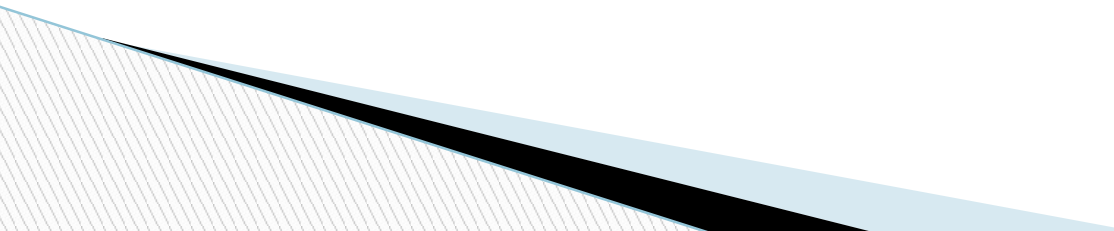


Распознавание образов

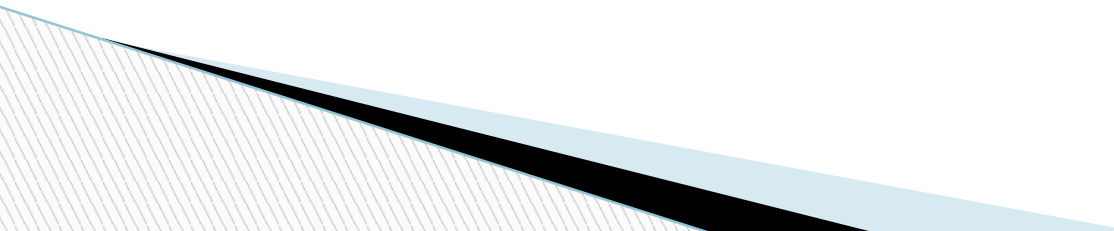


Распознавание образов

- ▣ раздел кибернетики, развивающий теоретические основы и методы классификации и идентификации предметов, явлений, процессов, сигналов, ситуаций и т. п. объектов, которые характеризуются конечным набором некоторых свойств и признаков.
 - ▣ Распознавание образов — это отнесение исходных данных к определенному классу с помощью выделения существенных признаков, характеризующих эти данные, из общей массы несущественных данных.
- 

- Имеется некоторый способ кодирования объектов, принадлежащих заранее известному конечному множеству классов $C = \{C_1, \dots, C_q\}$, и некоторое конечное множество объектов (обучающее множество), про каждый из которых известно, какому классу он принадлежит. Нужно построить алгоритм, который по любому входному объекту, не обязательно принадлежащему обучающему множеству, решает, какому классу этот объект принадлежит, и делает это достаточно хорошо. Качество распознавания оценивается как вероятность (т.е. частота) ошибки классификации на другом конечном множестве объектов с заранее известными ответами (тестовом множестве).

Примеры задач

- Распознавание текста
 - Распознавание лиц
 - Распознавание речи и звуков
 - Анализ сцен
 - Распознавание ситуаций
- 

Система распознавания

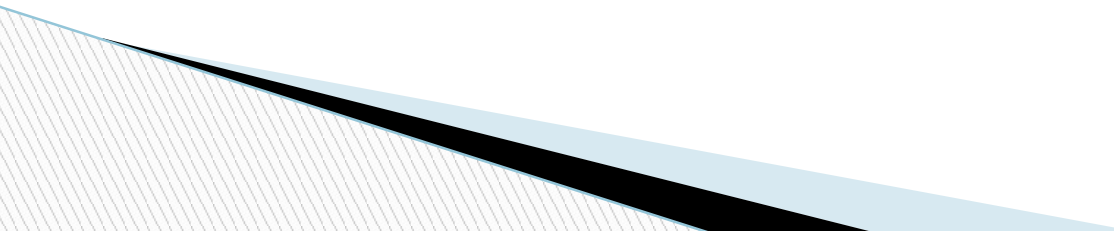
- Типичная система распознавания состоит из трех частей: извлечение признаков, собственно распознавание и принятие решения.
- Извлечение признаков - это преобразование входных объектов к единообразному, компактному и удобному виду с потерей подавляющей части содержащейся в объекте информации, слабо влияющей на классификацию. Удобным оказывается представление объекта точкой стандартного евклидова пространства \mathbf{R}^d , принадлежащей некоторому фиксированному компакту (кубу, шару, сфере, ...). Размерность d должна быть достаточно большой для успешного (в смысле качества) распознавания и достаточно малой для успешного (в смысле скорости) распознавания - реально это порядка нескольких десятков. Способ извлечения признаков зависит от природы и исходной кодировки объектов и подбирается вручную.

- Алгоритм распознавания строит отображение F из пространства признаков \mathbf{R}^d в единичный куб в пространстве \mathbf{R}^q . Желаемые значения F в точках пространства признаков, соответствующих обучающему множеству, известны, так что остается только построить в некотором смысле аппроксимирующее отображение. Качество аппроксимации будет проверяться не на всей области определения, а только на тестовом множестве.
- Интерпретацией вычисленных вероятностей занимается отдельная от распознавания процедура принятия решений, которая строится вручную и не зависит ни от природы входных объектов, ни от пространства признаков, ни от обучающих данных. Она зависит только от того, для чего эта система распознавания предназначена.

