

САНКТ–ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ

Математико–механический факультет
Кафедра системного программирования

SaaS-платформа для прогнозирования
продаж

Магистерская диссертация студента 661 группы
Афанасенко Никиты Владимировича

Научный к.ф.–м.н., доц. Графеева Н. Г.
руководитель /подпись/

Рецензент I. Shablinsky, Ph.D.
/подпись/

“Допустить к защите” д.ф.–м.н., проф. Терехов А. Н.
Заведующий кафедрой /подпись/

Санкт–Петербург
2012

SAINT PETERSBURG STATE UNIVERSITY

Mathematics & Mechanics Faculty

Software Engineering Chair

SaaS platform for sales forecasting

by

Afanasenko Nikita Vladimirovich

Master's graduation paper

Supervisor Associated Professor N.G. Grafeeva

Reviewer I. Shablinsky, Ph.D.

“Approved by” Professor A.N. Terekhov
Head of Department

Saint Petersburg

2012

Содержание

Введение	5
Постановка задачи	7
Обзор предметной области	8
Существующие приложения для прогнозирования продаж	8
Методы прогнозирования	10
Методы, основанные на скользящем среднем	13
Регрессионные модели	14
Методы Бокса–Дженкинса	15
Census X11	17
Нейронные сети	19
Особенности прогнозирования продаж	20
Разработка метода прогнозирования продаж	21
Анализ требований к методу прогнозирования	21
Описание прогнозирующей модели	24
Метод “Гусеница”	24
Нейронная сеть	27
Сравнение с существующими методами	29
Исполнение в распределенной системе	31
Разработка SaaS платформы	32
Общая архитектура сервиса	33
Реализация компонентов приложения	35
Фронтенд–приложение	35
Веб–сервис для прогнозирования	35
Прогнозирующие агенты	37

Заключение	38
Направления развития	38
Список использованной литературы	39

Введение

В современных условиях жесткой конкуренции ключевой составляющей успеха многих компаний является их способность адаптироваться к изменяющимся условиям в их бизнес-среде. Таким образом критически важным для компании становятся хорошие прогнозы ключевых показателей бизнеса.

Эффективная система прогнозирования способна оптимизировать работу и процессы практически во всех областях деятельности предприятия: отдел продаж может лучше выстраивать ценовую политику для каждой товарной позиции и оптимизировать сбыт; производственные подразделения могут планировать объемы производства; отдел маркетинга – оценивать и планировать продажи в зависимости от рыночной стратегии; отдел логистики – лучше планировать цепочку поставок: от доставки до складских помещений.

Вопрос планирования логистики особенно остро стоит для компаний из пищевой отрасли – конечный срок годности продуктов не позволяет запасаться большими объемами впрок. С другой стороны существует риск упущенной прибыли из-за отсутствия необходимых продуктов на складе. В работе [1] говорится о том, что 48% компаний пищевой отрасли так или иначе не довольны своими прогнозами.

Например, при планировании закупок компаниями нередко используются простые эвристические методы, например ABC-анализ, основанный, по-сути, на предположении о том, что в следующем месяце все будет примерно так же, как и в предыдущем, и правиле “за 80% продаж отвечают 20% запасов”. Безусловно,

подобный подход к прогнозированию значительно уступает более точным математическим методам.

Следует также отметить недоступность хороших инструментов небольшим компаниям, не обладающим достаточными ресурсами, чтобы позволить себе дорогостоящие программные продукты для прогнозирования. Например, интернет–магазин детских игрушек с несколькими десятками продаж за день не станет покупать приложение за несколько тысяч долларов, с которым, чтобы добиться приемлемых результатов, придется работать эксперту.

Все вышесказанное подводит нас к актуальности поставленной в данной работе проблемы: необходим удобный и доступный инструмент для прогнозирования продаж, встраиваемый в привычные для компании бизнес–процессы.

Постановка задачи

Задача данной работы состоит в построении прототипа универсальной SaaS–платформы для прогнозирования спроса на произвольные виды товаров и товарных групп на примере базы данных продаж крупной Московской сети ресторанов.

- На основании результатов анализа существующих программных средств, выявить их общие недостатки.
- Проанализировать существующие методы прогнозирования, а также выявить особенностей задачи прогнозирования продаж.
- Разработать метод прогнозирования продаж и сделать сравнение с наиболее часто употребляемыми на данный момент методами.
- Спроектировать и разработать веб–сервис, а также SaaS–платформу на его основе, которая может быть использована в качестве базы для построения произвольного клиентского приложения.

Обзор предметной области

В этой главе будут рассмотрены программные средства для прогнозирования продаж, существующие у них недостатки и способы их исправления, а также методы прогнозирования и особенности прогнозирования продаж в частности.

Существующие приложения для прогнозирования продаж

Для данных ниже приложений будут рассматриваться используемые ими методы прогнозирования (если доступны), простота процесса прогнозирования для неэксперта, а также возможности интеграции в другие программные комплексы.

На момент публикации данной работы многие из существующих решений как платформу **используют MS Excel**, например, это [2] и ранние версии [3].

Обычно в таких приложениях от пользователя требуется явно указать ряд как диапазон ячеек и, в зависимости от метода, будет либо сразу выстроен прогноз, либо необходимо будет ввести параметры прогнозирования. Такой способ построения прогнозов может быть удобен только в случае тесной интеграции отчетов по продажам в MS Excel в бизнес–процесс компании. Но даже в этом случае, при анализе и сведении прогнозов по большим группам товаров или, например, планировании склада, данный продукт влечет за собой большое количество рутинных действий по импорту/экспорту данных.

Приложение [4] (также на MS Excel) в качестве метода анализа ряда продаж используется некий метод декомпозиции с возможностью выделения нескольких вариантов тренда и определения сезонного коэффициента. Для прогнозирования используется несколько вариантов метода скользящего среднего.

Продукт [5] – это клиентское приложение ОС Windows.

Единственный способ интеграции с другими приложениями – импорт/экспорт данных. В качестве моделей для прогнозирования используются классические методы: скользящего среднего, Бокса–Дженкинса, Хольта–Уинтерса.

[6] – Windows–приложение; для прогнозирования используется собственный метод, суть которого состоит в оценке степени применимости различных моделей для конкретного ряда; выбирается в итоге та, что лучше всех аппроксимирует/прогнозирует наблюдаемый ряд. Такая стратегия по сравнению со стратегиями конкурирующих продуктов выглядит лучше, т.к. итоговая модель быстрее адаптируется к изменению статистических характеристик ряда. Данный вывод подтверждается результатами конкурса продуктов для прогнозирования, в котором ForecastX занял первое место. [7]

[8] – так же, как и предыдущий продукт, подбирает лучшую для прогнозирования модель, но в выборе больше полагается на мнение эксперта. Помимо классических методов, для прогнозирования в версии Ultimate реализован метод Census X-11.

