Методы решения задач классификации с категориальными признаками

Глушкова Таисия Воронкова Дарья

12 декабря, 2016

1. Обозначения

- Матрица объект-признак
  $F = ||f_{ij}||_{mxn}$,
  m - число объектов, n - число признаков,
  f_{ij} - значение j-го признака на i-м объекте
- Целевой вектор
  $(y_1, ..., y_m)^T$,
  y_i - значение целевого признака на i-м объекте
  Будем рассматривать задачу классификации с двумя непересекающимися классами:
  $\{y_1, ..., y_m\} = \{0, 1\}$
- Постановка задачи
  Нужно разработать алгоритм, который по признаковому описанию нового объекта $(f_1, ..., f_n)$ выдает значение его целевого признака y

1. Обозначения

- Отложенный контроль (hold-out)
- leave-one-out
- Будем считать, что все признаки в задаче категориальные
- Всегда можно перенумеровать категории
  $f_{ij} \in \{1, 2, ..., n_j\}$,
  n_j - число разных категорий j-го признака,
  $j \in \{1, 2, ..., n\}$

1. Обозначения

- one hot encoding
  $\{1, 2, ..., n_*\}$ - значения категориального признака
  Заменим столбец $(h_1, ..., h_m)^T$ матрицы F (соответсвующего признака) бинарной матрицей $||\delta_{ij}||_{m \times n_*}$,

$$\delta_{ij} = \begin{cases} 1, & h_i = j \\ 0, & h_i \neq j \end{cases} \tag{1}$$

  $i \in \{1, 2, ..., m\}$,
  $j \in \{1, 2, ..., n_*\}$.
- Нет потерь информации
- В случае большого числа категорий (n_*) такая перекодировка может существенно увеличить число столбцов в матрице объект-признак

1. Обозначения

- Будем считать, что алгоритм по описанию объекта выдает значения из отрезка $[0; 1]$.
- В качестве функционала качества будем использовать площадь под ROC-кривой (receiver operating characteristics): AUC (area under curve).
- $y^1, ..., y^q$ - верные метки контрольных объектов
  $a^1, ..., a^q$ - значения, которые выдал алгоритм на контрольных объектах
  ROC-кривая образуется соединением точек $(fp(c), tp(c))$,

$$fp(c) = \frac{|\{t \in \{1, 2, ..., q\} | y^t = 0, a^t \geq c\}|}{|\{t \in \{1, 2, ..., q\} | y^t = 0\}|}, \tag{2}$$

$$tp(c) = \frac{|\{t \in \{1, 2, ..., q\} | y^t = 1, a^t \geq c\}|}{|\{t \in \{1, 2, ..., q\} | y^t = 1\}|}, \tag{3}$$

  при варьировании порога c.

1. Обозначения

- $(f_{1j}, ..., f_{mj}), (f_{1t}, ..., f_{mt})$
- Сформируем новый признак $((f_{1j}, f_{1t}), ..., (f_{mj}, f_{mt}))$ - конъюнкция двух исходных признаков
- Конъюнкция порядка k: формирование признака на основе k категориальных

2. Сингулярное разложение матриц в анализе данных

- Разложение заключается в представлении матрицы Z размера $m \times n$ в виде произведения $U\Lambda V$, где
  $U_{m\times m}$ и $V_{n\times n}$ - ортогональные матрицы
  $\Lambda_{m\times n}$ - матрица с элементами $\lambda_1, ..., \lambda_r, 0, ..., 0$ на главной диагонали, остальные - нули
  $r = rank(Z)$
  $\lambda_1 \geq ... \geq \lambda_r \geq 0$
- $\lambda_1, ..., \lambda_r, 0, ..., 0$ - сингулярные числа, которые равны квадратным корням собственных значений матрицы ZZ^T

