|  |  |  |  |
| --- | --- | --- | --- |
| Министерство образования Республики Беларусь  Учреждение образования  Белорусский Государственный Университет Информатики и Радиоэлектроники | | | |
| Факультет | Компьютерных сетей и систем | | |
| Кафедра | Информатики  Дисциплина: Конструирование те технологии электронных вычислительных средств | | |
|  |  | | |
| **РЕФЕРАТ**  по курсу «Машинное обучение»  **Задача распознавания лиц** | | | |
| Студент:  гр. 758641  Ярош Г.И. | |  | Проверил:  Заливако С. С. |
| Минск, 2019 | | | |

СОДЕРЖАНИЕ

ДЕРЕВЬЯ РЕШЕНИЙ 3

1. Введение 3

2. Преимущества и недостатки деревьев решений 4

3. Процесс построения деревьев решений 5

Критерий расщепления 5

Проблема слишком ветвистых деревьев 5

Остановка построения дерева 6

Сокращение дерева или отсечение ветвей 6

4. Алгоритмы построения деревьев решений 6

Алгоритм ID3 7

Алгоритм С4.5 8

Алгоритм СART 8

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 10

# ЗАДАЧА РАСПОЗНАВАНИЯ ЛИЦ

## Введение

## Eigenfaces

Eigenfaces относится к подходам к решению задачи распознавания лиц, который основан на подсчете дисперсии коллекции изображений лиц и использует эту информацию для кодирования и сравнения изображений отдельных лиц в целом (в отличие от основанного на деталях или основанного на особенностях метода). В основе Eigenfaces лежит метод главных компонент, где главные компоненты распределения пикселей изображений лиц называются собственными лицами или, другими словами, собственными векторами [ковариационной](http://www.scholarpedia.org/article/Covariance) матрицы множества изображений лиц, где изображение *N* пикселей считается точкой (или вектором) в *N* мерном пространстве. Идея использования метода главных компонент для представления человеческих лиц была разработана Сировичем и Кирби (Sirovich and Kirby 1987) и использовалась Турком и Пентлендом (Turk and Pentland 1991) для обнаружения и распознавания лиц. Подход Eigenface, как полагают многие, является первой рабочей технологией распознавания лиц, и он послужил основой для одного из лучших коммерческих продуктов технологии распознавания лиц. Со времени его первоначальной разработки и публикации было создано много расширений оригинального метода и много новых разработок в системах автоматического распознавания лиц. Eigenfaces все еще часто рассматриваются как метод для сравнения по базовой линии, чтобы продемонстрировать минимальную ожидаемую эффективность такого рода систем.

### Описание алгоритма

Для корректной работы алгоритма изображения лица нормализуются для выравнивания глаз и рта, а затем все повторно дискретизируются с одинаковым разрешением в пикселях. Собственные лица затем извлекаются из данных изображения с помощью анализа главных компонентов (PCA) следующим образом:

1. Пусть дано *N* изображений лиц, каждое из которых представлено вектором размером *d*:
2. Для всех векторов подсчитываем среднее значение :
3. Вычисляем ковариационную матрицу :
4. Вычисляем собственные значения и собственные векторы ковариационной матрицы :
5. Выбираем собственных векторов, которые соответствуют наиболльшим собственным значениям.
6. Вычисление главных компонент вектора происходит по формуле:

### Распознавание изображений

Распознавание изображения с помощью Eigenface происходит следующим образом:

1. Для всех тренировочных изображений посчитываются собственные лица :
2. Для распознаваемого изображения лица также подсчитываются подсчитывается собственные лица :
3. В качестве критерия похожести распознаваемого изображения на изображение в тренировочной выборке можно использовать евклидово расстояние:

Если расстояние меньше изначально заданного порога, то можно рассматривать, что входное изображение распознано, как .

Подход Eigenface был важным шагом на пути к распознаванию лиц в компьютерном [зрении](http://www.scholarpedia.org/article/Vision) . Тем не менее, метод чувствителен к изменению освещения, масштаба, позы, выражения лица и окклюзии. Чтобы алгоритм работал хорошо, лицо должно быть представлено в виде спереди, в соответствующем масштабе, в аналогичном освещении, в определенном (обычно нейтральном) выражении и не включенным. Для решения множества проблем, с которыми сталкиваются реальные системы распознавания лиц, было предложено множество модификаций и расширений с момента первоначальной реализации метода.

