GRAPHICS & MEDIA LAB
VIDEO GROUP

Обзор алгоритмов машинного обучения

Воронов Александр

Video Group
CS MSU Graphics & Media Lab

Содержание

- Введение
- Дерево решений
- Статистические алгоритмы
- Метрические алгоритмы
- SVM
- AdaBoost

Постановка задачи
Терминология

- Множество объектов: X
- Конечное множество классов: Y
- Любой объект $x \in X$ соответствует хотя бы одному классу $y_i \in Y$

Постановка задачи

По конечной выборке прецедентов $X^l : (x_i, y_i)_{i=1}^l$ построить отображение $a: X \to Y$, удовлетворяющее следующим условиям:

- Эффективная программная реализация
- Воспроизведение заданных ответов на обучающей выборке
- Обобщающая способность для всего множества X
- Априорные ограничения (соответствие модели)

Постановка задачи
Оценка обобщающей способности

Функционал качества:

$$Q(a, X^l) = \frac{1}{l}\sum_{i=1}^{l} \alpha(a, x_i)$$

a - тестируемый алгоритм

$\alpha(a, x_i)$ - функция, определяющая величину ошибки алгоритма

$$\mu(X^l) = \arg\min_{a \in A} Q(a, X^l)$$

Постановка задачи
Оценка обобщающей способности

- Дана выборка $X^L = (x_i, y_i)_{i=1}^L$
- Разобьём её N способами на обучающую X_n^l и контрольную X_n^k подвыборки ($k = L - l$)
- Оценка скользящего контроля (cross-validation):

$$\mathrm{CV}(\mu, X^L) = \frac{1}{N} \sum_{n=1}^{N} Q(\mu(X_n^l), X_n^k)$$

- CV совпадает с матожиданием потерь

Примеры прикладных задач

GRAPHICS & MEDIA LAB
VIDEO GROUP

- Медицинская диагностика
- Распознавание спама
- Рубрикация текста
- Распознавание рукописных символов
- Оценивание заёмщиков
- Прогнозирование потребительского спроса
- и т.д.

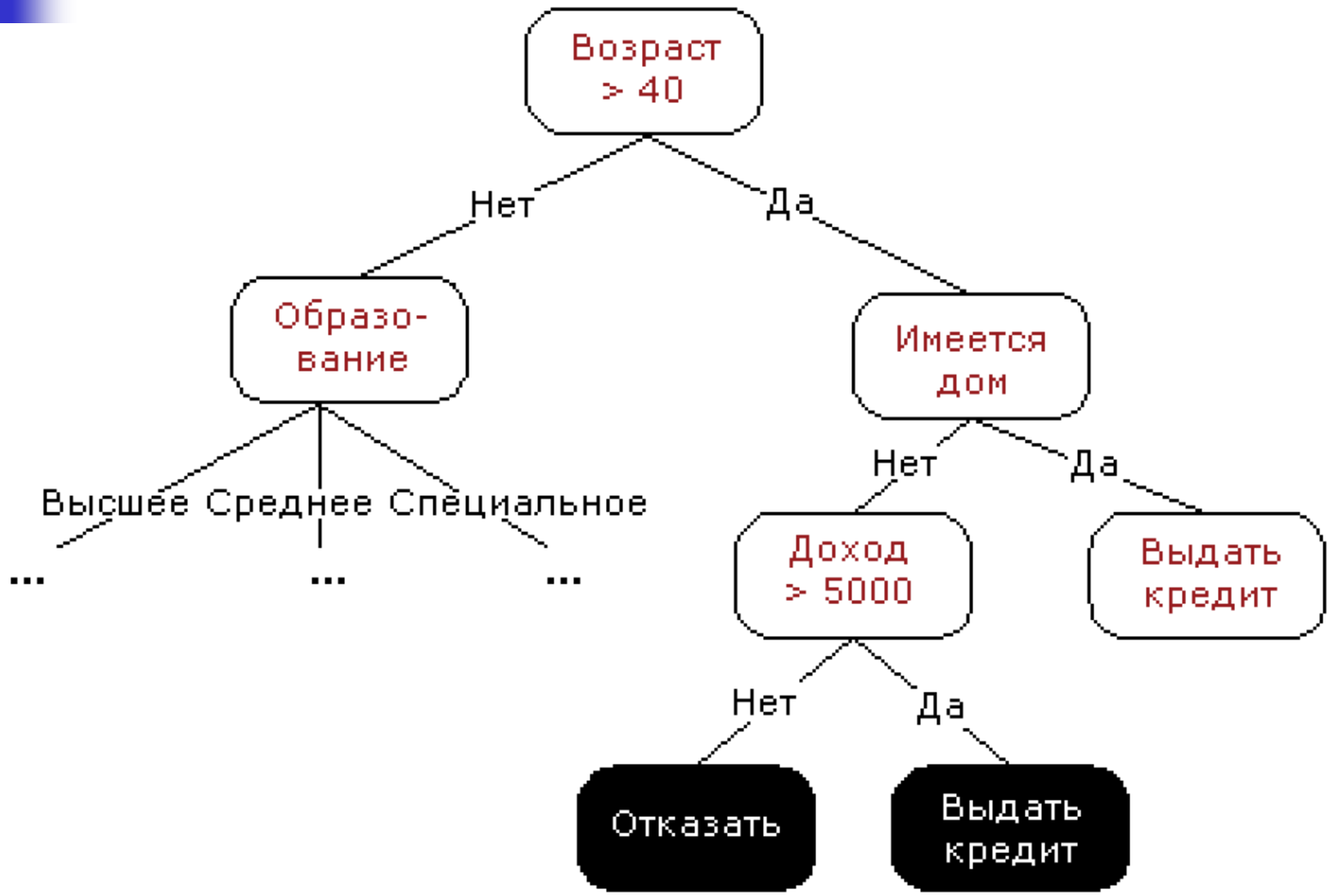
Эвристические принципы обучения по прецедентам