2. Сингулярное разложение матриц в анализе данных

- Разложение можно переписать в сокращённом представлении, считая, что матрица U имеет размеры $m \times r$, V – $m \times r$, Λ – $r \times r$, тогда

$$Z = \sum_{i=1}^{r} \lambda_i u_i v_i^T$$

u_i – i-й столбец матрицы U,
v_i^T – i-я строка матрицы V

2. Сингулярное разложение матриц в анализе данных

- Наилучшее приближение матрицы Z среди всех матриц ранга k в L_2-норме:

$$Z = \sum_{i=1}^{r} \lambda_i u_i v_i^T$$

$k < rank(Z)$
$||Z - Z_k||_2 = \lambda_{k+1}$ (теорема Эккарта– Янга)
Поэтому мы будем говорить об усечённом сингулярном разложении:

$$Z \approx U_{m \times k} \Lambda_{k \times k} V_{k \times n}$$

2. Сингулярное разложение матриц в анализе данных

- Одно из наиболее частых применений SVD - сокращение размерности пространства.
  Пусть $F = ||f_{ij}||_{mxn}$ - исходная матрица объект-признак и число признаков n очень велико.
  Тогда сделаем усечённое сингулярное разложение матрицы

$$F \approx U\Lambda V$$

  , где U - новая матрица объект признак
  Если известны контрольные объекты, то раскладываем матрицу признаковых описаний для всех объектов.
  Если контрольные объекты не известны, то вместо U используем матрицу $FV^T\Lambda^{-1}$, а при классификации объекта с признаковым описанием $f = (f_1, ..., f_n)$ его заменяем на $fV^T\Lambda^{-1}$

3. Реальная прикладная задача с категориальными признаками

- Amazon.com - Employee Access Challenge
  https://www.kaggle.com/c/amazon-employee-access-challenge/data
- Матрица объект-признак 32769x10
- 8 признаков
- Число категорий для различных признаков

Номер признака	1	2	3	4	5	6	7	8
Число категорий	7518	4243	128	177	449	343	2358	67

- Обучение - первые 25000 объектов

4. Линейные методы

- - методы, которые при классификации объекта $(f_1, ..., f_n)$ используют значение линейной комбинации

$$L = w_1 f_1 + ... + w_n f_n$$

$(f_1, ..., f_n)$ - вещественные признаки (либо исходные, либо новые, полученные некоторым преобразованием из исходных)

- Пример линейного метода - линейный классификатор, результат действия которого - индикатор сравнения с порогом

$$S(x) = \begin{cases} 1, & L \geq w_0, \\ 0, & L < w_0 \end{cases}$$

- Можно использовать значение L в качестве ответа и перевести на отрезок $[0, 1]$ преобразованием

$$\frac{1}{1 + e^{-L}}$$

4. Линейные методы

- Простейший персептронный алгоритм
  Последовательно перебираются все объекты обучения и вектор весов $(w_1, ..., w_n)$ корректируется в случае, если i-й объект классифицируется неверно: к нему прибавляется

$$\lambda(2y_i - 1)(f_{i1}, ..., f_{in})$$

- Если в процессе перебора объектов не было коррекций весов, то классификатор настроен, иначе объекты перебираются снова (в случайном порядке)
- В задаче Employee Access Challenge параметр λ оптимальнее было выбирать так:

$$\lambda = \frac{1}{\log(s+1)}$$

  s - номер итерации (номер прохождения по обучающей выборке)

4. Линейные методы

- Простейший персептронный алгоритм:
  - настраивается достаточно быстро
  - показывает неплохое качество (0.8285 AUC)
  - может быть использован в качестве «бенчмарка» (для сравнения с ним более сложных моделей алгоритмов)

- Логистическая регрессия
  Последовательно пересчитываются веса по формуле:

$$(w_1, ..., w_n) = (w_1, ..., w_n) + \lambda \sum_{i=1}^{m} \left(y_i + \frac{1}{1 + e^{-(w_1 f_1 + ... + w_n f_n)}} \right)$$

  • настраивается аналогично, но качество существенно выше: 0.8713

4. Линейные методы

- Для экспериментов был использован пакет LIBLINEAR, в котором реализованы алгоритмы логистической регрессии и SVM для больших разреженных матриц.
  В логистической регрессии можно выбрать тип регуляризации: L1 или L2, а в SVM – вид функции потерь: L1 или L2.

