

Санкт-Петербургский Государственный Университет
Математическое обеспечение и администрирование информационных
систем

Информационно-аналитические системы

Волжина Елена Григорьевна

Методы машинного обучения в задаче предсказания погоды

Курсовая работа

Научный руководитель:
доцент Михайлова Е. Г.

Санкт-Петербург
2017

SAINT-PETERSBURG STATE UNIVERSITY
Software and Administration of Information Systems

Analytical Information Systems

Elena Volzhina

Machine learning methods in weather prediction problem

Course Work

Scientific supervisor:
associate professor Elena Mikhaylova

Saint-Petersburg
2017

Оглавление

Введение	4
1. Существующие решения	5
1.1. Классические методы	5
1.2. Машинное обучение, технология Метеум	6
2. Методология	8
2.1. Данные для обучения	8
2.2. Решающие деревья и градиентный бустинг	10
2.3. Метрики	10
3. Эксперименты	12
3.1. Сбор обучающей выборки	12
3.2. Удалённость времени прогноза	12
3.3. Расстояние до ближайших станций	13
Заключение	14
Список литературы	15

Введение

С ростом производительности вычислительных систем, объемов накопленных данных об окружающем мире и опыта в их обработке находятся всё новые и новые области для применения анализа данных. Одной из таких областей сегодня является и прогнозирование погодных условий и в целом состояния атмосферы.

Прогнозы погоды полезны как обычным людям для принятия бытовых решений, так и в более серьезных областях: в авиации, судоходстве, а также в сельском хозяйстве. Большую пользу знания о погоде в будущем могут принести и бизнесу, в особенности сезонному, спрос в котором сильно зависит от погоды.

В Яндексе для прогноза погоды используется технология Метеум, комбинирующая прогнозы от нескольких поставщиков и другие данные с помощью методов машинного обучения, а именно градиентного бустинга над решающими деревьями. Модель, полученная таким образом, улучшает качество прогноза относительно каждого из поставщиков.

В рамках этой работы я добавляла к используемым моделью данным информацию о погоде в интересующей нас области в тот момент, когда мы вычисляем свой прогноз. Это может улучшить предсказания для ближайшего будущего, так как оно естественным образом зависит от настоящего.

1. Существующие решения

Наблюдения за погодой и попытки её предсказывать ведутся с тем или иным успехом уже несколько веков. Изначально прогнозы делались на основе опыта наблюдений и примет, но уже в XIX веке с развитием гидродинамики и термодинамики появляются первые математические инструменты для прогнозов.

В начале XX века Льюис Фрай Ричардсон составил систему уравнений, описывающих процессы, по которым меняется состояние атмосферы. Решив эту систему при текущих начальных условиях методом конечных разностей можно было получить прогноз изменения атмосферного давления[3]. Во второй половине XX века появляются всё новые сведения об атмосферных процессах, уточняющие точность моделирования, а также вычислительные ресурсы, необходимые для регулярного решения подобных систем в разумных временных рамках.

1.1. Классические методы

Классический подход к задаче прогноза погоды включает в себя несколько компонент. В основе всего лежит сбор данных: как долгосрочные наблюдения о температуре, осадках и прочих параметрах в конкретной местности, так и регулярные замеры с помощью разнообразной техники: приборов, установленных на наземных метеостанциях, метеорологических зондов и даже метеорологических спутников. Далее в дело вступают математические модели атмосферы, настроенные с помощью собранных данных. В этих моделях уравнения аэродинамики, гидродинамики и термодинамики описывают процессы в атмосфере, которые влияют на интересующие нас величины, а подставив в качестве параметров реальные данные о состоянии атмосферы в данный момент мы можем получить прогноз её состояния через некоторое время.

Модели атмосферы можно разделить на *глобальные*, покрывающие всю планету, и *локальные*, описывающие ограниченную территорию. Первые более универсальны, зато вторые могут давать лучшее качество

для своих областей за счет более тонкой настройки и высокого расширения. Сами прогнозы можно разделить на группы по времени, на которое мы смотрим в будущее: *краткосрочные* (до 36 часов), *среднесрочные* (от 36 часов до 10 суток) и *долгосрочные* (более 10 суток).

Построение и обновление математических моделей атмосферы очень трудозатратно и требует проведения большого количества различных экспериментов, поэтому из-за постоянных изменений климата все использующиеся модели являются в большей или меньшей степени устаревшими. Помимо этого, так как физика многих процессов в атмосфере еще недостаточно изучена, у всех моделей есть те или иные погрешности в прогнозировании, при этом зачастую эти погрешности имеют постоянную природу: например, одна из моделей может завышать температуру, а другая – занижать давление в горах.