- сходства
- минимизации эмпирического риска
- регуляризации (штраф на сложность алгоритма)
- разделимости (можно описать некоторую поверхность, разделяющую классы)
- отделимости и закономерности (можно описать область, которая включает объекты только одного класса)
- самоорганизации моделей (структура модели алгоритма заранее не известна)
- композиции

Содержание

- Введение
- **Дерево решений**
- Статистические алгоритмы
- Метрические алгоритмы
- SVM
- AdaBoost

Дерево решений
Пример



Дерево решений
Автоматическое построение

$X = \{x_1, \ldots, x_n\}$, p_i – вероятность события x_i

$H(X) = -\sum_{i=1}^{n} p_i \log_2 p_i$ - энтропия множества X

$C = \{c_1, \ldots, c_m\}$ – множество классов

F – признак с возможными значениями $\{f_1, \ldots, f_d\}$

Количество информации класса C относительно признака F:

$$I(C,F) = \sum_{i=1}^{m}\sum_{j=1}^{d} P(C = c_i, F = f_i) \log_2 \frac{P(C = c_i, F = f_i)}{P(C = c_i)P(F = f_i)}$$

Дерево решений
Автоматическое построение

1. Признак с наибольшим количеством информации выбирается в качестве корневого узла
2. Если подмножество событий ветви не совпадает с одним из классов, то алгоритм запускается рекурсивно для этой ветви

Содержание

- Введение
- Дерево решений
- **Статистические алгоритмы**
- Метрические алгоритмы
- SVM
- AdaBoost

Статистические алгоритмы
Обозначения

- $P_y = P(y)$ – априорная вероятность класса y
- $p_y(x) = \mathrm{p}(x|y)$ – функция правдоподобия класса y
- $p(x,y)$ – плотность распределения
- λ_{ys} – величина потери при отнесении объекта класса y к классу s
- $A_y = \{x \in X \mid a(x) = y\}, y \in Y$

Статистические алгоритмы
Обозначения

- Функционал среднего риска:

$$R(a) = \sum_{y \in Y} \sum_{s \in Y} \lambda_{ys} P_y(A_s \mid y)$$

- Формула Байеса

$$P(y \mid x) = \frac{p(x,y)}{p(x)} = \frac{p_y(x) P_y}{\sum_{s \in Y} p_s(x) P_s}$$

Статистические алгоритмы
Схема работы

1. Задаются штрафы ошибочной классификации λ_{ys}.
2. По обучающей выборке вычисляются функции, характеризующие классы.
3. На основе этих функций строится алгоритм, который минимизирует функционал среднего риска.