Комплексный продукт [9] помимо модуля прогноза продаж, также, например, имеет модуль управления складом (в планировании основывается на данные прогноза). Имеет ряд внедрений для

крупных Российских компаний: ТД «Перекресток», ГК «Связной», ОАО «Пивоваренная компания Балтика».

[10] – SaaS платформа для прогнозирования продаж с закрытым API и единственным клиентским приложением под ОС Windows.

В целом к минусам рассмотренных решений можно отнести:

- закрытость API для доступа к функциям прогнозирования (отсюда следует сложность или невозможность интеграции с существующими продуктами),
- небольшое количество доступных моделей прогнозирования, а также их относительная простота,
- сложности работы с большими количествами прогнозируемых рядов,
- ограничение вычислений на один компьютер – для больших объемов данных и большого количества товаров это сразу уменьшает время получения прогноза,
- в большинстве случаев, приложение реализовано для единственной операционной системы – Windows.

Методы прогнозирования

Рассмотрим подробнее методы, используемые в существующих сейчас программных продуктах.

В целом методы прогнозирования продаж можно разделить на три группы:

- методы экспертных оценок;
- причинно–следственные методы;

- методы анализа и прогнозирования временных рядов.

Методы экспертных оценок – это группа методов так или иначе опирающихся на субъективное мнение специалистов, причем для уменьшения риска ошибки, общая картина прогноза обычно складывается из взвешенных оценок нескольких экспертов. Эти методы зачастую применяются к новым товарам или рынкам, данных по которым обычно недостаточно для адекватного прогноза автоматизированными средствами.

Экспертами в данном случае обычно предоставляется распределение вероятностей по возможным значениям предсказываемой величины, например:

Объем продаж, тыс. руб.	Вероятность, %
350–450	25
450–500	50
500–550	25

Причинно–следственные методы требуют глубокого изучения составляющих частей исследуемого экономического процесса, поскольку прогнозирующая модель строится на основе связанных с ним процессов и событий, оценки и предположения о которых обычно может дать только эксперт. На предварительном этапе работы осуществляется поиск факторов, влияющих на наблюдаемый процесс, а также производится оценка их связи. Например, в работе [11] проведен расчет регрессионной модели потребления напитка “Тархун”:

$$Y(X_1; X_2; \dots; X_n) = \varepsilon + b_1 X_1 + \dots + b_n X_n,$$

где Y – прогнозируемое значение объема продаж; X_n – независимые факторы (в данном примере: уровень доходов потребителей, цены на продукты конкурентов); ε и b_n – ошибка и коэффициенты регрессии соответственно.

В результате проведенного анализа, у авторов получилась следующая модель:

$$Y_{t+1} = 2,021 + 0,743A_t + 0,856Y_t,$$

где Y_t – объем продаж в месяце t ; A_t – затраты на рекламу в месяце t .

В целом рассмотрение методов, включающих в себя любые экспертные оценки, выходит за рамки данной работы, цель которой, как было отмечено выше, состоит в построении полностью автоматизированного инструмента прогнозирования.

Математические методы анализа временных рядов чаще всего так или иначе основаны на разбиении наблюдаемого ряда на составляющие (выделение постоянной компоненты ряда, состоящей из тренда и сезонных колебаний, а также предсказание случайной независимой величины) и экстраполяции полученных из прошлого наблюдений. Помимо этого в прогнозировании объема продаж используются также методы искусственного интеллекта, основанные, например, на применении нейросетей [12].

Тренд — это изменение, определяющее общее направление развития, основную тенденцию ряда. Часто выделяют линейный, логарифмический, полиномиальный и экспоненциальный тренды.

Сезонные колебания — повторяющиеся изменения значения ряда в определенные промежутки времени. Сезонность в продажах, в целом, свойственна большому числу товаров и услуг. Например, прохладительные напитки лучше продаются в летний период, услуги авиаперевозчиков пользуются гораздо большим спросом в популярные периоды отпусков. Сезонность может проявляться не только в виде годовых колебаний – парикмахерским вполне могут быть свойственны, например, 4–6 недельные колебания.

Ниже будут рассмотрены основные модели, используемые для прогнозирования объема продаж в экономике в целом, а также в описанных выше программных продуктах в частности.

Методы, основанные на скользящем среднем

В основе данной группы методов лежит предположение о том, что некоторый заданный период в прошлом хорошо описывает будущие значения наблюдаемого ряда. Моделью обычно является некоторая достаточно простая функция от значений в прошлом.

Простейший пример модели скользящего среднего:

$$SMA = (P_{i-1} + P_{i-2} + \dots + P_{i-n}) / n,$$

где P_i – наблюдение в момент i .

Модель взвешенного скользящего среднего же учитывает, например, давность наблюдений, сопоставляя каждому значению в формуле некоторый вес:

$$WMA = (w_{i-1}P_{i-1} + w_{i-2}P_{i-2} + \dots + w_{i-n}P_{i-n}) / (w_{i-1} + \dots + w_{i-n})$$

Описанные модели могут использоваться в достаточно простых случаях на устоявшихся рынках для экстраполяции трендов, но они

сильно проигрывают при наличии ярко выраженных колебаний и не способны адаптироваться к сезонности.

Регрессионные модели

В общем виде регрессионную модель, опирающуюся на факторы, не зависящие от наблюдаемого ряда, можно отнести к причинно–следственной группе моделей, но существуют вариации, использующие только значения ряда из прошлого – это модель авторегрессии $AR(p)$ порядка p .

$$AR(p) = c + \sum_{i=1}^p b_i X_{t-i} + \varepsilon_t,$$

где C – константа, b_i – параметры модели, ε_i – ошибки.

Предполагается, что ошибки независимы и нормально распределены.

Задача метода состоит в определении параметров b_i . Обычно это делается с помощью системы уравнений Юла–Уокера [13], но можно воспользоваться и более простым методом наименьших квадратов.

Данная модель лучше адаптируется к изменению показателей и сезонности ряда, чем модель скользящего среднего, но как основа для прогноза так же редко используется на практике.

Описанные выше модели достаточно просты и обычно могут служить основой для прогнозирования только достаточно устойчивых рядов.