- Распознающий алгоритм - это вектор-функция двух векторных переменных $Y=F(W,X)$, где X - d -мерный вектор признаков, Y - q -мерный вектор вероятностей, а W - параметр. Как правило параметр тоже является вектором в евклидовом пространстве, причем очень многомерном (порядка тысяч), а функция F - непрерывна и даже дифференцируема.
- Обучение распознающей системы - это (как правило) подбор хорошего значения параметра W . В каком смысле и в какой степени хорошего - предмет многочисленных научных работ и экспериментов с системами распознавания.

- Обучение, в котором про обучающие вектора признаков известно, какому классу они принадлежат - называется обучением с учителем. Бывает еще обучение без учителя - когда имеется некоторое обучающее множество без знания о том, какой элемент принадлежит какому классу.

Методы распознавания

- Искусственные нейронные сети
 - Комитет рандомизированных решающих деревьев (Random Forest)
 - Машины опорных векторов (SVM)
 - Бустинг (AdaBoost)
 - Скрытые марковские модели (HMM)
- 

Random Forest

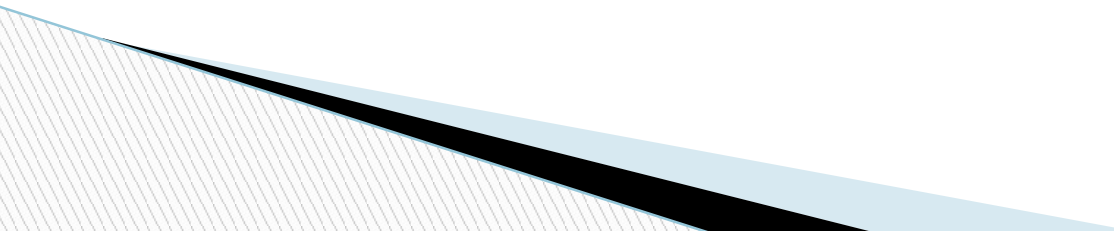
- Алгоритм применяется для задач классификации, регрессии и кластеризации.
- Заключается в использовании комитета (ансамбля) решающих деревьев

Алгоритм

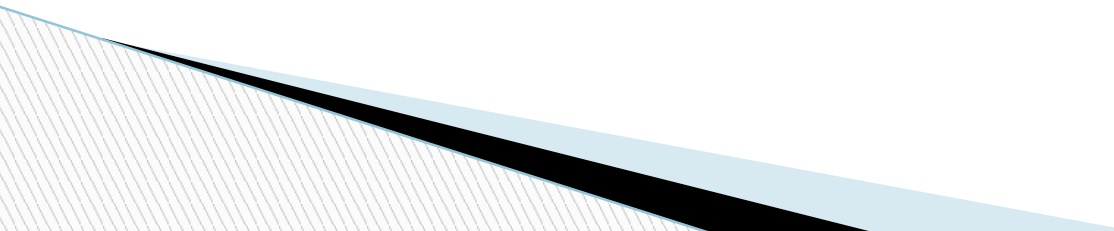
- Пусть обучающая выборка состоит из N примеров, размер пространства признаков равен M , и задан параметр m .
- Все деревья комитета строятся независимо друг от друга по следующей процедуре:
 1. Генерируем случайную подвыборку с повторением размером N из обучающей выборки. (Таким образом, некоторые примеры попадут в неё несколько раз, а примерно $N/3$ примеров не войдут в неё вообще)
 2. Построим решающее дерево, классифицирующее примеры данной подвыборки, причём в ходе создания очередного узла дерева будем выбирать признак, на основе которого производится разбиение, не из всех M признаков, а лишь из m случайно выбранных. Выбор наилучшего из этих m признаков может осуществляться различными способами. В оригинальном коде Бреймана используется критерий Гини, применяющийся также в алгоритме построения решающих деревьев CART. В некоторых реализациях алгоритма вместо него используется критерий прироста информации.
 3. Дерево строится до полного исчерпания подвыборки и не подвергается процедуре прунинга (в отличие от решающих деревьев, построенных по таким алгоритмам, как CART и ID3).

- Классификация объектов проводится путём голосования: каждое дерево комитета относит классифицируемый объект к одному из классов, и побеждает класс, за который проголосовало наибольшее число деревьев.
- Оптимальное число деревьев подбирается таким образом, чтобы минимизировать ошибку классификатора на тестовой выборке. В случае её отсутствия, минимизируется оценка ошибки out-of-bag: доля примеров обучающей выборки, неправильно классифицируемых комитетом, если не учитывать голоса деревьев на примерах, входящих в их собственную обучающую подвыборку.