Статистические алгоритмы
Обозначения

- Оптимальный алгоритм классификации

$$a(x) = \arg\min_{s \in Y} \sum_{y \in Y} \lambda_{ys} P_y p_y(x)$$

- При условии, что $\lambda_{ys} \equiv \lambda_y$

$$a(x) = \arg\max_{y \in Y} \lambda_y P_y p_y(x) = \arg\max_{y \in Y} \lambda_y \mathrm{P}(y \mid x)$$

- Разделяющая поверхность:

$$\lambda_t P_t p_t(x) = \lambda_s P_s p_s(x)$$

Статистические алгоритмы
Восстановление плотности

- Оценка априорной вероятности класса y:

$$\hat{P}_y = \frac{l_y}{l}, l_y = \left| X_y^l \right|, y \in Y$$

- Чтобы восстановить функции правдоподобия $p_y(x)$, рассмотрим общую задачу:

  Для выборки $X^m = \{x_1, \ldots, x_m\}$ построить эмпирическую оценку плотности, приближающую $p(x)$ на всём X.

Статистические алгоритмы

Предположим, что $p(x) = \varphi(x,\theta)$

- φ – фиксированная функция
- θ – параметр, значение которого выбирается из принципа максимума правдоподобия:

$$L(X^m, G^m, \theta) = \sum_{i=1}^{m} g_i \ln \varphi(x_i, \theta) \to \max_{\theta}$$

- $G^m = (g_1, \ldots, g_m)$

Статистические алгоритмы

- Предположим, что

$$\varphi(x,\theta) = N(x,\mu,\Sigma) = (2\pi)^{-\frac{n}{2}} \left|\Sigma\right|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(x-\mu)^{\tau}\Sigma^{-1}(x-\mu)\right)$$

то есть n-мерное нормальное распределение с матожиданием $\mu \in \mathrm{R}^{n}$, $\Sigma \in \mathrm{R}^{nxn}$

Статистические алгоритмы

- Вычисление:

$$\sum_{i=1}^{m} g_i = 1$$

$$\hat{\mu} = \sum_{i=1}^{m} g_i x_i; \qquad \hat{\Sigma} = \sum_{i=1}^{m} g_i (x_i - \hat{\mu})(x_i - \hat{\mu})^{\tau}$$

- Можно положить $g_i = \frac{1}{m}$
- Несмещённая оценка ков.матрицы:

$$\hat{\Sigma} = \frac{1}{m-1} \sum_{x=1}^{m} (x_i - \hat{\mu})(x_i - \hat{\mu})^{\tau}$$

Статистические алгоритмы
Квадратичный дискриминант

Если классы имеют нормальные функции правдоподобия, то решающее правило задает квадратичную разделяющую поверхность. Поверхность вырождается в линейную, если ков.матрицы классов равны.

Статистические алгоритмы
Линейный дискриминант Фишера

Фишер предложил считать ковариационные матрицы равными, даже если они на самом деле не равны.

$$a(x) = \arg\max_{y \in Y}(\lambda_y P_y p_y(x)) =$$

$$= \arg\max_{y \in Y}(\ln(\lambda_y P_y) - \frac{1}{2}\hat{\mu}_y^{\tau}\hat{\Sigma}^{-1}\hat{\mu}_y + x^{\tau}\hat{\Sigma}^{-1}\hat{\mu}_y) =$$

$$= \arg\max_{y \in Y}(x^{\tau}\alpha_y + \beta_y)$$

Статистические алгоритмы
Линейный дискриминант Фишера

Обучение сводится к оцениванию матожидания и общей ковариационной матрицы для всей выборки.

Статистические алгоритмы
Наивный байесовский классификатор

- Если предположить , что признаки объекта независимы и нормально распределены, то общая плотность вычисляется как произведение плотностей характеристик
- Плотность каждой характеристики внутри класса вычисляется значительно проще
- В реальности такая ситуация встречается редко, на большинстве задач качество классификации будет относительно низким

Статистические алгоритмы
Наивный байесовский классификатор

$\xi_1 = f_1(x),\ldots,\xi_n = f_n(x)$ - признаки

$$p_y(x) = p_{y1}(\xi_1)\ldots p_{yn}(\xi_n)$$

Итоговый алгоритм:

$$a(x) = \arg\max_{y \in Y}\left(\ln\frac{\lambda_y l_y}{l} + \sum_{j=1}^{n}\ln\hat{p}_{yj}(\xi_j)\right)$$

Статистические алгоритмы
Выводы

Преимущества:

- Байесовское решающее правило оптимально, имеет простую формулу, легко реализуется программно
- Имеет широкую область применения, часто используется в качестве эталона при тестировании других алгоритмов

Недостатки:

- При неправильном подходе к восстановлению функций правдоподобия качество работы алгоритма может быть очень низким

