	0.9087	0.5000		

Таблица 7. LogisticRegression with NMF 200

В таблицах приведены лучшие (по F1-score) модели.

Вывод

Лучшие результаты показала модель Random Forest Classifier на Unigrams с гиперпараметрами $max\_depth = None, min\_samples\_leaf = 3, min\_samples\_split = = 5, n\_estimators = 2000$ со средними значениями F1_score = 0.8615 и AUC ROC = 0.5961 на трёх выборках.

Литература

[1] Hongping Hu, Li Tang, Shuhua Zhang, Haiyan Wang (2018) *Predicting the direction of stock markets using optimized neural networks with Google Trends*, Neurocomputing.

[2] Mikhail Kuznetsov, Anastasia Motrenko, Rita Kuznetsova, Vadim Strijov (2016) *Methods for Intrinsic Plagiarism Detection and Author Diarization*, CLEF (Working Notes).

[3] Heeyoung Lee, Mihai Surdeanu, Bill MacCartney, Dan Jurafsky (2014) *On the Importance of Text Analysis for Stock Price Prediction*, Proceedings of the Ninth International Conference on Language Resources and Evaluation.

[4] Anna Potapenko, Artem Popov, Konstantin Vorontsov (2017) *Interpretable probabilistic embeddings: bridging the gap between topic models and neural networks*, CoRR.

[5] Andrew Sun, Michael Lachanski, Frank J. Fabozzi (2016) *Trade the tweet: Social media text mining and sparse matrix factorization for stock market prediction*, International Review of Financial Analysis.

[6] Усманова К. Р., Кудияров С. П., Мартышкин Р. В., Замковои А. А., Стрижов В. В. (2018) *Анализ зависимостей между показателями при прогнозировании объема грузоперевозок*, Системы и средства информатики.