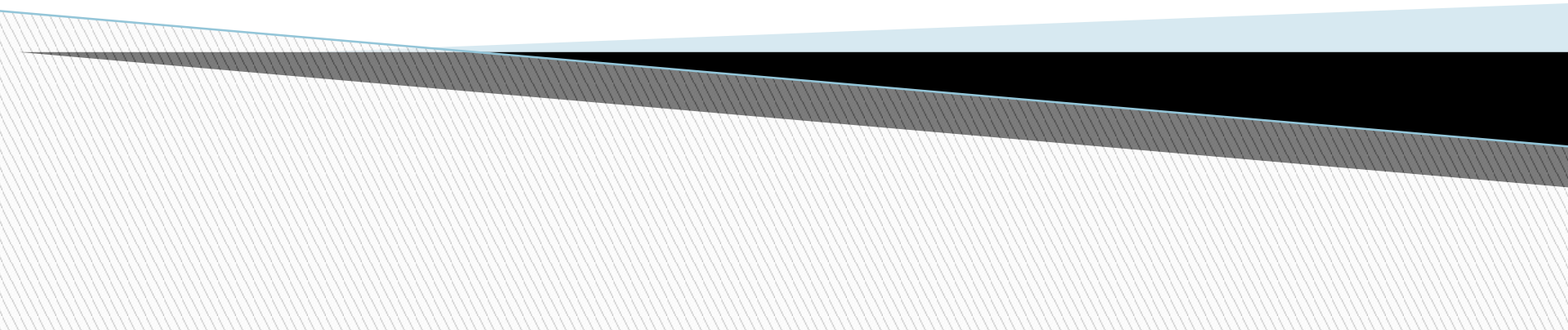
Преимущества

- Высокое качество получаемых моделей, сравнимое с SVM и бустингом, и лучшее, чем у нейронных сетей.
 - Способность эффективно обрабатывать данные с большим числом признаков и классов.
 - Нечувствительность к масштабированию (и вообще к любым монотонным преобразованиям) значений признаков.
 - Одинаково хорошо обрабатываются как непрерывные, так и дискретные признаки. Существуют методы построения деревьев по данным с пропущенными значениями признаков.
 - Существует методы оценивания значимости отдельных признаков в модели.
 - Внутренняя оценка способности модели к обобщению (тест out-of-bag).
 - Высокая параллелизуемость и масштабируемость.
- 

Недостатки

- ▣ Алгоритм склонен к переобучению на некоторых задачах, особенно на зашумленных задачах.
 - ▣ Большой размер получающихся моделей.
- 

Бустинг



- Усиление простых классификаторов - подход к решению задачи классификации (распознавания), путём комбинирования примитивных <слабых> классификаторов в один <сильный>. Под <силой> классификатора в данном случае подразумевается эффективность (качество) решения задачи классификации.
- В основе метода усиления простых классификаторов лежит простая предпосылка: скомбинировать некоторое количество элементарных (простых) признаков, таким образом, чтобы получить один, но более мощный.

Пример

- Пускай человек, играющий на скачках, решил создать программу, которая бы предсказывала, придёт ли интересующая его лошадь первой к финишу.
- Опросив некоторое количество играющих людей, он смог определить несколько эмпирических правил: ставь на лошадь, которая победила в трёх предыдущих заездах, ставь на лошадь, ставки на которую максимальны и т.д.
- Ясно, что каждое из таких правил по отдельности недостаточно надёжно и встает вопрос можно ли их оптимально скомбинировать для получения надёжных результатов.

AdaBoost

- Относится к классу статических ассоциативных машин
- Требуется построить классифицирующую функцию $F: X \rightarrow Y$, где X - пространство векторов признаков, Y - пространство меток классов.
Пусть в нашем распоряжении имеется обучающая выборка $(x_1, y_1), \dots, (x_N, y_N)$.
- Также у нас есть семейство простых классифицирующих функций $H: X \rightarrow Y$.

Алгоритм Discrete AdaBoost

- 1. Пусть $(x_1, y_1), \dots, (x_m, y_m)$ – начальная обучающая выборка и $D_0(i) = 1/m$ – начальное распределение для каждого i
- 2. Для каждого шага $t = 1, 2, \dots, T$:
 - а. Выбираем наилучший на текущем распределении $D_t(i)$ слабый классификатор $h_t \in H$:
 - $h_t = \max_{h_t \in H} |0.5 - (e_t = \sum_{i=1}^m D_t(i) I(h_t(x_i) \neq y_i))|$
 - б. Вычисляем коэффициент α_t
 - $\alpha_t = \frac{1}{2} \ln \frac{1-e_t}{e_t}$

- ◦ с. Запоминаем $f_t = \alpha_t h_t(x)$ и обновляем распределение
 - $D_{t+1}(i) = \frac{D_t(i) \exp(-y_i f_t(x_i))}{Z_i}$
 - где Z_i – нормализующий коэффициент, такой что
 - $\sum_{i=1}^m D_{t+1}(i) = 1$
- ▶ 3. Составляем комитет (сильный классификатор) следующим образом:
- ▶ $F(x) = \text{sign}(\sum_{t=1}^T f_t(x))$

- AdaBoost адаптивно настраивается на ошибки слабых классификаторов
- Верность классификации – 91,72%, при этом:
 - 1 эксперт 75,15%
 - 2 эксперт 71,44%
 - 3 эксперт 68,90%