Содержание

- Введение
- Дерево решений
- Статистические алгоритмы
- **Метрические алгоритмы**
- SVM
- AdaBoost

Метрические алгоритмы

- Метрические алгоритмы:

  основанные на анализе сходства объектов.
- Гипотеза компактности:

  Классы образуют компактно локализованные множества в пространстве объектов.
- Вводится метрика $\rho(x, x')$ в пространстве объектов X

Метрические алгоритмы

u – рассматриваемый объект

X^l – обучающая выборка

$w(i, u)$ – оценка степени важности i-го соседа

$\Gamma_y(u, X^l) = \sum_{i=1}^{l} [y_u^{(i)} = y] w(i,u)$ - суммарный вес ближайших обучающих объектов

Метрический алгоритм:

$$a(u, X^l) = \arg\max_{y \in Y} \Gamma_y(u, X^l)$$

Метрические алгоритмы
Схема работы

GRAPHICS & MEDIA LAB
VIDEO GROUP

Обучение:

1. Выбор метрики сходства между объектами
2. Удаление из обучающей выборки неинформативных и шумовых объектов

Классификация:

Объект относится к тому классу, для которого максимален вес ближайших объектов из обучающей выборки.

Метрические алгоритмы
Весовые функции

- Метод ближайшего соседа (1NN):

  $w(i, u) = [i = 1]$
- Метод k ближайших соседей (kNN):

  $w(i, u) = [i \le k]$
- Метод взвешенных ближайших соседей:

  $w(i, u) = [i \le k]q^i$

Метрические алгоритмы
Метод парзеновского окна

$K(z)$ – функция ядра, невозрастающая на $[0, \infty)$

$$w(i,u) = K\left(\frac{\rho(u, x_u^{(i)})}{h}\right)$$

При неравномерном распределении объектов можно использовать окно переменной ширины:

$$h(u) = \rho(u, x_u^{(k+1)})$$

Доп.ограничение на K: $z > 1, K(z)=0$

Метрические алгоритмы
Отбор эталонных объектов

- Эталоны – типичные представители классов
- При исключении из выборки шумовые и неинформативные объекты повышается качество классификации и уменьшается объём хранимых данных

Метрические алгоритмы
Отбор эталонных объектов

Отступ объекта x_i относительно алгоритма $a(u)$

$$M(x_i) = \Gamma_{y_i}(x_i) - \max_{y \in Y \setminus y_i} \Gamma_y(x_i)$$

Объекты:

- Эталонные (большой положительный отступ)
- Неинформативные (положительный отступ)
- Пограничные (отступ, близкий к нулю)
- Ошибочные объекты (отрицательный отступ)
- Шумовые объекты или выбросы (большой отрицательный отступ)

Из выборки удаляются неинформативные и шумовые объекты

Метрические алгоритмы
Выводы

Преимущества:

- Нет необходимости выделять признаки (прецедентная логика)
- Простота реализации

