Классификация изображений и введение в машинное обучение

Антон Конушин

Slides adapted from Fei-Fei Li, Rob Fergus, Antonio Torralba, Jean Ponce and Svetlana Lazebnik

Ранее рассмотренный метод

Схема простого алгоритма выделения и распознавания объектов

Что было сложно и плохо?

- Можем работать только с контрастными объектами, которые довольно легко выделить
- Признаков может быть много (несколько моментов, площадь, положение, гистограммы яркости, каналов цвета и т.д.)
- Подбирать правила приходится вручную, много гипотез, когда признаков много и распределение сложно, это невозможно.

Как должно быть в идеале

Что такое машинное обучение?

- Обучение ≠ «заучивание наизусть»
  - Заучить наизусть – для машины не проблема
  - Мы хотим научить машину делать выводы!
  - Машина должна корректно работать на новых данных, которые мы ей раньше не давали
  - По конечному набору обучающих данных машина должна научиться делать выводы на новых данных

Машинное обучение

- Нейронные сети - Перспептрон (Розенблатт, 1958), когнитрон (Фукушима, 1975)
- Нейронные сети - обратное распространение ошибки (1980е)
- Метод опорных векторов (1990е)
- Бустинг (конец 1990х)
- Рандомизированный решающий лес (начало 2000х)
- Нейронные сети - «глубинное обучение» (2006 и далее)

Сегодня рассмотрим «на пальцах» один метод – «метод опорных векторов»

Задача классификации образов

- Дано множество объектов, каждый из которых принадлежит одному из нескольких классов. Нужно определить, какому классу принадлежит данный экземпляр
- Каждый объект с номером j можно описать вектором признаков $\boldsymbol{x}_j$
- Каждому объекту можно приписать метку класса $\boldsymbol{y}_j$
- Всё множество известных наблюдений (конечное) можно записать в следующем виде:

Каждое наблюдение (x,y) ещё называют прецедентом

Задача классификации образов

- Задача построить функцию **y=f(x)**, которая для каждого вектора-признаков **x** даёт ответ **y**, какому классу принадлежит объект **x**
- Функция **f()** назвается *решающее правило или классификатор*
- Любое *решающее правило* делит пространство признаков на *решающие регионы* разделенные *решающими границами*

Пространство признаков

Решающая граница

Классификация

- Будем выбирать функции f_t из **параметрического семейства** ***F*** (т.е. будем выбирать подходящий набор параметров ***t***)
- Введем некоторую **функцию потерь** $L(f_t(x,t), y)$,
  - В случае классификации часто используют $L(f(\mathbf{x}), y) = I[y \neq f(\mathbf{x})]$,где $f(\mathbf{x})$ - предсказанный класс
  - Можем использовать и другие функции потерь
- Задача обучения состоит в том, чтобы найти набор параметров **t** классификатора **f**, при котором потери для *новых* данных будут минимальны
- «Метод классификации» = параметрическое семейство F + алгоритм оценки параметров по обучающей выборке

Общий риск

- Общий риск – математическое ожидание потерь:

$$R(x) = E\big(L(f(x,t),y)\big) = \int\limits_{x,y} L(f(x,t),y)dP$$

- Наша задача – найти такие параметры t, при которых общий риск минимален
- Но общий риск рассчитать невозможно, поскольку распределение *P* для множества наших объектов (x,y) неизвестно

Эмпирический риск

- Дано $X^m = \{\mathbf{x_1}, ..., \mathbf{x_m}\}$ - обучающая выборка

- Эмпирический риск (ошибка обучения):

$$R_{emp}\left(f, X^m\right) = \frac{1}{m}\sum_{i=1}^{m} L(f(x_i), y_i)$$

- Метод минимизации эмпирического риска:

$$f = \arg\min_{f \in F} R_{emp}\left(f, X^m\right)$$

Смысл: подбираем параметры ***t*** решающего правила ***f*** таким образом, чтобы эмпирический риск $\boldsymbol{R_{emp}}$ был минимален