## Fisherfaces

Метод главных компонент (PCA), который является основой метода Eigenfaces, находит линейную комбинацию функций, которая максимизирует общую дисперсию данных. Хотя это, несомненно, мощный способ представления данных, он не учитывает какие-либо классы, и поэтому при отбрасывании компонентов может быть потеряно много дискриминационной информации. Алгоритм Fisherface использует метод, называемый линейным дискриминантным Анализом. Метод выполняет уменьшение размерности с учетом класса объекта и был изобретен знаменитым математиком Р. А. Фишером. Он успешно использовал его для классификации цветов в своей работе 1936 года «Использование нескольких измерений в таксономических задачах». Чтобы найти комбинацию признаков, которая лучше всего разделяет классы, линейный дискриминантный анализ максимизирует соотношение между классами и разброс внутри классов, а не максимизирует общий разброс. Идея проста: одни и те же классы должны близко располагаться друг к другу, в то время как разные классы находятся как можно дальше друг от друга в пространстве более низкой размерности.

### Описание алгоритма

Пусть дано классов изображений . Алгоритм Fisherface заключается в следующем:

1. Различия внутри класса могут быть оценены с помошью матрицы рассеяния, вычисляемой следующим способом:
2. Похожим образом вычисляются межклассовые различия используя межклассовую матрицу рассеяния:

## Процесс построения деревьев решений

Алгоритмы конструирования деревьев решений состоят из этапов построения дерева (tree building) и сокращения дерева (tree pruning). В ходе построения дерева решаются вопросы выбора критерия расщепления и остановки обучения (если это предусмотрено алгоритмом). В ходе этапа сокращения дерева решается вопрос отсечения некоторых его ветвей.

### Критерий расщепления

Процесс создания дерева происходит сверху вниз, т.е. является нисходящим. В ходе процесса алгоритм должен найти такой критерий расщепления, иногда также называемый критерием разбиения, чтобы разбить множество на подмножества, которые бы ассоциировались с данным узлом проверки. Каждый узел проверки должен быть помечен определенным атрибутом. Существует правило выбора атрибута: он должен разбивать исходное множество данных таким образом, чтобы объекты подмножеств, получаемых в результате этого разбиения, являлись представителями одного класса или же были максимально приближены к такому разбиению. Последняя фраза означает, что количество объектов из других классов в каждом классе должно стремиться к минимуму.

Существуют различные критерии расщепления. Наиболее известные - мера энтропии, мера информационного выигрыша и неопределенность Gini.

В некоторых методах для выбора атрибута расщепления используется так называемая мера информативности подпространств атрибутов, которая основывается на энтропийном подходе и известна под названием "мера информационного выигрыша" (information gain) или мера энтропии.

Другой критерий расщепления реализован в алгоритме CART и называется неопределенность Gini. При помощи этого индекса атрибут выбирается на основании расстояний между распределениями классов.

### Проблема слишком ветвистых деревьев

Чем больше частных случаев описано в дереве решений, тем меньшее количество объектов попадает в каждый частный случай. Такие деревья называют "ветвистыми", они состоят из неоправданно большого числа узлов и ветвей, исходное множество разбивается на большое число подмножеств, состоящих из очень малого числа объектов. В результате "переполнения" таких деревьев их способность к обобщению уменьшается, и построенные модели не могут давать верные ответы.

В процессе построения дерева, чтобы его размеры не стали чрезмерно большими, используют специальные процедуры, которые позволяют создавать оптимальные деревья.

### Остановка построения дерева

Рассмотрим правило остановки. Оно должно определить, является ли рассматриваемый узел внутренним узлом, при этом он будет разбиваться дальше, или же он является конечным узлом, т.е. узлом решением.

Один из вариантов правил остановки - "ранняя остановка" (prepruning). Она определяет целесообразность разбиения узла. Преимущество использования такого варианта - уменьшение времени на обучение модели. Однако здесь возникает риск снижения точности классификации.

Второй вариант остановки обучения - ограничение глубины дерева. В этом случае построение заканчивается, если достигнута заданная глубина.