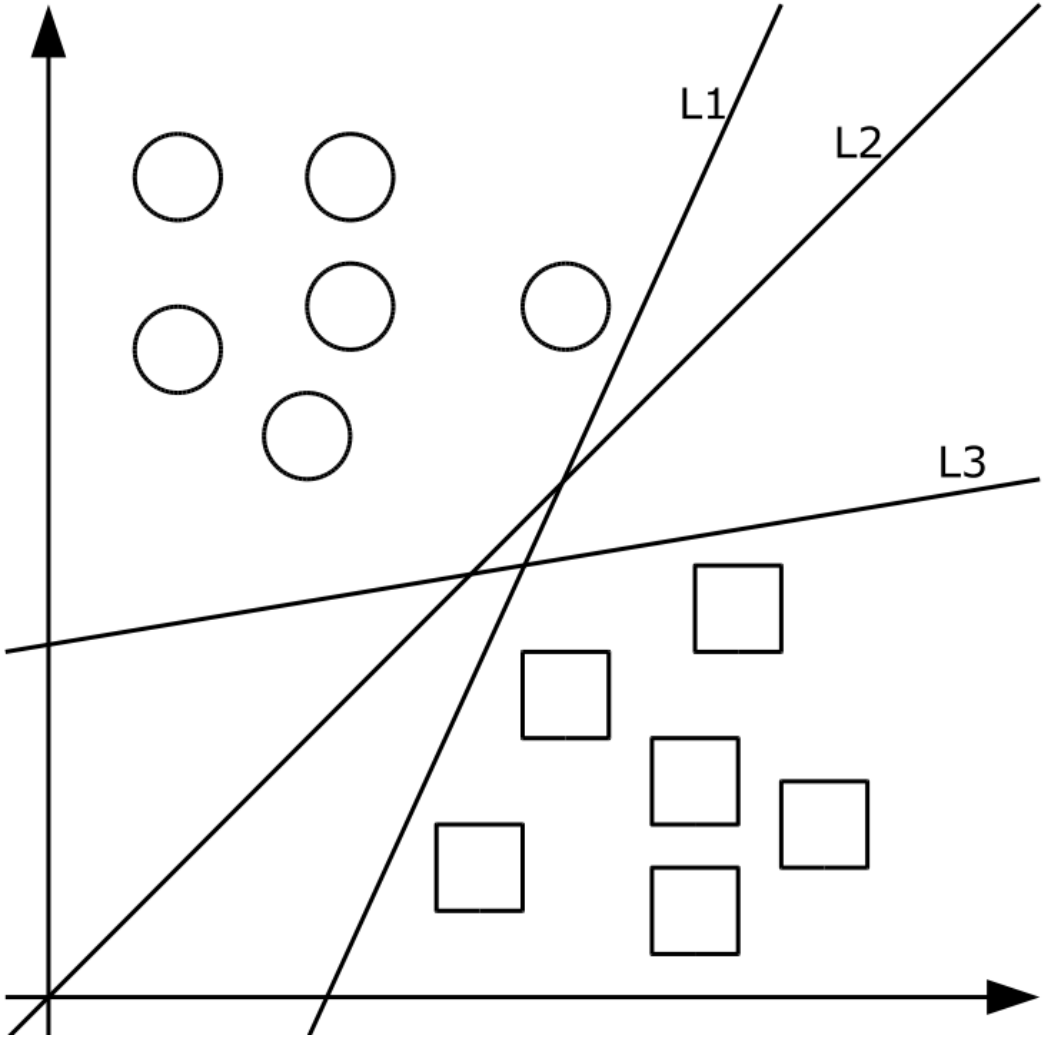
Недостатки:

- Необходимость хранить обучающую выборку
- Поиск ближайших соседей предполагает большой число сравнений