Учитывая объемы информации, человеку сложно найти эти шаблоны в ошибках моделей. Но благодаря современным статистическим методам и вычислительным мощностям этот процесс можно автоматизировать. Именно эта идея лежит в основе технологии Метеум, которая используется для прогноза погоды в Яндексe. По накопленным данным о прогнозах разных моделей и фактической погоде в моменты, на которые делались эти прогнозы, можно построить модель, корректирующую прогнозы поставщиков и улучшающую качество итогового прогноза.

1.2. Машинное обучение, технология Метеум

Сервис Яндекс.Погода использует для прогноза погоды технологию Метеум. На основе накопленных данных составляются отдельные модели машинного обучения для интересующих нас параметров: температуры, давления, скорости и направления ветра, а также типа облачности и осадков. Важно уточнить, что температура предсказывается не сама по себе, а как разница между температурой и климатическими данными – усредненными показаниями за десятки

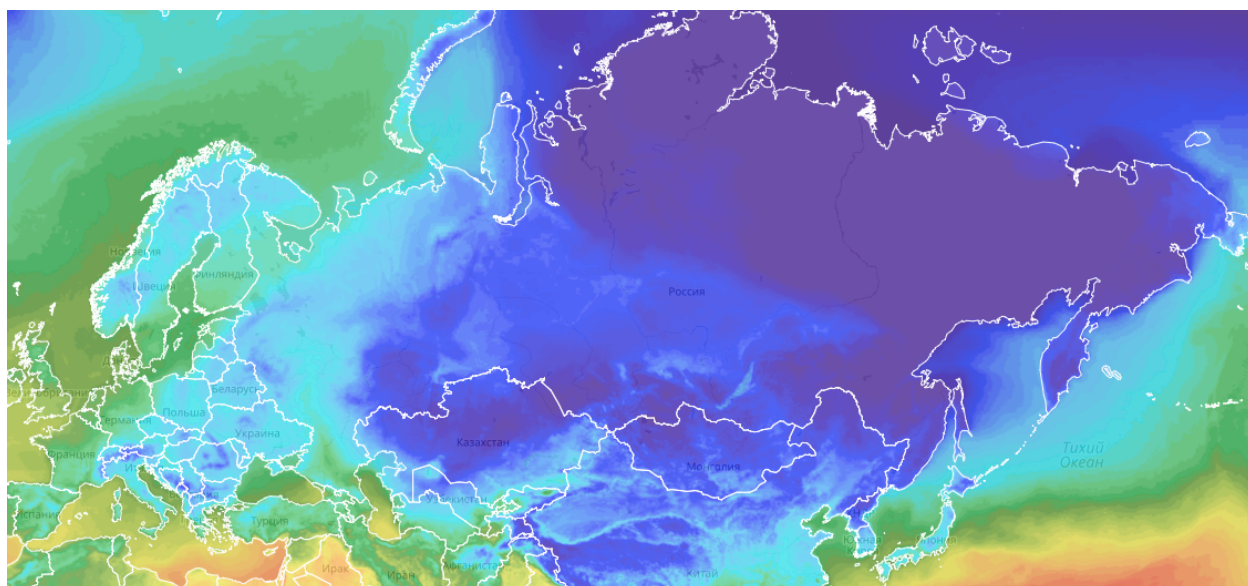


Рис. 1: Карта температуры

лет для этого времени года. Это позволяет частично избавиться от сезонности в изменениях температуры.

Для обучения используются данные, вычисляемые через классические модели прогнозирования. Сравнивая эти прогнозы с фактическими показаниями, модель находит зависимости целевой переменной от прогнозов разных показателей (например, для осадков кроме, собственно, прогнозов осадков, важными факторами оказываются также прогнозы облачности), а также учится исправлять упомянутые выше погрешности разных моделей.

В рамках данной работы будет проверена гипотеза о том, что полезной информацией для предсказаний погоды в ближайшем будущем окажется состояние погоды на данный момент. Для этого будут использоваться показания с одной или двух ближайших метеорологических станций.

2. Методология

Прежде чем приступать к экспериментам, следует разобраться с данными, которыми мы располагаем, решить, как мы будем собирать обучающую выборку с известными ответами и готовить её к использованию. Далее нужно выбрать метод для поиска решающей функции, которая бы приближала правильные ответы, находя зависимости в данных. Также необходимо определить метрики, с помощью которых будет приниматься решение, успешным считать эксперимент или нет. После этого можно будет приступать к основной задаче: добавить к обучающей выборке данные о текущих погодных условиях в месте, для которого рассчитывается прогноз; попробовать обучить на расширенных данных модель для предсказания температуры; оценить по метрикам улучшение.