Методы Бокса–Дженкинса

Это методология и класс алгоритмов, разработанных и исследованных в середине девяностых двумя статистиками, Боксом (Box) и Дженкинсом (Jenkins). Для описания классической методологии ARIMA потребуется ввести ряд определений:

Стационарность – неизменность статистических характеристик ряда со временем. В широком смысле означает:

- $MX(t) = const$ – неизменность математического ожидания со временем;
- $\rho(t, s) = \rho(t - s)$ – зависимость автокорреляционной функции только от $(t - s)$.

Интегрированность порядка k – свойство нестационарного временного ряда, при котором разности ряда порядка k являются стационарным временным рядом.

$ARMA(p, q)$ – модель авторегрессии порядка p , скользящего среднего порядка q , используется для прогнозирования стационарных временных рядов. В нотации Бокса–Дженкинса записывается как:

$$\left(1 + \sum_{i=1}^p b_i L^i\right) X_t = \left(1 + \sum_{i=1}^q w_i L^i\right) \varepsilon_t,$$

где b_i, w_i – параметры модели, $\{\varepsilon_t\}$ – последовательность независимых нормально распределенных случайных величин, $L^i X_t = X_{t-i}$ – оператор лага. Эту модель можно представить как

авторегрессионную модель, где в качестве остатка используется скользящее среднее из элементов белого шума.

Модель ARIMA [14] является обобщением модели ARMA на нестационарные ряды и заключается в следующем:

- оценивается стационарность ряда;
- определяется порядок интегрированности ряда (обычно ограничен первым или вторым порядком);
- в случае интегрированности порядка больше 0 ряд преобразуется взятием конечных разностей в стационарный ряд, для которого строится модель ARMA:

$Y_t = (1 - L)^d X_t$, где d – порядок интегрированности;

$$\left(1 + \sum_{i=1}^p b_i L^i\right) Y_t = \left(1 + \sum_{i=1}^q w_i L^i\right) \varepsilon_t$$

- теперь стационарный процесс Y_t может быть спрогнозирован, а из прогноза Y_t можно получить прогноз X_t .

Как видно из определения, описанная модель оперирует данными только из предыстории, однако существуют варианты реализации, при которых учитываются также некоторые независимые факторы [15].

Модель достаточно часто используется на практике и, как было показано выше, реализована в нескольких популярных продуктах для прогнозирования продаж.

Census X11

Модель Census X11 является улучшением классического метода Census I сезонной декомпозиции ряда [16], разработанной и поддерживаемой Бюро Переписи США (US Census Bureau).

В общем случае любой ряд раскладывается на следующие четыре компоненты:

- тренд (T_t),
- сезонную компоненту (S_t),
- циклическую компоненту (C_t , отличается от сезонной тем, что периодичность может меняться от цикла к циклу),
- случайную величину (I_t).

Двумя основными способами взаимосвязи этих компонентов являются:

- аддитивная модель: $X_t = T_t C_t + S_t + I_t$,
- мультипликативная модель: $X_t = T_t C_t S_t I_t$.

Для декомпозиции используется подход состоящий из следующих шагов (рассматривается аддитивная модель, для мультипликативной все аналогично):

- выделение сезонной компоненты путем вычитания из исходного ряда ряда скользящего среднего с шириной окна равной периоду сезонности, и взятию среднего от значений ряда, соответствующих данной точке сезона;

- сезонная корректировка – вычитание из исходного ряда сезонной составляющей;
- выделение тренд–циклической компоненты, например, путем взятия пятиточечного взвешенного скользящего среднего с весами 1, 2, 3, 2, 1;
- выделение случайной составляющей как остатка от вычитания сезонной и тренд–циклической компонент.

Модель Census X11 вносит коррективы в классический метод, показавшие свою состоятельность при практическом использовании метода:

- поправки на число рабочих дней – из–за разного количества дней в месяце может возникать эффект, при котором в определенный месяц попадает больше пиковых дней (например, выходных), что является причиной ошибок в расчетах сезонности; метод позволяет внести корректировки в исходный ряд для исключения этого эффекта;
- корректировка выбросов – перед выделением сезонной компоненты правятся значения, выбивающиеся из определенного диапазона;
- описанные выше корректировки применяются многократно на нескольких этапах метода.

Как классический, так и улучшенный методы Census нередко используются для моделирования и прогнозирования, однако при автоматизированном моделировании без эксперта, для выбора оптимальных параметров требуются дополнительные сравнения и проверки моделей.

Нейронные сети

Нейронные сети для задачи прогнозирования продаж используются с середины девяностых годов [17]. Их неоспоримым плюсом является простота построения модели со многими независимыми величинами (например, регрессионная модель на основе ARIMA не будет обладать достаточной гибкостью при большом количестве независимых величин), а также способность выделять тренд и сезонность путем подачи на входы нейросети данных за предыдущие периоды, а также значений независимых величин.

В самом простом варианте используется многослойный с несколькими (до трех) скрытыми слоями, на выход которого подается прогнозируемое значение. В исследовании [12] показано как нейросетевая модель дает на 2% лучший показатель средней ошибки прогноза по сравнению с моделью ARIMA.

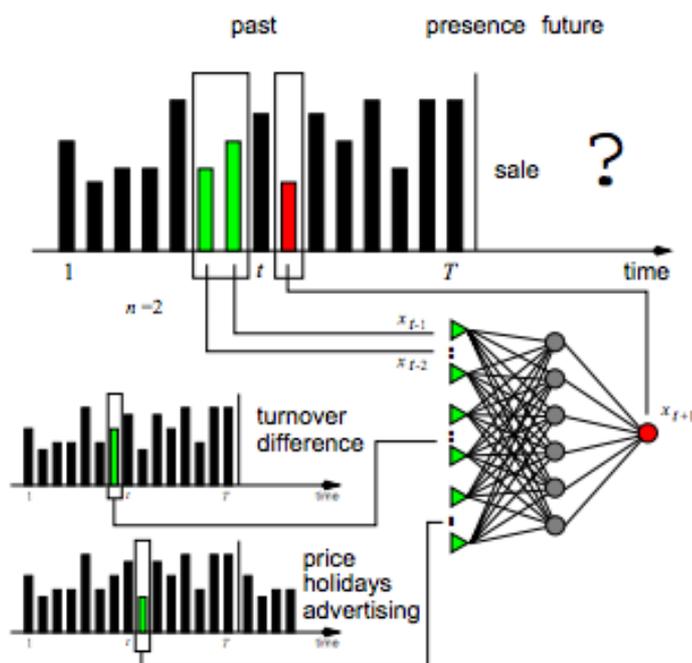


Рисунок 1. Вход и выход нейросети из примера в работе [12]