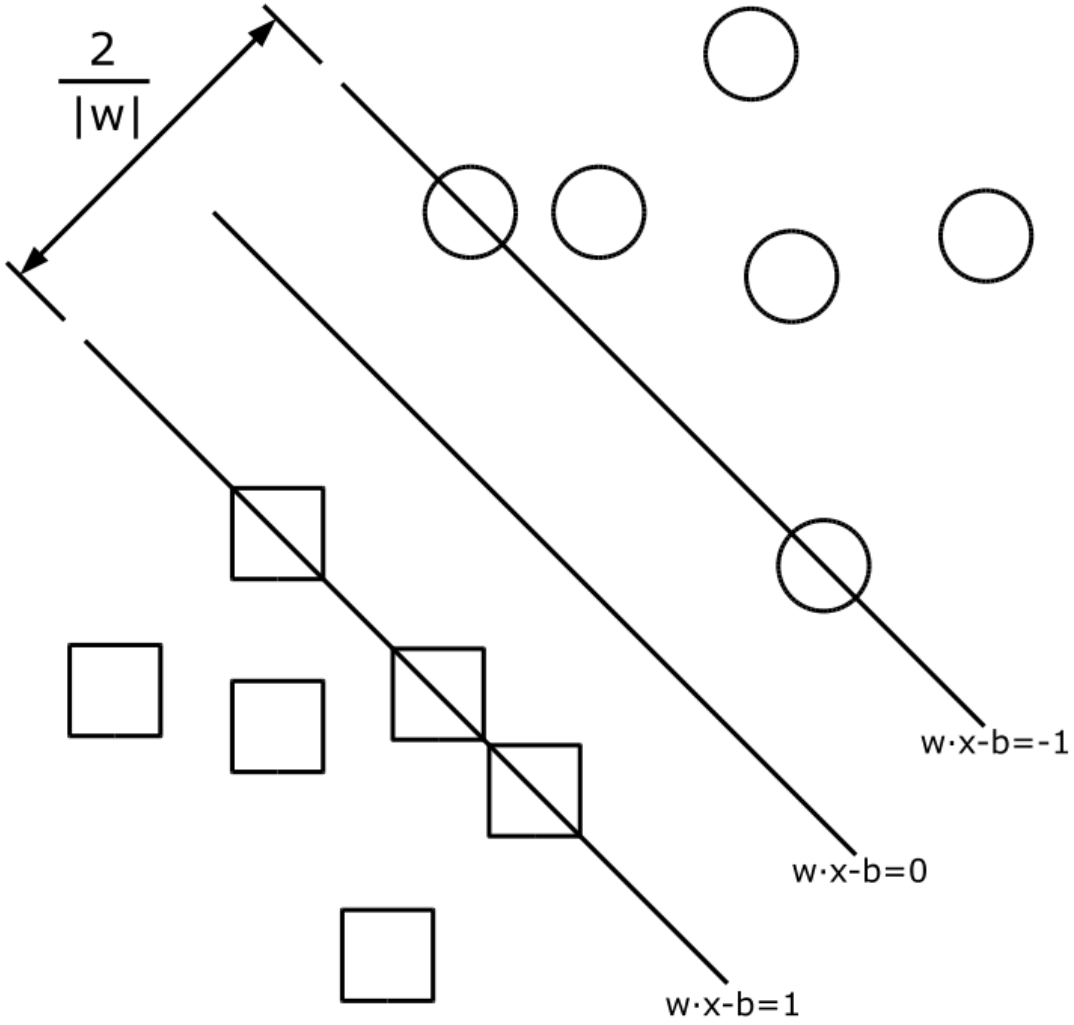
Содержание

- Введение
- Дерево решений
- Статистические алгоритмы
- Метрические алгоритмы
- **SVM**
- AdaBoost

SVM



SVM



SVM

$$X = \mathrm{R}^n, \quad Y = \{-1, +1\}$$

$$a(x) = \operatorname{sgn}\left(\sum_{j=1}^{n} w_j x^j - w_0 \right) = \operatorname{sgn}\left(\langle w, x \rangle - w_0 \right)$$

- w, w_0 – параметры алгоритма
- $\langle w, x \rangle = w_0$ - разделяющая гиперплоскость

SVM
Схема работы

Обучение:

1. Для поиска максимальной ширины разделяющей полосы при минимальной ошибке составляется функция Лагранжа
2. Ищется седловая точка функции Лагранжа.
3. Находятся опорные точки, на их основе вычисляются параметры алгоритма

SVM
Ширина разделяющей полосы

x_+ и x_- - произвольные точки классов, лежащие на границе полосы

Тогда ширина полосы:

$$\left\langle (x_+ - x_-), \frac{w}{\|w\|} \right\rangle = \frac{\langle w, x_+ \rangle - \langle w, x_- \rangle}{\|w\|} = \frac{(w_0 + 1) - (w_0 - 1)}{\|w\|} = \frac{2}{\|w\|}$$

Для линейно разделимой выборки требуется найти параметры w, w_0, такие, что при выполнении условия $y_i(\langle w, x_i \rangle - w_0) \geq 1$ норма w будет минимальна.

SVM
Задача поиска седловой точки

$$\begin{cases} L(w, w_0, \lambda) = \frac{1}{2}\langle w, w\rangle - \sum_{i=1}^{l} \lambda_i \left(y_i \left(\langle w, x_i \rangle - w_0 \right) - 1 \right) \to \min_{w, w_0} \max_{\lambda} \\ \lambda_i \geq 0, i = 1, \ldots, l \\ \lambda_i = 0, \textit{либо} \langle w, x_i \rangle - w_0 = y_i, i = 1, \ldots, l \end{cases}$$

Необходимые условия седловой точки:

$$\frac{\partial L}{\partial w} = w - \sum_{i=1}^{l} \lambda_i y_i x_i = 0 \quad \Rightarrow \quad w = \sum_{i=1}^{l} \lambda_i y_i x_i$$

$$\frac{\partial L}{\partial w_0} = -\sum_{i=1}^{l} \lambda_i y_i = 0 \quad \Rightarrow \quad \sum_{i=1}^{l} \lambda_i y_i = 0$$

SVM
Задача поиска седловой точки

Из необходимых условий седловой точки следует :

$$\begin{cases} -L(\lambda) = -\sum_{i=1}^{l} \lambda_i + \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \lambda_i \lambda_j y_i y_j \left(\langle x_i, x_j \rangle \right) \to \min\limits_{\lambda} \\ \lambda_i \geq 0, i = 1, \ldots, l \\ \sum_{i=1}^{l} \lambda_i y_i = 0 \end{cases}$$