2.1. Данные для обучения

При составлении прогноза мы располагаем множеством данных, полученных от поставщиков или вычисленных самостоятельно. Можно выделить несколько групп этих данных:

1. *прогнозы погоды*, сделанные при помощи математического моделирования физических процессов в атмосфере; в числе поставщиков следующие глобальные модели: Global Forecast System (GFS), Japan Meteorological Agency (JMA), European Centre for Medium-Range Weather Forecasts (ECMWF), Canadian Meteorological Centre (CMC). Также на серверах Яндекс.Погоды рассчитывается локальная модель для собственных прогнозов – Weather Research and Forecasting Model (WRF). У разных моделей могут различаться частота расчётов, координатная сетка, а также набор параметров в прогнозе, это необходимо учитывать при сборе обучающей выборки;
2. *фактические данные* о погоде, снятые на метеостанциях по всему миру с помощью статических приборов и метеозондов.

Используются станции из сети Всемирной метеорологической организации;

3. *климатические данные*, посчитанные на основе десятков лет наблюдений метеостанций;

4. *радарные данные* об осадках и облачности, *спутниковые снимки*.

Чтобы применять на этих данных какие-либо алгоритмы машинного обучения, нужно очистить их от выбросов и шума, привести всех поставщиков к единой координатной сетке и собрать обучающую выборку. В момент, когда мы делаем прогноз, мы можем использовать полученные ранее прогнозы от поставщиков, статистические данные о климате, а также информацию о погоде в интересующей нас точке в данный момент (или, с учётом задержек, в недавнем прошлом).

Нам потребуется множество наблюдений $X = \{x_i = (f_i^1, \dots, f_i^M), i \in \overline{1..N}\}$, элементы которого соответствуют векторам признаков, полезных для предсказания в текущий момент *gentime* состояния погоды на момент в будущем *time*. Например, в числе этих признаков будут прогнозы различных погодных параметров: температуры, влажности, скорости и направления ветра, осадков и облачности. Эти прогнозы должны быть сделаны на время, близкое к интересующему нас *time*, при этом при сборе обучающей выборки важно не заглядывать в будущее, то есть время генерации этих прогнозов поставщиками должно быть не позже, чем *gentime* (а с учётом задержек при передаче данных лучше брать прогнозы, сделанные еще раньше). Помимо множества наблюдений X нам понадобится множество истинных значений $y = \{y_i, i \in \overline{1..N}\}$, в нашем случае это разница температуры, которая была получена с метеостанции в момент *time* (на это время мы делали прогноз), с усредненной за много лет температурой в этот день и в это время: $temperature_delta = fact_temperature - climate_temperature$.

2.2. Решающие деревья и градиентный бустинг

Для проведения экспериментов по прогнозу температуры на основе имеющихся данных, был выбран алгоритм *градиентного бустинга над решающими деревьями*. Использовалась реализация этого алгоритма под названием Матрикснет, разработанная и используемая в компании Яндекс.

Решающее дерево – это алгоритм предсказания, описывающийся бинарным деревом, у которого каждой внутренней вершине v поставлен в соответствие некоторый предикат $\beta_v : X \rightarrow \{0, 1\}$, а каждому листу соответствует метка с ответом алгоритма[1]. После обучения применяется дерево следующим образом: для фиксированного $x \in X$ начинаем с корневой вершины (v_{root}). Вычисляем значение $\beta_{v_{root}}(x)$. Если получили 0, переходим в левого потомка, если 1 – в правого. Продолжаем до тех пор, пока не окажемся в листе, метку которого и возвращаем в качестве ответа.

Градиентный бустинг – алгоритм, с помощью которого можно по обучающей выборке найти композицию решающих деревьев, обладающую всякими хорошими свойствами [2] [4]. Я напишу про это позже.

2.3. Метрики

Для ответа на вопрос, улучшают изменения прогноз или ухудшают, необходимо выбрать метрики, с помощью которых в дальнейшем можно будет сравнивать качество разных моделей.

Стандартной метрикой в задаче регрессии (предсказания действительного числа) является *квадратный корень из среднеквадратичной ошибки* (Root Mean Square Error, RMSE). Он показывает, как сильно в среднем прогнозы модели отличаются от фактических данных. Для столбца фактических показаний y и столбца ответов модели \hat{y} значение этой метрики вычисляется следующим

образом:

$$RMSE(y, \hat{y}) = \sqrt{\sum_{i=1}^n (y_i - \hat{y}_i)^2}$$

Помимо усредненной ошибки нам также хотелось бы понимать, как часто формула ошибается достаточно сильно для того, чтобы вызвать недоумение пользователя сервиса. В качестве порогового значения было выбрано отклонение прогноза от фактической температуры на 5 градусов. Получаем метрику

$$Errors_5(y, \hat{y}) = \sum_{i=1}^n [|y_i - \hat{y}_i| > 5],$$

где $[x] = 1$, если x верно, иначе 0.