Особенности прогнозирования продаж

Задача прогнозирования временных рядов продаж имеет ряд особенностей, налагающих ограничения на методы ее решения:

- ряды продаж нестационарны, могут менять свои характеристики со временем, претерпевать значительные всплески (вызванные, например, промо–акциями);
- падение уровня продаж не всегда может отображать реальное падение спроса на товар, часто это может быть лишь отражением проблем в поставках товара;
- статистические характеристики рядов могут значительно отличаться для различных товаров, так некоторые товары могут устойчиво продаваться ежедневно, когда другие имеют только несколько продаж в неделю;
- величина потерь при завышении или занижении прогнозируемого значения относительно фактического различна, поскольку каждый случай ведет к различным уровням расходов, например, на логистику и склад;
- временной ряд продаж может находиться в сложной зависимости от большого количества независимых величин, таких как расходы на рекламу, праздничные дни, температура воздуха и прочих.

Разработка метода прогнозирования продаж

В данной главе дано описание предлагаемого метода прогнозирования, который, как будет показано, способен улучшить показатели ошибки прогнозирования и потерь, связанных с неточностью прогнозов.

На основании исследования особенностей задачи, а также существующих подходов, описаны общие концепции, которые легли в основу метода, дано описание общей архитектуры метода, а также приведены пример работы и сравнение результатов с результатами прогнозов по классической модели ARIMA.

Анализ требований к методу прогнозирования

На основе анализа особенностей задачи прогнозирования продаж, выделим ряд требований к методу, а также отметим возможные варианты решения связанных с ними задач:

- **Устойчивость к единичным всплескам и падениям**

В случае временных рядов объема продаж единичные всплески и падения являются нередкой ситуацией (например, во время промо-акций или ограничения поставок). Метод должен правильно классифицировать такие значения как аномалии и, по возможности, прогнозировать их отдельно. Нередко это задача решается с помощью фильтрации ряда перед построением прогноза. Например, для прогнозирования осадков описан метод, когда аномалии выделяются методом Fuzzy C-means [18], а затем прогнозируются отдельно с помощью нейронной сети [19].

- **Возможность работы с редко продаваемыми товарами**

Некоторые товары (например, автомобили у небольших дилеров) могут продаваться не чаще, чем раз в неделю. Обычно редкими

считают ряды, в которых больше 30% значений – нули (Рисунок 2). Для таких рядов разработан ряд специфических методов (например, метод Виллемейна [20]), но в простейшем случае, если не требуется точных дат продаж, мы можем масштабировать ряд так, чтобы он приобрел привычный вид.

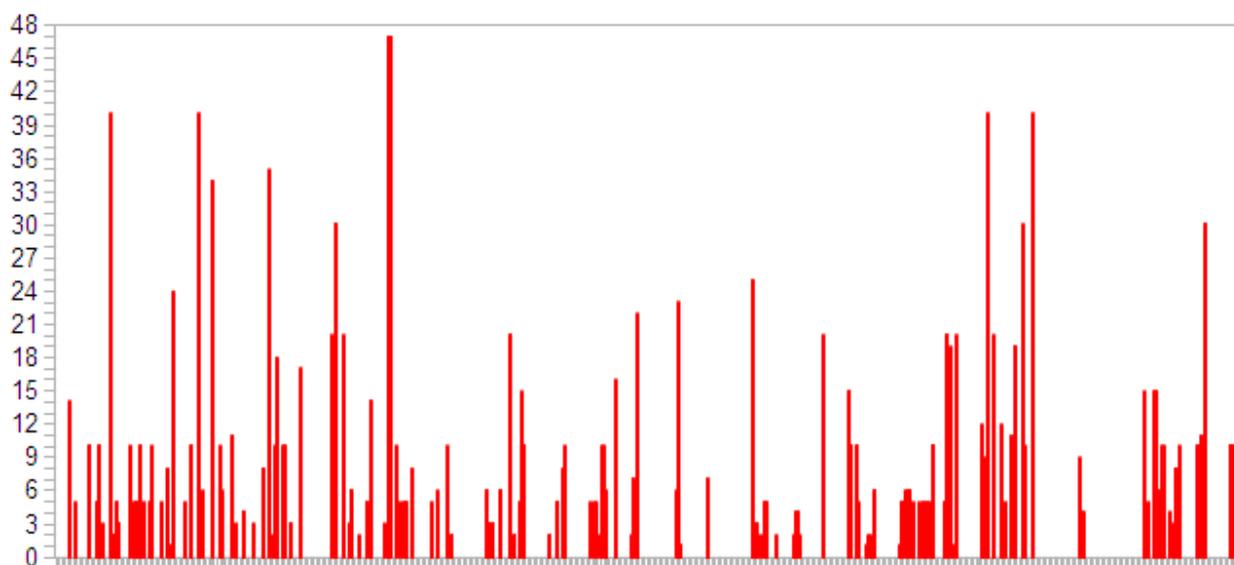


Рисунок 2. Ряд редко продаваемых товаров

- **Учет сезонности продаж**

Большинство классических методов (например, основанных на скользящем среднем и регрессионных моделях) никак не выделяют сезонную компоненту, что может сказаться на качестве прогноза. Некоторые же, например, Census, ограничиваются сезонным коэффициентом, что в случае присутствия нескольких сезонных компонент (недельный и полугодовой) может также повлиять на прогнозы разных масштабов. Необходим метод, способный выделить несколько сезонных компонент – это даст нам возможность получить более точный анализ ряда (в том числе и для нахождения коррелирующих рядов), а так же прогноз.

- **Метод должен настраиваться сам, без участия эксперта**

Большая часть параметризованных методов требуют участия экспертов в той или иной мере, но этого нередко можно избежать,

используя эвристические методы для нахождения оптимальных параметров. Например, методы для сезонных рядов дают лучший результат, если значением параметра сезонности дать точную длину периода. Для простых рядов, эксперт может определить период просто взглянув на ряд, мы же, обладая знанием о том, что чаще всего периодом является неделя, месяц и т.д., можем построить несколько моделей и каждый раз выбирать для оптимальную [21]. Также для поиска оптимальной модели могут использоваться генетические алгоритмы [22].