SVM

- После решения задачи вычисляем:

$$w = \sum_{i=1}^{l} \lambda_i y_i x_i$$

$$w_0 = med\left\{\langle w, x_i \rangle - y_i : \lambda_i > 0, i = 1,\ldots,l\right\}$$

- Итоговый алгоритм:

$$a(x) = \operatorname{sgn}\left(\langle w, x \rangle - w_0\right)$$

SVM
Линейно неразделимая выборка

Добавим в исходную задачу минимизации нормы w штраф за суммарную ошибку:

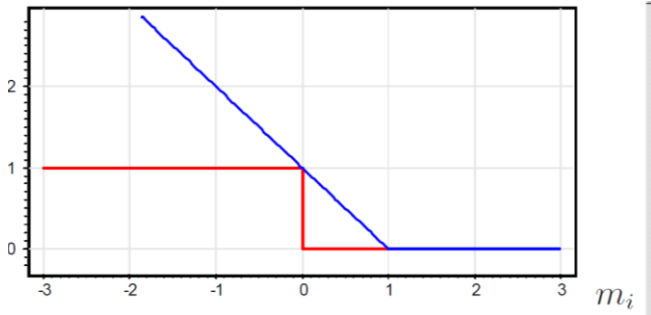
$$\begin{cases} \frac{1}{2}\langle w, w\rangle + C\sum_{i=1}^{l}\xi_i \to \min\limits_{w, w_0, \xi} \\ y_i\left(\langle w, x_i\rangle - w_0\right) \geq 1 - \xi_i, \qquad i = 1,\ldots,l \\ \xi_i \geq 0, \quad i = 1,\ldots,l \end{cases}$$

SVM
Линейно неразделимая выборка

- Введём понятие отступа: $m_i = y_i\left(\langle w, x_i\rangle - w_0\right)$
- Рассмотрим функционал числа ошибок:

$$Q(a, X^l) = \sum_{i=1}^{l} [m_i < 0]$$

- Заменим пороговую функцию на её верхнюю оценку:
$[m_i < 0] \le (1 - m_i)_+$



- Добавим к Q штрафное слагаемое $\tau\|w\|^2$, учитывающее норму w

SVM
Линейно неразделимая выборка

Задача минимизации полученного функционала

$$Q(a, X^l) = \sum_{i=1}^{l} (1 - m_i)_+ + \tau \|w\|^2 \to \min_{w, w_0}$$

эквивалентна исходной задаче

$$\begin{cases} \frac{1}{2}\langle w, w \rangle + C \sum_{i=1}^{l} \xi_i \to \min_{w, w_0, \xi} \\ y_i \left(\langle w, x_i \rangle - w_0 \right) \geq 1 - \xi_i, \qquad i = 1, \ldots, l \\ \xi_i \geq 0, \quad i = 1, \ldots, l \end{cases}$$

при $\tau = \frac{1}{2C}$

SVM
Линейно неразделимая выборка

- Соответствующая функция Лагранжа:

$$L(w, w_0, \xi, \lambda, \eta) = \frac{1}{2}\langle w, w\rangle - \sum_{i=1}^{l} \lambda_i \left(y_i \left(\langle w, x_i \rangle - w_0 \right) - 1 \right) -$$

$$- \sum_{i=1}^{l} \xi_i (\lambda_i + \eta_i - C) \to \min_{w, w_0} \max_{\lambda}$$

SVM
Линейно неразделимая выборка

GRAPHICS & MEDIA LAB
VIDEO GROUP

Задача поиска седловой точки:

$$\begin{cases} L(w, w_0, \xi, \lambda, \eta) \to \min\limits_{w, w_0, \xi} \max\limits_{\lambda, \eta} \\ \xi_i \geq 0,\ \lambda_i \geq 0,\ \eta_i \geq 0, \qquad i = 1, \ldots, l \\ \lambda_i = 0, \quad \textit{либо} \quad y_i\left(\langle w, x_i \rangle - w_0\right) = 1 - \xi_i, \qquad i = 1, \ldots, l \\ \eta_i = 0, \quad \textit{либо} \quad \xi_i = 0, \qquad i = 1, \ldots, l \end{cases}$$

SVM
Спрямляющие пространства

- Ещё один способ решения проблемы линейной неразделимости:
переход из пространства объектов X в пространство H с помощью преобразования $\psi: X \to H$
- Пространство H называется спрямляющим
- SVM строится так же, только на основе объектов $\psi(x_i)$ вместо x_i.
- $K(x, x') = \langle \psi(x), \psi(x') \rangle$ - ядровая функция