Еще один вариант остановки - задание минимального количества примеров, которые будут содержаться в конечных узлах дерева. При этом варианте ветвление продолжаются до того момента, пока все конечные узлы дерева не будут чистыми или будут содержать не более чем заданное число объектов.

### Сокращение дерева или отсечение ветвей

Решением проблемы слишком ветвистого дерева является его сокращение путем отсечения (pruning) некоторых ветвей.

Качество классификационной модели, построенной при помощи дерева решений, характеризуется двумя основными признаками: точностью распознавания и ошибкой.

Точность распознавания рассчитывается как отношение объектов, правильно классифицированных в процессе обучения, к общему количеству объектов набора данных, которые принимали участие в обучении.

Ошибка рассчитывается как отношение объектов, неправильно классифицированных в процессе обучения, к общему количеству объектов набора данных, которые принимали участие в обучении.

Отсечение ветвей или замену некоторых ветвей поддеревом следует проводить там, где эта процедура не приводит к возрастанию ошибки. Процесс проходит снизу вверх, т.е. является восходящим. Это более популярная процедура, чем использование правила остановки. Деревья, получаемые после отсечения некоторых ветвей, называют усеченными.

Если такое усеченное дерево все еще не является интуитивным и сложно для понимания, при визуализации используют извлечение правил, которые объединяют в наборы для описания классов. Каждый путь от корня дерева до его вершины или листа дает одно правило. Условиями правила являются проверки на внутренних узлах дерева.

## Алгоритмы построения деревьев решений

Алгоритмы построения деревьев решений различаются следующими характеристиками:

* вид расщепления - бинарное (binary), множественное (multi-way);
* критерии расщепления – информационная энтропия, мера информационного выйгрыша, индекс Gini;
* возможность обработки пропущенных значений;
* процедура сокращения ветвей или отсечения;
* возможности извлечения правил из деревьев.

Для построения деревьев решений используются следующие алгоритмы:

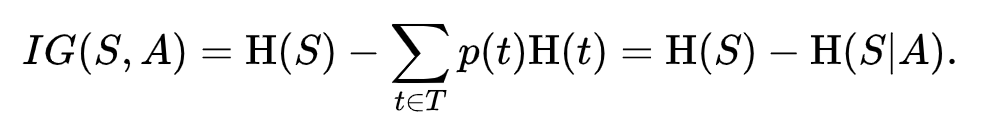
* [Алгоритм ID3](https://ru.wikipedia.org/wiki/ID3_(%D0%B0%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC)), где выбор атрибута происходит на основании прироста информации ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) [Gain](https://en.wikipedia.org/wiki/Information_gain_in_decision_trees)) либо минимизации информационной энтропии;
* [Алгоритм C4.5](https://ru.wikipedia.org/wiki/C4.5) (улучшенная версия ID3), где выбор атрибута происходит на основании нормализованного прироста информации;
* Алгоритм [CART](https://ru.wikipedia.org/wiki/CART_(%D0%B0%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC));
* Автоматический детектор взаимодействия Хи-квадрат (CHAID). Выполняет многоуровневое разделение при построении деревьев;
* MARS: расширяет деревья решений для улучшения обработки цифровых данных.

Далее будут рассмотрены некоторые из них.

### Алгоритм ID3

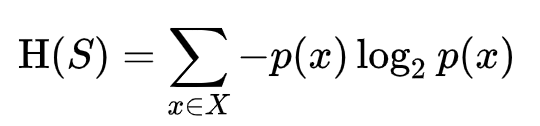
Алгоритм ID3 предложен в 1986 году Россом Квинланом. Алгоритм строит деревья решений со множественными путями, используя меру информационного выигрыша как критерий расщепления.

Мера информационного выигрыша определяет разницу информационной энтропии до и после разделения набора данных S по некоторому атрибуту A и определяется по формуле:



где H(S) – мера информационной энтропии, T – подмножества, на которое был разделен набор данных S, p(t) – соотношение количества элементов в множестве t относительно количества элементов во всем наборе данных S, H(t) – мера информационной энтропии подмножества t.

Мера информационной энтропии определяется следующей формулой:



где S – текущий набор данных, X – множество всех классов, p(x) – соотношение между количеством элементов класса x и всех элементов в наборе данных.