- **Учет потерь при завышении/занижении прогноза**

Как было показано выше, потери при завышении и занижении прогноза могут быть различны. Этот коэффициент для разных товаров так же может отличаться и в простейшем случае зависит от их цены и себестоимости. Для учета этой особенности мы можем использовать собственную функцию ошибки для каждого товара, для которого доступна необходимая информация.

- **Учет зависимости от внешних факторов**

На уровень объема продаж влияют внешние факторы, значения которых мы часто можем предсказывать с большой вероятностью (например, расходы на рекламу). Таким образом, модель должна строить прогноз, учитывая зависимости от значений внешних рядов. Выше было отмечено как для этих целей используются регрессионные модели и нейронные сети.

- **Возможность многопоточного исполнения**

Немаловажным требованием к современным методам является их возможность работать в многопоточных/распределенных системах. Особенно важным это требование становится при использовании метода в бизнес-процессах быстрорастущей компании, потребности в прогнозировании которой увеличиваются быстрее, чем оптимизируется скорость работы алгоритма прогнозирования.

Описание прогнозирующей модели

Взяв в расчет вышеизложенные требования, автор пришел к следующей модели прогнозирования.

Метод “Гусеница”

Для удовлетворения требований фильтрации аномальных данных, а также декомпозиции на сезонные составляющие и тренд был использован метод “Гусеница” [23]. Метод независимо разрабатывался в России, Великобритании и США (где его аналог получил название SSA – Singular Spectrum Analysis) и отлично зарекомендовал себя как инструмент анализа и прогнозирования рядов в метеорологии, геофизике, экономике и других областях.

Результатом применения метода является разложение временного ряда на простые компоненты: медленные тренды, сезонные и другие периодические составляющие, а также шумовые компоненты. Полученное разложение может служить основой прогнозирования как самого ряда, так и его отдельных компонентов.

Суть метода состоит в следующем:

- одномерный ряд преобразуется в матрицу Z с помощью сдвигового оператора с параметром m – временем жизни гусеницы;
- выполняется SVD–разложение ковариационной матрицы полученной выборки:

$$C = \frac{1}{n} ZZ^T = V \Lambda V^T,$$

где $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_\tau)$ – диагональная матрица собственных

чисел, $V = (\mathbf{v}^{(1)}, \dots, \mathbf{v}^{(\tau)}), (\mathbf{v}^{(i)})^T \mathbf{v}^{(j)} = \delta_{ij}$ – ортогональная матрица собственных векторов;

- далее рассматривается система главных компонент:

$$U = V^T Z, U = (U^{(1)}, \dots, U^{(\tau)})^T,$$

- для прогноза необходимо выделить несколько главных компонент, по которым будет строиться продолжение ряда; на практике эксперт оценивает величины собственных чисел, свойства каждого собственного вектора, соответствующего достаточно большим собственным числам, а также исходный ряд, восстановленный по наблюдаемым наборам главных компонент:

$$Z' = V' U',$$

$$\text{где } V' = (\mathbf{v}^{(i_1)}, \dots, \mathbf{v}^{(i_r)}), U' = V'^T Z;$$

по матрице Z' исходный ряд восстанавливается как средние значения по каждой диагонали;

- само прогнозируемое значение для ряда $(f_i)_{i=1}^N$ по набору главных компонент i_1, \dots, i_r в точке $N + 1$ вычисляется так:

$$f_{N+1} = \omega (V_*^T V_*)^{-1} V_*^T Q, \text{ где}$$

$$\omega = \begin{pmatrix} \mathbf{v}_\sigma^{(i_1)} & \dots & \mathbf{v}_\sigma^{(i_r)} \\ \dots & \dots & \dots \\ \mathbf{v}_\tau^{(i_1)} & \dots & \mathbf{v}_\tau^{(i_r)} \end{pmatrix},$$

$$V_* = \begin{pmatrix} \mathbf{v}_1^{(i_1)} & \dots & \mathbf{v}_1^{(i_r)} \\ \dots & \dots & \dots \\ \mathbf{v}_{\sigma-1}^{(i_1)} & \dots & \mathbf{v}_{\sigma-1}^{(i_r)} \\ \mathbf{v}_{\sigma+1}^{(i_1)} & \dots & \mathbf{v}_{\sigma+1}^{(i_r)} \\ \dots & \dots & \dots \\ \mathbf{v}_{2\sigma-1}^{(i_1)} & \dots & \mathbf{v}_{2\sigma-1}^{(i_r)} \\ \dots & \dots & \dots \\ \mathbf{v}_{\tau-1}^{(i_1)} & \dots & \mathbf{v}_{\tau-1}^{(i_r)} \end{pmatrix}$$

$$Q = (f_{N-\sigma+2}^{(1)}, \dots, f_N^{(1)}, \dots, f_{N-\sigma+2}^{(s)}, \dots, f_N^{(s)})^T.$$

Таким образом, базовый метод параметризуется двумя значениями: длиной окна, а также набором главных компонент. В работе [24] авторы рекомендуют для рядов с несколькими сезонными компонентами использовать длину окна, кратную периодам сезонности и близкую к половине длины ряда.

На основе анализа предоставленных рядов продаж было выявлено несколько значений длины окон гусеницы, соответствующих рекомендациям авторов, а также показавших лучшую отделимость компонентов и лучший результат прогноза при адекватном времени работы метода. Для дневных рядов это значения 30, 90, 180 и близкие к ним. Также отметим, что на практике в качестве набора главных компонент как правило использовался набор из некоторого числа первых собственных чисел.

Для реализации метода “Гусеница” в работе использовались компоненты библиотеки GNU Scientific Library.

В связи с относительно большим временем работы метода на продолжительных рядах, необходимо было ограничить перебор возможных параметров при их оценке. Для этого реализован простейший вариант генетического алгоритма:

- в качестве особи используются вектора (w, n) , где w – длина окна, $w \in [25..35, 80..100, 160..200]$, n – количество первых собственных компонент, $n \in [5..30]$,
- в первоначальную популяцию берутся особи из всех комбинаций значений $[30, 90, 180]$ и $[5, 7, 10, 13, 18, 25]$, размер популяции всегда постоянен,
- для оценки особи используется значение среднеквадратической ошибки прогноза на случайном участке ряда (выбирается для каждой итерации),
- процедура скрещивания меняет один из параметров местами, вероятность мутации (± 1 к значению) равна 5%, скрещиваются 10 лучших особей,
- в качестве условий останова используются ограничение на число итераций и заданная точность прогноза лучшего метода.