SVM
Выводы

Преимущества:

- Решение задачи хорошо оптимизируется: сводится к задаче квадратичного программирования
- Более уверенная классификация за счёт максимизации ширины разделяющей полосы

Недостатки:

- Неустойчивость к шуму, выбросы существенно учитываются
- Нет общих методов построения ядер или спрямляющих пространств

Содержание

- Введение
- Дерево решений
- Статистические алгоритмы
- Метрические алгоритмы
- SVM
- **AdaBoost**

AdaBoost
Постановка задачи

- Классификация на два класса: $Y = \{-1,+1\}$
- $b_t(x)$ – некоторые базовые алгоритмы
- Искомый алгоритм – взвешенная сумма базовых:

$$a(x) = \operatorname{sgn}\left(\sum_{t=1}^{T} \alpha_t b_t(x)\right), \quad x \in X$$

- Функционал качества композиции:

$$Q_T = \sum_{t=1}^{l} \left[y_i \sum_{t=1}^{T} \alpha_t b_t(x_i) < 0 \right]$$

AdaBoost

Упрощение задачи минимизации функционала Q_t:

- *Эвристика 1*: при добавлении в композицию нового слагаемого оптимизировать только его, не трогая предыдущих
- *Эвристика 2*: аппроксимировать пороговую функцию потерь в Q_t непрерывно дифференцируемой оценкой сверху.

AdaBoost

- Аппроксимация экспонентой:

$$Q_t \le \tilde{Q}_t = \sum_{i=1}^{l} \exp\left(-y_i \sum_{t=1}^{T} \alpha_t b_t(x_i) \right) =$$

$$= \sum_{i=1}^{l} \underbrace{\exp\left(-y_i \sum_{t=1}^{T-1} \alpha_t b_t(x_i) \right)}_{w_i} \exp(-y_i \alpha_T b_T(x_i))$$

- Введём нормированный вектор весов объектов:

$$\tilde{W}^l = (\tilde{w}_1, \ldots, \tilde{w}_l), \tilde{w}_i = \frac{w_i}{\sum_{j=1}^{l} w_j}$$

AdaBoost

Теорема 1:

$$\begin{cases} Q(b, U^l) = \sum_{i=1}^{l} u_i [y_i b(x_i) < 0], \qquad \sum_{i=1}^{l} u_i = 1 \\ \min_b Q(b, W^l) < 1/2, \qquad \forall W^l : \sum_{i=1}^{l} w_i = 1 \end{cases}$$

Тогда:

$$b_T = \arg\min_b Q(b, \tilde{W}^l)$$

$$\alpha_T = \frac{1}{2} \ln \frac{1 - Q(b_T, \tilde{W}^l)}{Q(b_T, \tilde{W}^l)}$$

AdaBoost

Теорема 2:

Если существует $\gamma \geq 0$ такое, что на каждом шаге $Q(b_t, \tilde{W}^l) < 1/2 - \gamma$, то *AdaBoost* гарантирует построение корректного алгоритма $a(x)$ за конечное число шагов.

AdaBoost
Алгоритм обучения

1. инициализация весов объектов: $w_i := 1/l, \quad i = 1, \ldots, l$
2. **для всех** $l = 1, \ldots, T$, пока не выполнен критерий останова
3. $b_t := \arg\min_b Q(b, W^l)$
4. $\alpha_t := \frac{1}{2} \ln \frac{1 - Q(b_t, W^l)}{Q(b_t, W^l)}$
5. пересчёт весов объектов: $w_i := w_i \exp(-\alpha_t y_i b_t(x_i)), \quad i = 1, \ldots, l$
6. нормировка весов объектов: $w_0 := \sum_{j=1}^{l} w_j$

$$w_i := w_i / w_0, \quad i = 1, \ldots, l$$

AdaBoost
Выводы

Достоинства:

- Хорошая обобщающая способность
- Простота реализации
- Возможность идентификации выбросов по высокому значению w_i

Недостатки:

- Переобучение при значительном уровне шума
- Требует длинных выборок
- Может привести к построению громоздких композиций

Литература

1. Курс лекций К.В. Воронцова по машинному обучений (2007-2008) http://www.machinelearning.ru/wiki/index.php?title=Машинное_обучение_(курс_лекций%2C_К.В.Воронцов)
2. Л.Шапиро, Дж.Стокман «Компьютерное зрение», глава 4, С.126-167