Линейная классификация

- Рассмотрим случай линейно разделимых данных
- Только для 2х классов
- Т.е. таких, что вектора в пространстве признаков можно отделить друг от друга гиперплоскостью

Линейный классификатор

- Найдем линейную функцию (гиперплоскость) и разделим положительные {y=+1} и отрицательные {y=-1} примеры

$$\mathbf{x}_i \; положительные: \quad \mathbf{x}_i \cdot \mathbf{w} + b \geq 0$$

$$\mathbf{x}_i \; отрицательные: \quad \mathbf{x}_i \cdot \mathbf{w} + b < 0$$

Для всех рассмотренных плоскостей эмпирический риск одинаковый

Какая гиперплоскость наилучшая?

Метод опорных векторов

- Найдем гиперплоскость, максимизирующую *отступ* между положительными и отрицательными примерами
- Support Vector Machine (SVM)

$\mathbf{x}_i$ *положительные* $(y_i = 1)$: $\mathbf{x}_i \cdot \mathbf{w} + b \geq 1$

$\mathbf{x}_i$ *отрицательные* $(y_i = -1)$: $\mathbf{x}_i \cdot \mathbf{w} + b \leq -1$

Для опорных векторов, $\mathbf{x}_i \cdot \mathbf{w} + b = \pm 1$

Расстояние от точки до гиперплоскости: $\dfrac{|\,\mathbf{x}_i \cdot \mathbf{w} + b\,|}{\|\,\mathbf{w}\,\|}$

Поэтому отступ равен $2 / \|\mathbf{w}\|$

http://www.machinelearning.ru/wiki/index.php?title=SVM

Поиск гиперплоскости

1. Максимизируем отступ $2/\|\mathbf{w}\|$
2. Правильно классифицируем все данные:

$$\mathbf{x}_i \textit{ позитивные } (y_i = 1): \qquad \mathbf{x}_i \cdot \mathbf{w} + b \geq 1$$

$$\mathbf{x}_i \textit{ негативные } (y_i = -1): \qquad \mathbf{x}_i \cdot \mathbf{w} + b \leq -1$$

- *Квадратичная оптимизационная задача*:

- Минимизируем $\frac{1}{2}\mathbf{w}^T\mathbf{w}$

  При условии $y_i(\mathbf{w}\cdot\boldsymbol{x}_i + b) \geq 1$

- Решается с помощью методом множителей Лагранжа

http://www.machinelearning.ru/wiki/index.php?title=SVM

Поиск гиперплоскости

- Решение: $\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i$

Обученные веса

Опорные вектора

- Для большей части векторов вес = 0!
- Все вектора, для которых вес >0 называются опорными
- Определяется только <u>опорными</u> векторами

http://www.machinelearning.ru/wiki/index.php?title=SVM

Поиск гиперплоскости

- Решение: $$\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i$$

  $b = y_i - \mathbf{w} \cdot \mathbf{x}_i$ для любого опорного вектора

- Решающая функция:

$$\mathbf{w} \cdot \mathbf{x} + b = \sum_i \alpha_i y_i \mathbf{x}_i \cdot \mathbf{x} + b$$

- Решающая функция зависит от скалярных произведений (inner product) от тестового вектора $\boldsymbol{x}$ и опорных векторов $\boldsymbol{x}_i$
- Решение оптимизационной задачи также требует вычисления скалярных произведений $\boldsymbol{x}_i \cdot \boldsymbol{x}_j$ между всеми парами векторов из обучающей выборки

http://www.machinelearning.ru/wiki/index.php?title=SVM

Реальный случай

Вводим дополнительные «slack» переменные: $\xi = (\xi_1, \ldots, \xi_n)^T$

Минимизируем $\frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{n}\xi_i$

При условии $y_i(wx_i + b) \geq 1 - \xi_i$

C – параметр регуляризации

В реальном случае идеально разделить данные невозможно (эмпирический риск всегда больше 0)

Нелинейные SVM

- На линейно разделимых данных SVM работает отлично:

- Но на более сложных данных не очень:

- Можно отобразить данные на пространство большей размерности и разделить их линейно там:

Slide credit: Andrew Moore

Нелинейные SVM

- **Идея:** отображение исходного пространства параметров на какое-то многомерное пространство признаков (feature space) где обучающая выборка линейно разделима:

Slide credit: Andrew Moore

Нелинейные SVM

- Вычисление скалярных произведений в многомерном пространстве вычислительно сложно
- *The kernel trick:* вместо прямого вычисления преобразования $\boldsymbol{\varphi}(\mathbf{x})$, мы определим ядровую функцию K:

$$K(\mathbf{x}_i, \mathbf{x}_{jj}) = \boldsymbol{\varphi}(\mathbf{x}_i) \cdot \boldsymbol{\varphi}(\mathbf{x}_j)$$

- Чтобы все было корретно, ядро должно удовлетворять условию Мерсера (*Mercer's condition*)
  - Матрица $K(x_i,x_j)$ должна быть неотрицательно определенной
- С помощью ядра мы сможем построить нелинейную решающую функцию в исходном пространстве:

$$\sum_i \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b$$

Пример ядра

- Полиномиальное: $K(x, y) = ((\mathbf{x}, y) + c)^d$

- Пусть d=2, x=(x_1,x_2):

$$K(x, y) = ((\mathbf{x}, y) + c)^2 = (x_1 y_1 + x_2 y_2 + c)^2 = \varphi(x) \cdot \varphi(y)$$

$$\varphi(x) = (x_1^2, x_2^2, \sqrt{2} x_1 x_2, \sqrt{2c} x_1, \sqrt{2c} x_2, c)$$

- Т.е. из 2х мерного в 6и мерное пространство

SVM - резюме

- Мощный и гибкий подход на основе ядер
  - Придумано много разных ядер
  - Линейный метод часто достаточен
- На практике работает очень хорошо, даже для маленьких обучающих выборок
- Много доступных библиотек: http://www.kernel-machines.org/software

Предсказательная способность

- Научились обучать классификатор SVM на обучающей выборке
- Нам нужно, чтобы классификатор хорошо работал на данных, которые не использовались при обучении
- «Предсказательная способность» - качество классификации вне обучающей выборки
- Как оценить *предсказательную способность*?

- Обучать на одной части известных данных, проверять на другой
  - «Удерживание»
  - «Скользящий контроль» (Cross-validation)

Удерживание

Будем разбивать имеющуюся обучающую выборку на части

- На одной части будем обучать классификатор (минимизировать эмпирический риск)
- На другой будем измерять ошибки обученного классификатора (оценивать предсказательную способность)

$$R(f, X) \sim P\left(f(x) \neq y \mid X^c\right) = \frac{1}{c}\sum_{j=1}^{c}\left[f(x_j) \neq y_j\right]$$

Свойства «удерживания»

- Быстро и просто рассчитывается
- Некоторые «сложные» прецеденты могут полностью попасть в только одну из выборок и тогда оценка ошибки будет смещенной

Ошибка произойдет не по вине классификатора, а из-за разбиения!

Скользящий контроль

- Разделим выборку на d непересекающихся частей и будем поочередно использовать одно из них для контроля а остальные для тренировки

  - Разбиваем: $\{X^i\}_1^d : X^i \cap X^j = 0, i \neq j$

    $$\bigcup_{i=1}^{d} X^i = X^k$$

  - Приближаем риск:

$$P(f(X^k) = y^*) \approx \frac{1}{d} \sum_{i=1}^{d} P(f(X^i) \neq y^* \mid \bigcup_{i \neq j} X^i)$$

Иллюстрация

Контроль

Обучение

Результат считается как средняя ошибка по всем итерациям

Свойства

- В пределе равен общему риску
- Каждый прецедент будет один раз присутствовать в контрольной выборке
- Обучающие выборки будут сильно перекрываться (чем больше сегментов, тем больше перекрытие)
  - Если одна группа «сложных прецедентов» попала полностью в один сегмент, то оценка будет смещенной
- Предельный вариант – “leave one out”
  - Обучаемся на всех элементах, кроме одного
  - Проверяем на одном
  - Повторяем для всех элементов