Нейронная сеть

Также в ряде работ отмечены хорошие прогнозирующие способности нейронных сетей, в том числе и в вопросе учета сезонных колебаний рядов. Очевидно, мы можем использовать нейросеть и прогнозы по методу “Гусеница” в одной модели [25] – на несколько нейронов входного слоя сети будут подаваться прогнозы по “Гусенице” с

различными параметрами. При этом ожидается, что сеть должна автоматически подстроиться под оптимальный прогноз с лучшими параметрами.

Также неоспоримым преимуществом нейронной сети является возможность удовлетворения требования по учету зависимости от внешних факторов. Так помимо исторических данных на вход будут подаваться значения рядов внешних факторов: для прототипа будет использоваться ряд с данными среднедневной температуры за день.

Таким образом была выбрана следующая структура входного слоя нейронной сети: 30 предыдущих значений исходного ряда, 5 значений прогнозов по методу “Гусеница” с параметрами, выбранными генетическим алгоритмом, 30 последних значений среднедневной температуры.

В общем случае мы не можем однозначно определить входной слой для каждого товара (в основном из-за случайной природы выбора параметров метода “Гусеница” и использования различных внешних рядов для разных товаров), поэтому, принимая во внимание требование отсутствия эксперта, нам необходимо автоматизировать процесс выбора топологии нейросети. Для этого используем реализованный в библиотеке Fast Artificial Neural Network метод каскадного обучения.

Каскадное обучение отличается от обычного тем, что начинается с пустой сети и добавляет нейроны один за другим в процессе тренировки. Основная идея этого способа состоит в последовательном обучении нейронов-кандидатов отдельно от целевой сети и добавлении лучших из них в скрытые слои нейросети.

В ходе экспериментов было отмечено, что такая нейросеть не дает желаемую стабильность результата – значения прогноза на обученных на одних и тех же участках ряда сетях были различны, причем некоторые – значительно, хотя на любом отрезке ряда можно было выбрать лучшую в плане прогнозирующих свойств модель. Для решения этой проблемы был использован подход, описанный в работе [21]:

- для ряда $y(\tau)$ и базовых моделей b_1, \dots, b_p строится модель:

$$a(t) = \sum_{i=1}^p w_{it} b_i(t), \sum_{i=1}^p w_{it} = 1,$$

- веса w_{it} рассчитываются в каждый момент времени t :

$$w_{it} = \begin{cases} 1, & i = I(t) \\ 0, & i \neq I(t) \end{cases}, I(t) = \arg \min_{i=1, \dots, p} \sum_{\tau=1}^{t-1} \theta^{t-\tau-1} (b_i(\tau) - y(\tau))^2,$$

где параметр $\theta \in [0, 1]$ задает “скорость забывания предыстории”.

Таким образом, общая схема разработанной модели представлена на Рисунке 3.

Сравнение с существующими методами

Для экспериментов использовались данные по объему продаж за последние 3 года работы, предоставленные крупной московской сетью ресторанов. Для 10 популярных продуктов по пяти 30–дневным интервалам вычислялась средняя абсолютная ошибка прогноза в процентах. Результаты сравнения описанного выше метода с методами прогнозирования с помощью нейронной сети

улучшение оказалось значительным, что явно говорит о необходимости учета внешних факторов при прогнозировании.

Исполнение в распределенной системе

В описанном методе вычисления строятся по каскадному принципу, что позволяет, например, реализовать его по модели MapReduce [26] с двумя этапами:

- на первом этапе на шаге Map подсчитываются прогнозы по каждому набору параметров метода гусеница, и соединяются в массив входов на шаге Reduce;
- на втором этапе на шаге Map подсчитываются прогнозы по каждой нейронной сети, а на шаге Reduce выполняется расчет весов в линейной комбинации, и считается итоговый прогноз.

Также отметим, что базовая реализация метода “Гусеница” является хорошо распараллеливаемой задачей из-за присутствия большого количества умножений матриц. Нейросети по своей природе также являются хорошим кандидатом для распараллеливания.

Разработка SaaS платформы

Как было отмечено ранее, к недостаткам современных продуктов для прогнозирования относятся:

- их дороговизна (и как следствие недоступность качественных решений для малых предприятий),
- неспособность интеграции этих продуктов с существующими приложениями, используемыми в бизнес-процессах компании,
- ограничение вычислительной мощности одним компьютером,
- реализация только для одной операционной системы – Windows.

Предоставление услуги прогнозирования по модели SaaS в виде веб-сервиса с публичным API в данном случае позволяет решить эти проблемы.

SaaS (software as a service, программное обеспечение как услуга) – модель продажи и использования ПО, при которой поставщик предоставляет доступ к продукту по запросу и через интернет. К несомненным преимуществам такого подхода относятся:

- **Минимизация затрат предприятия на ПО**

По-сути программный продукт сдается в аренду за относительно небольшую ежемесячную плату, либо оплачивается на основе количества использованных ресурсов или произведенных операций. Также отсутствуют затраты на техническую поддержку – как правило, она уже включена в стоимость.

- **Доступ компании к сложной инфраструктуре без необходимости ее поддержки**

Компания-клиент не несет расходов на поддержку инфраструктуры продукта, которая может быть сколько угодно

сложной (особенно в задачах прогнозирования, где для расчетов может использоваться кластер из десятков машин).

- **Обновление ПО происходит незаметно для клиента**
Также компания не вкладывает свои средства в разработку новых версий.
- **SaaS-модель исключает возможность нелицензионного использования ПО.**

Общая архитектура сервиса

К компонентам приложения относятся:

- веб-приложение, выполняющее роль фронт-офиса для клиентов, где они могут зарегистрироваться, управлять своим счетом и получать данные для авторизованного использования API,
- публичный веб-сервис, реализующий протокол общения между сервисом прогнозов и клиентскими приложениями,
- приложения, получающие задачи из общей очереди и выполняющие непосредственно само прогнозирование рядов.

Общая схема архитектуры взаимодействия компонентов разработанного решения представлена на Рисунке 4.

При выборе инструментов для реализации прототипа описываемого ПО решающими были следующие факторы:

- **Скорость разработки**
Поскольку задачей является реализация именно прототипа, демонстрирующего принципы архитектуры в целом, скорость разработки сервиса и уровень абстракции являются одними из наиболее важных факторов.