В начале работы алгоритм рассматривает весь набор данных S. На каждой итерации для каждого неиспользованного атрибута подсчитывается мера информационного выигрыша. Далее выбирается атрибут с наибольшим значением меры. Он используется для разделения набора данных на подмножества. Для каждого значения атрибута в текущем наборе данных создается подмножество и соответствующая ветвь дерева с условием. Далее алгоритм рекурсивно продолжает работу для каждого из подмножеств не учитывая атрибут, использованный на этом шаге.

Алгоритм для текущего множества останавливается в ряде случаев:

* Каждый элемент из текущего множества принадлежит к одному классу. Текущий узел дерева помечается этим классом.
* Больше нет атрибутов, по которым можно разделить множество. В данном случае узел дерева помечается классом, к которому принадлежит наибольшее количество элементов в множестве.
* Нет ни одного элемента в множестве. В таком случае текущий узел дерева помечается классом, наиболее представленным в родительском узле дерева.

В результате работы алгоритма получается дерево решений, где каждый внутренний узел определяет набор условий для конкретного атрибута, а внешние узлы определяют классы.

### Алгоритм С4.5

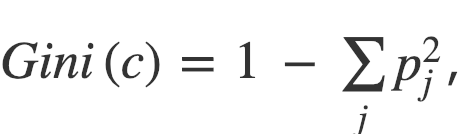
Алгоритм С4.5 является развитием алгоритма ID3. Алгоритм использует нормированную меру информационного выигрыша. Нормировка происходит по количеству всех возможных значений атрибута, для которого подсчитывается мера. Данное усовершенствование позволяет бороться с проблемой переобучения, когда дерево решений чаще использует атрибуты с большим количеством значений, создавая множество ветвей и запоминая данные из исходной выборки.

Также алгоритм C4.5 имеет следующие усовершенствования относительно ID3:

* Возможность обрабатывать непрерывные значения атрибутов. Для обработки непрерывных значений, алгоритм создает пороговое, по которому разделяет выборку на подмножества, где значения атрибута больше, равны, либо меньше порогового.
* Обработка пропущенных значений атрибутов. Пропущенные значения не используются при подсчете мер информационной энтропии и выйгрыша.
* Возможность настройки весов для атрибутов.
* Отсечение ветвей у построенного дерева. После создания дерева, алгоритм проходится по всем узлам дерева и удаляет ветви, которые не ухудшают эффективность классификации, заменяя их на узел с пометкой класса.

### Алгоритм СART

Алгоритм CART является алгоритмом построения двоичных деревьев классификации. Он использует неопределенность Gini как критерий расщепления:



где pj – соотношение количества элементов класса j и количества элементов в текущей выборке.

Неопределенность Gini показывает вероятность того, что элемент текущего набора данных будет неправильно классифицирован. Алгоритм стремится к минимизации данного значения для всех узлов дерева.

Алгоритм разбиения похож на алгоритм C4.5. На каждой итерации, для каждого атрибута вычисляется наилучшее разбиение его значений на два подмножества основываясь на минимальном значении неопределенности Gini для подмножеств. Выбирается атрибут с наилучшим разбиением.

Для численный атрибутов в узле дерева устанавливаются условия вида меньше либо равно. Для категориальных атрибутов устанавливаются условия принадлежности значения к некоторому множеству.

После того, как алгоритм построил дерево, происходит отсечение лишних ветвей основываясь на проведении кросс валидации между двумя вариантами дерева: с и без ветви.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

[1] Методы классификации и прогнозирования. Деревья решений [Электронный ресурс]. – Электронные данные. – ­Режим доступа: https://www.intuit.ru/studies/professional\_skill\_improvements/1210/courses/6/lecture/174?page=1.

[2] ID3 algorithm [Электронный ресурс]. – Электронные данные. – ­Режим доступа: https://en.wikipedia.org/wiki/ID3\_algorithm.

[3] C4.5 algorithm [Электронный ресурс]. – Электронные данные. – ­Режим доступа: https://en.wikipedia.org/wiki/C4.5\_algorithm.

[4] Sklearn. Decision Trees [Электронный ресурс]. – Электронные данные. – ­Режим доступа: https://scikit-learn.org/stable/modules/tree.html.

[5] Decision tree learning [Электронный ресурс]. – Электронные данные. – ­Режим доступа: https://en.wikipedia.org/wiki/Decision\_tree\_learning.