Ошибки I и II рода

- В наших задачах обычно классы объектов неравнозначны
  - Лицо человека – «основной» класс (положительные примеры)
  - Фон – «вторичный» класс (отрицательные примеры)

- **Ошибка первого рода** равна вероятности принять основной класс за вторичный
  - Вероятность «промаха», когда искомый объект будет пропущен (пропустить лицо)

- **Ошибка второго рода** равна вероятности принять вторичный класс за основной
  - Вероятность «ложной тревоги», когда за искомый объект будет принят «фон»

Ошибки I и II рода

Ошибка II рода
(ложные срабатывания)

Ошибка I рода
(пропущенные объекты)

Ошибки I и II рода

- При анализе изображений классы обычно несбалансированы
  - Объектов основного класса («лица») - мало
  - Объектов второго класса («фон») - много
- Особенно важно оценивать ошибки I и II рода раздельно при несбалансированности классов:
  - Пусть $P(y=+1)=0.0000001; P(y=-1)=0.9999999$
  - Тогда при ошибке II рода 0 и ошибке I рода 0.5

    $P(f(x)=-1 \mid y=+1)=0.5$
  - Общая ошибка всего лишь

    $P(a(x)\neq y)=0.00000005$

Точность и полнота

- Точность (Precision)
  - Доля истинных объектов основного класса среди всех классифицированных, как основной класс
- Полнота (Recall)
  - Доля правильно распознанных объектов основного класса среди всех объектов основного класса из тестовой выборки

- Интегральный показатель F (F-measure)
  - $\mathrm{F} = 2 \times \frac{Precision \times Recall}{Precision + Recall}$

Точность и полнота

- Полнота – 10 из 13, т.е. 77%
- Точность – 10 из 12, т.е. 83%

Регулировка баланса

Почти все алгоритмы классификации допускают регулировку соотношения ошибки I и II рода за счет варьирования некоторого параметра

Параметр b в линейном классификаторе y = wx+b

ROC-кривая

M. Mathias. et. al. Face detection without bells and whistles. ECCV2014

ROC-кривая - зависимость между видами ошибок в зависимости от параметров

ROC кривая - Построение

- Для различных значений параметра строится таблица ошибок
  - Сам параметр в таблице не участвует!
  - Классификатор строится и оценивается на разных выборках!

Precision	Recall
100%	0%
90%	15%
83%	20%
…	…
0%	100%

- По таблице строиться набор точек в плоскости параметр1 x параметр 2
  - Например, precision/recall
  - Каждая строка таблицы – точка
- По точкам строиться кривая

Анализ ROC кривой

- По ROC-кривой мы можем выбрать параметр с оптимальным для наших целей соотношением ошибок

- Можем посчитать интегральные параметры:
  - Площадь под графиком – AUC
    - Дает некоторый объективный показатель качества классификатора
    - Позволяет сравнивать разные кривые
  - Другой параметр aP – average precision
    - Средняя точность для всех значений полноты

Пример

M. Mathias. et. al. Face detection without bells and whistles. ECCV2014

По ROC-кривой мы можем выбрать параметр с оптимальным для наших целей соотношением ошибок

Категоризация изображений

Это город?

Да

Нет

Это автобус?

Нет

Да

Задача категоризации изображений: определить, есть ли на изображении объект (сцена) заданной категории

Категоризация изображений

Будем собирать кучу признаков по всему изображению, делать из них вектор-признак, и его классифицировать

Обучение классификатора

Признаки изображения

- Мы будем рассматривать в основном признаки изображения
- Методы классификации можно использовать любые
    - Но сейчас у нас в руках только SVM

Признаки пикселей

Пространство цветов RGB

Пространство цветов HSV

Пространство цветов L*a*b*

Градиенты в каждом пикселе

Наличие и ориентация края в каждом пикселе

Использование напрямую

- Можно ли использовать признаки всех пикселей напрямую?
- Можно, если все изображения для классификации будут одинакового размера
- Нормализуем изображения, т.е. приведем их к одному размеру
- Вытянем изображение в вектор, признаки пикселей будут элементами вектора

Для распознавания, изображение должно описываться вектор-признаком фиксированной длины!