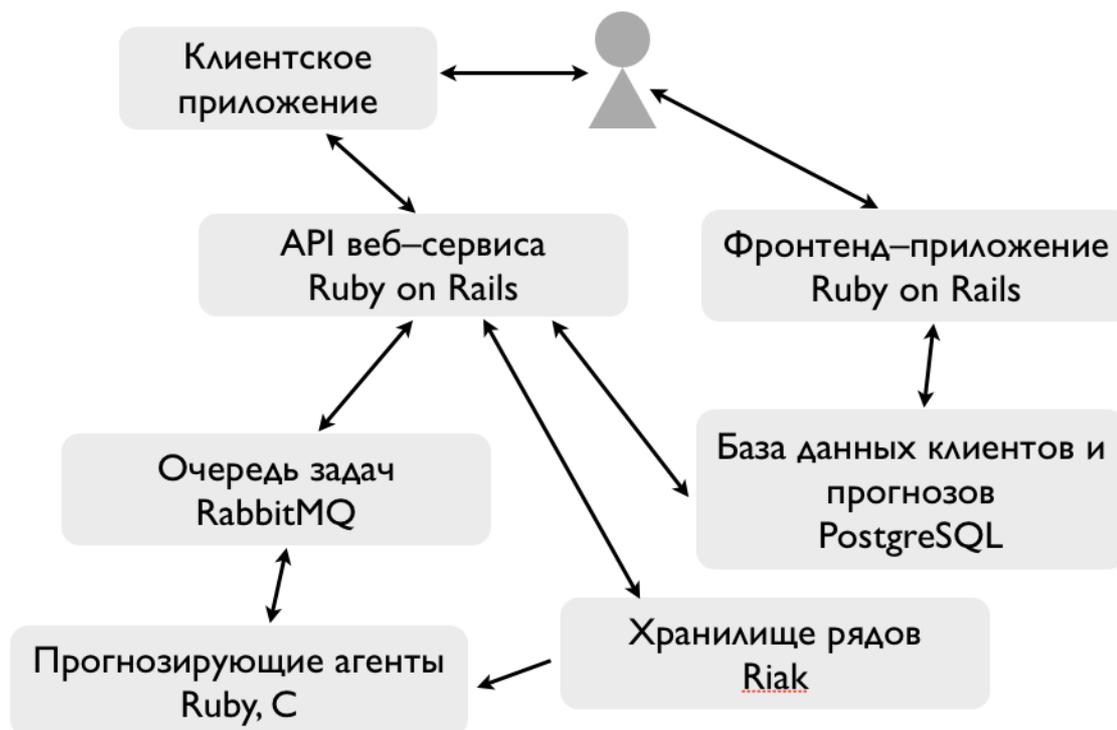


Рисунок 4. Архитектура разработанного решения.

- **Библиотека готовых решений**

Наличие фреймворков для веб-разработки, доступа к данным и передачи сообщений по популярным протоколам также является важным критерием, способствующим ускорению разработки и удешевлению поддержки разрабатываемой системы.

- **Возможность использования прототипа в качестве первой версии продукта**

Технологии, используемые для построения прототипа должны, тем не менее, позволять на его основе быстро реализовать готовый к продаже продукт, способный развиваться независимыми слабо-связанными частями. Прототип должен обладать достаточной функциональностью для удовлетворения базовых потребностей клиентов в прогнозировании.

- **Возможность быстрого масштабирования сервиса**

При необходимости, разработанный сервис должен иметь возможность быстрого горизонтального масштабирования, путем

добавления новых прогнозирующих агентов, работающих независимо друг от друга.

Таким образом, для реализации фронтенд–приложения, а также веб–сервиса был выбран язык Ruby, в большей степени благодаря фреймворку для быстрой веб–разработки Ruby on Rails.

Также язык Ruby поддерживает удобный механизм FFI (foreign function interface), позволяющий простейшим образом использовать функции из C–библиотек в Ruby. Таким образом, критические в плане времени исполнения части приложения могут быть написаны на низкоуровневом C, в то время как управление этими частями и реализация протоколов взаимодействия компонентов могут быть осуществлены с помощью более высокоуровневого Ruby.

Реализация компонентов приложения

Фронтенд–приложение

Реализованное на Ruby on Rails веб–приложение для управления счетом. В прототипе не используется никакой вариант приема оплаты, на счет клиента просто заносится 10000 “прогнозов”. Также используется для получения пользователем уникального идентификационного токена, по которому на веб–сервисе для прогнозирования будет осуществляться авторизация клиентского приложения.

Веб–сервис для прогнозирования

Веб–сервис так же реализован на Ruby on Rails в виде REST API [27] (подразумевает использование протокола HTTP для передачи сообщений между сервером и клиентом). Для передачи данных

используется формат JSON. Авторизация осуществляется на основе HTTP basic access authentication – передачи авторизационного токена, полученного через фронтенд–приложение, в заголовок каждого HTTP–запроса к сервису.

Предполагается, что сервис обладает знанием только о двух сущностях: прогнозируемый ряд и прогноз (запрос на прогноз с некоторыми параметрами и результат). Таким образом, реализация связи каждого продукта и его ряда продаж на сервере ложится на плечи клиентского приложения.

Ниже дано описание API сервиса (HTTP метод, URI и параметры запроса), а также комментарии по его работе.

Посредством HTTP PUT запроса, сервис принимает от клиентского приложения ряд, по которому в дальнейшем могут строиться запросы и сохраняет его в виде файла в распределенной базе данных Riak [28], возвращая клиенту идентификатор ряда.

- **PUT /series**

В теле запроса передается файл со значениями ряда.

Пример ответа: {"series": {"id": "2jaolmzav53a3fj"}}

Используя идентификатор ряда, клиентское приложение создает запрос на прогноз, при этом со счета списываются деньги, в зависимости от параметров прогноза. Сервис добавляет задачу для прогнозирующих агентов в очередь задач.

- **PUT /series/2jaolmzav53a3fj/forecasts**

Параметрами метода являются:

- `period` – период прогноза (`month`, `week`, `day`),
- `amount` – количество прогнозируемых значений,

- `external` – массив используемых для прогноза внешних факторов (сейчас поддерживается только температура: `temp`).

Пример ответа: `{"forecast": {"id": "f93nair30nf7vt", "status": "enqueued"}}`

Затем, используя идентификатор конкретного прогноза, клиентское приложение может запросить его статус и результат.

- **GET** `/series/2jaolmzav53a3fj/forecasts/f93nair30nf7vt`

Примеры ответов:

- Для прогноза в работе – `{"forecast": {"id": "f93nair30nf7vt", "status": "inprogress"}}`
- Для готового прогноза – `{"forecast": {"id": "f93nair30nf7vt", "status": "done", "result": [250, 267, 258]}}`

Прогнозирующие агенты

Прогнозирующие агенты – написанные на Ruby демоны, принимают задачи для прогнозирования из очереди, скачивают ряд из хранилища и выполняют прогнозирование. Основные вычислительно–сложные части алгоритма прогнозирования написаны на C и объединены Ruby–интерфейсом.