Чтобы значения метрик объективно отражали качество моделей, нужно считать их на новых для модели примерах. В нашем случае достаточно разделить обучающую выборку на два непересекающихся подмножества (X_{train} и X_{test}): наблюдения до определенной даты и после неё, на первом подмножестве обучить модель, а на втором вычислить метрики. Так мы не только будем проверять модель на примерах, которые не встречались ей при обучении, но и не дадим ей подглядывать в будущее (это плохо, так как погоду в последовательные моменты времени нельзя считать независимой).

3. Эксперименты

Данные о текущих погодных условиях могут помочь предсказать погоду в будущем только на небольшой срок. Для экспериментов было выбрано ограничение в 7 часов, и при более дальних прогнозах данные с ближайших станций не использовались.

3.1. Сбор обучающей выборки

К уже существующей обучающей выборке, собранной из векторов прогнозов поставщиков в качестве множества наблюдений ($X = \{x_i = (f_i^1, \dots, f_i^M), i \in \overline{1..N}\}$) и реальных *temperature_delta* в качестве верных ответов ($y = \{y_i, i \in \overline{1..N}\}$), нужно присоединить данные о фактической погоде в момент генерации прогноза. Для этого было построено следующее соответствие: для каждой станции, с которой поступают фактические данные, вычислены две ближайшие станции в пределах 150 километров. Далее в обучающую выборку для конкретной станции попадают либо данные с этих двух ближайших, либо только с одной из них (а на место второй – данные с самой станции).

Необходимо при сборе данных для обучения учесть, что в условиях реального применения модели будут задержки в получении данных от поставщиков прогнозов и фактов. Таким образом, если для генерирующегося в момент *gentime* прогноза мы будем брать фактические данные, снятые на станции в тот же момент *gentime*, мы получим слишком позитивную оценку метрик формулы. В реальности задержки имеют порядок десятков минут, и при генерации прогноза на сервисе использоваться будут соответственно устаревшие данные.

3.2. Удалённость времени прогноза

[Определение горизонтов]

[Тут на каждый пункт будут картинки с метриками, feature strength фич с ближайших станций]

При наивной реализации для всех краткосрочных и среднесрочных горизонтов получили небольшое улучшение метрик на ранних горизонтах, в которых присутствовали сведения о погоде с ближайших станций, но также и ухудшение на более отдаленных горизонтах.

Решили обучать только на горизонтах ≤ 7 , улучшение такое-то.

Поэкспериментировали с числом горизонтов, на 3-4 метрики улучшаются сильнее, но при просмотре формулы на карте обнаружили пятна.

[Тут будет картинка с пятнами на карте]

3.3. Расстояние до ближайших станций

[Тут на каждый пункт будут картинки с метриками, feature strength фич с ближайших станций]

Добавили фичи расстояния, взвешенную сумму, поменяли радиус применения: пятен нет, по метрикам прогноз лучше.

Заключение

При работе над этой задачей было показано, что данные с ближайших станций могут помочь улучшить прогноз температуры. Использовать их нужно с некоторым сглаживанием, чтобы не получить ярко выраженных окрестностей вокруг станций.

В качестве возможных направлений развития этой задачи перечислю следующие:

- использовать ближайшие станции не только для *temperature_delta*, но и для других показателей;
- поэкспериментировать с параметрами градиентного бустинга;
- попробовать другие способы сглаживания показаний со станций по мере удаления точки прогноза от них.

[Тут будет картинка какого-нибудь эксперимента с параметрами матрикснета, дающими улучшение]

Список литературы

- [1] Bishop Christopher M. Pattern Recognition and Machine Learning (Information Science and Statistics). — Secaucus, NJ, USA : Springer-Verlag New York, Inc., 2006.
- [2] Friedman Jerome H. Greedy function approximation: a gradient boosting machine // Annals of statistics. — 2001. — P. 1189–1232.
- [3] Lynch Peter. The origins of computer weather prediction and climate modeling // Journal of Computational Physics. — 2008. — Vol. 227, no. 7. — P. 3431–3444.
- [4] Stochastic gradient boosting // Computational Statistics & Data Analysis. — 2002. — Vol. 38, no. 4. — P. 367 – 378. — Nonlinear Methods and Data Mining.