Гистограммы

- Мы можем измерять разные признаки (features) изображения
  - Цвет, края, градиенты, текстуру и т.д.
- Гистограммы – стандартный способ непараметрического описания распределения признаков

Images from Dave Kauchak

Квантование признаков

Image gradients

- Как построить гистограмму для признаков, которые изменяются непрерывно?
- Или разрешение слишком большое?
- «Квантование» - дискретизация признаков
  - Пр: все направления градиента разбиваем на 8 вариантов
  - Каждый вариант нумеруем от 1 до 8

Квантование признаков

- Есть фотографии тигров и медведей
- Хотим научиться отличать одни от других по цвету
- Оптимально ли использовать гистограмму цветов на интервале от [0,255]?
- Не все цвета встречаются!

Адаптивное квантование

- Хотим разбить элементы (пр. пиксели) на часто встречающиеся группы
- Затем можем каждый элемент сопоставить своей группе (дать ему номер группы)
- Кластеризация!
  - Например, методом k-средних

Для нашего примера

- Зададим количество кластеров (например, 100)
- Кластеризуем пиксели по цветам
  - Будут кластеры в белой, зеленой, черной, оранжевой областях
- Для каждого изображения построим гистограмму распределения цветов пикселов по кластерам

Пространственное распределение

У всех этих трех изображений похожие гистограммы цветов

Пространственная пирамида

Вычислим гистограму в каждом блоке и объединим все гистограммы в один вектор-признак

Уровень 0

Уровень 1

Уровень 2

S. Lazebnik et. Al. **Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories** *CVPR 2006*

Общая схема построения вектор-признака

Вычислим признаки по изображению → Квантуем признаки → Определим область для вычисления → Агрегируем признаки

RGB

Квантование на 10 уровней

Посчитанная гистограмма

Features → Coding → Spatial pooling

Мешок слов

Классификация текстов

- Представление документа без порядка: частоты слов из словаря («мешок слов») Salton & McGill (1983)

Классификация текстов

- Представление документа без порядка: частоты слов из словаря («мешок слов») Salton & McGill (1983)

Классификация текстов

- Представление документа без порядка: частоты слов из словаря («мешок слов») Salton & McGill (1983)

«Визуальные слова»

- Что в нашем случае «слово» и «словарь»?
- «Визуальное слово» – часто повторяющийся фрагмент изображения («visual word»)
- В изображении визуальное слово может встречаться только один раз, может ни разу, может много раз

Sivic, J. and Zisserman, A. Video Google: A Text Retrieval Approach to Object Matching in Videos. ICCV 2003

Словарь и мешок слов

- «Словарь» – набор фрагментов, часто повторяющихся в коллекции изображений

- Как составить словарь?
  - Составить большой список всех фрагментов по всей коллекции
  - Разделить весь список на похожие группы
  - Будем считать все фрагменты в одной группе – «экземплярами» одного и того же слова

- «Мешок слов» - гистограмма частот появления визуальных слов в изображении

Схема метода «мешок слов»

1. Извлечение фрагментов из коллекции изображений
2. Построение “визуального словаря”
3. Построение «мешка слов» для каждого изображения

1. Извлечение фрагментов

- Например, по регулярной сетке
- Можем без перекрытия, можем с перекрытием
- Есть и другие способы выбора фрагментов!