Заключение

В результате выполнения данной работы достигнуты следующие результаты:

- Проведен анализ предметной области, существующих программных решений и методов для прогнозирования продаж, выявлены их общие недостатки.
- Проанализированы и описаны особенности прогнозирования продаж, а также требования к методу прогнозирования.
- Разработан метод, удовлетворяющий описанным требованиям, который превзошел доступные в открытом доступе классические и современные методы прогнозирования для данной предметной области.
- Спроектирован и разработан прототип веб-сервиса и SaaS платформы для прогнозирования продаж.

Направления развития

К двум основным направлениям развития приложения относятся:

- Оптимизация метода прогнозирования: как нейронная сеть, так и метод “Гусеница” являются хорошо распараллеливаемыми задачами, чем можно воспользоваться для ускорения построения прогнозов на кластерах из нескольких машин.
- Оптимизация работы прогнозирующего кластера: для экономии ресурсов, а также обеспечения скорости для трудоемких прогнозов, может быть использован кластер с подключаемыми *по запросу (через специальный API)* виртуальными машинами. Такую услугу предоставляют большинство облачных вычислительных сервисов, например, Amazon AWS и RackSpace.

Список использованной литературы

1. Identifying problems in forecasting consumer demand in the fast moving consumer goods sector. Adebajo Dotun, Mann Robi. Benchmarking: An International Journal, Volume 7, Number 3, 2000 , pp. 223-230(8)
2. Алгоритм прогнозирования объёма продаж в MS Excel. [http://www.cfin.ru/finanalysis/sales_forecast.shtml]
3. Lokad Sales Forecasting [<http://www.lokad.com/>]
4. Forecast4AC PRO [<http://www.4analytics.ru/>]
5. KonSi-Forexsal [http://www.marketing-soft.ru/forecast/forexsal_1.html]
6. ForecastX [http://www.johngalt.com/galt_university/statistical_methods.shtml]
7. M3 Forecasting Competition Results [http://www.johngalt.com/galt_university/m3article.shtml]
8. Forecast Pro [<http://forecastpro.ru/>]
9. Goods4Cast (Forecsys Demand Management Solution) [<http://www.forecsys.ru/ru/site/products/Goods4Cast/>]
10. Delphus' Virtual Forecasting™ SaaS solution [<http://www.delphus.com/>]

11. Методы прогнозирования объема продаж. Бушуева Л.И.
Журнал "Маркетинг в России и за рубежом" 2004-10 [<http://www.ippnou.ru/print/000511/>]
12. Sales Forecasting Using Neural Networks. Frank M. Thiesing, Oliver Vornberger. Proceedings ICNN'97, 9–12 June 1997, Vol. 4, pp. 2125–2128, IEEE, 1997
13. The Yule Walker Equations for the AR Coefficients. Gidon Eshel. [http://www.stat.sc.edu/~vesselin/STAT520_YW.pdf]
14. ARIMA model for forecasting oil palm price. Rangan Nochai, Titida Nochai. Proceedings of the 2nd IMT–GT Regional Conference on Mathematics, Statistics and Applications Malaysia, June 13–15, 2006.
15. A MULTIVARIATE ARIMA MODEL TO FORECAST AIR TRANSPORT DEMAND. Alberto Andreoni, Maria Nadia Postorino. Mediterranea University of Reggio Calabria - Engineering Faculty. Association for European Transport and contributors 2006. [http://stuff.mit.edu/afs/athena/course/11/11.951/oldstuff/albacete/Other_Documents/Europe%20Transport%20Conference/applied_methods_in_tra/a_multivariate_ari1399.pdf]
16. Makridakis S., Wheelwright S. C., Forecasting Methods for Management, John Wiley & Sons, p. 86, 1989
17. Time series sales forecasting for short shelf-life food products based on artificial neural networks and evolutionary computing. Philip Doganis, Alex Alexandridis, Panagiotis Patrinos, Haralambos Sarimveis. Journal of Food Engineering 75 (2006) 196–204

18. A Fuzzy Clustering Model of Data and Fuzzy c-Means. S. Nascimento, B. Mirkin and F. Moura-Pires. Department of Computer Science, Birkbeck College, London, UK. [<http://sbisc.sharif.edu/~miap/Files/FuzzIEEE00.pdf>]
19. A hybrid model coupled with singular spectrum analysis for daily rainfall prediction. K. W. Chau and C. L. Wu. IWA Publishing 2010 Journal of Hydroinformatics, 12.4, 2010
20. Accurate Intermittent Demand Forecasting for Inventory Planning: New Technologies and Dramatic Results. By Charles N. Smart. [http://smartcorp.com/pdf/Intermittent_Demand_Forecasting_WhitePaper.pdf]
21. Динамически адаптируемые композиции алгоритмов прогнозирования. К. В. Воронцов, Е. В. Егорова. Работа при поддержке Российского Фонда Фундаментальных Исследований, проекты 04-01-08063-офи, 05-01-00877. [<http://www.ccas.ru/frc/papers/students/VoronEgorova05mmro.pdf>]
22. Gene Expression Programming: A New Adaptive Algorithm for Solving Problems. Cândida Ferreira. Complex Systems, Vol. 13, issue 2: 87-129, 2001 [<http://www.gene-expression-programming.com/webpapers/GEP.pdf>]
23. Метод “Гусеница”–SSA: анализ временных рядов. Н. Э. Голяндина. Учебное пособие. СПб., 2004. 76 с.
24. Метод «Гусеница»-SSA: прогноз временных рядов: Учеб. пособие. СПб: Изд-во СПбГУ, 2004. 52 с. Приложение В.

25. Combining Singular–Spectrum Analysis and neural networks for time series forecasting. F. Lisi, O. Nicolis, Marco Sandri. *Neural Processing Letters*, Vol. 2, No. 4, 6–10, 1995
26. Google’s MapReduce Programming Model — Revisited. Ralf Lammel, Microsoft Corp. [http://web.cs.wpi.edu/~cs3013/a11/Papers/Lammel_MapReduce_Revisited.pdf]
27. Architectural Styles and the Design of Network-based Software Architectures. DISSERTATION by Roy Thomas Fielding 2000. [<http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>]
28. Riak – highly scalable, fault-tolerant distributed database. [<http://basho.com/products/riak-overview/>]