Вычисление признаков фрагментов

- Неупорядоченный список фрагментов из множества изображений
- Нужно описать каждый фрагмент вектор-признаком
  - Обычно на основе гистограмм ориентаций градиентов (подробнее в следующей лекции)

2. Обучение словаря

2. Обучение словаря

Slide credit: Josef Sivic

2. Обучение словаря

Slide credit: Josef Sivic

Пример словаря

Fei-Fei et al. 2005

Визуальные словари

- Как выбрать размер словаря?
  - Маленький: слова не могут описать все особенности
  - Большой: переобучение

- Вычислительная сложность
  - Деревья словарей (Nister & Stewenius, 2006)
  - Приближенные методы
  - Хеширование

3. Построение «мешка слов»

- Соберём тем же способом фрагменты для выбранного изображения
- «Квантуем» каждый фрагмент
  - Определим, какому слову из словаря он соответствует (фактически, сопоставим номер от 1 до N, где N – размер словаря)
- Построим гистограмму частот фрагментов для изображения

Резюме мешка слов

- Идея «визуальных слов» и «мешка слов» оказалась очень интересной и эффективной
- Мешок слов используется как сам по себе, так и совместно с другими признакми изображения
- Есть хорошие реализации:
  - Библиотека VLFeat http://www.vlfeat.org/

Резюме признаков

- Гистограммы – широко используемый метод описания распределения признаков в изображении (но не единственный)
- Гистограмму можем строить по регулярной сетке, а можем адаптивно, с помощью кластеризации
- Можем вычислить разные признаки:
  - Гистограммы цветов
  - Гистограммы градиентов
  - Гистограммы визуальных слов
- Каждый вид признаков можем считать по всему изображению, а можем по области
  - Часто используется пирамида разбиений из 3х уровней (1 + 4 + 16 гистограмм)

Категоризация изображений

- Описывать изображение научились, теперь нужно его категоризировать (классифицировать)
  - Nearest Neighbour (+kNN)
  - SVM
  - Random Forest
  - Boosting
  - Neural Networks

- Мы знаем SVM
- Можем придумать специальные ядра для сравнения гистограмм

Функции сравнения гистограмм

- Histogram intersection
(нормализованные гистограммы)

$$D(h_i, h_j) = 1 - \sum_{m=1}^{K} \min\left(h_i(m), h_j(m)\right)$$

- L1 distance

$$D(h_1, h_2) = \sum_{i=1}^{N} | h_1(i) - h_2(i) |$$

- χ^2 distance

$$D(h_1, h_2) = \sum_{i=1}^{N} \frac{\left(h_1(i) - h_2(i)\right)^2}{h_1(i) + h_2(i)}$$

- Quadratic distance (*cross-bin*)

$$D(h_1, h_2) = \sum_{i,j} A_{ij} (h_1(i) - h_2(j))^2$$

Jan Puzicha, Yossi Rubner, Carlo Tomasi, Joachim M. Buhmann: Empirical Evaluation of Dissimilarity Measures for Color and Texture. ICCV 1999

Проблемы сравнения гистограмм

- Все метрики сравнения гистограмм чувствительны к размеру ячеек
- Можно использовать более широкие ячейки, но при этом теряем «разрешающую способность»

Ядра для нелинейного SVM

- Поскольку вектор-признак – гистограмма, нам потребуются специальные ядра
- Ядро пересечения гистограмм:

$$I(h_1, h_2) = \sum_{i=1}^{N} \min(h_1(i), h_2(i))$$

- Обобщенное Гауссово ядро:

$$K(h_1, h_2) = \exp\left(-\frac{1}{A} D(h_1, h_2)^2 \right)$$

- *D* может быть евклидовым расстоянием, χ^2 , и т.д.

J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid, Local Features and Kernels for Classifcation of Texture and Object Categories: A Comprehensive Study, IJCV 2007

Резюме лекции

- Мы познакомились с задачей классификации и методом SVM
- Категоризация изображений – одна из основных задач распознавания изображений
- В качестве признаков чаще всего используют гистограммы распределений цвета, градиентов, краёв, визуальных слов
- Любой метод классификации на основе машинного обучения можно использовать
- Текущие результаты не идеальны, задача в процессе исследования