4. Линейные методы

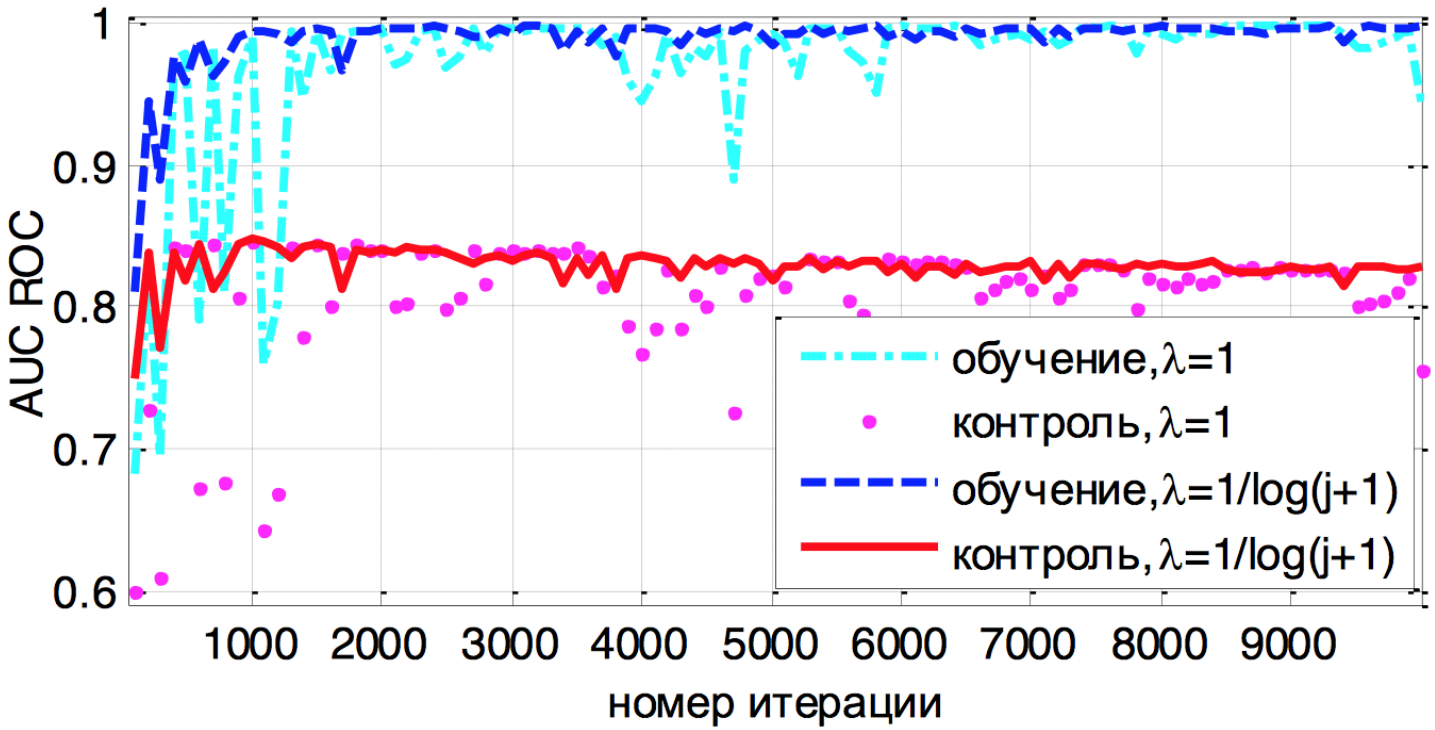

Рис.: Качество при настройке простейшего персептронного алгоритма

4. Линейные методы

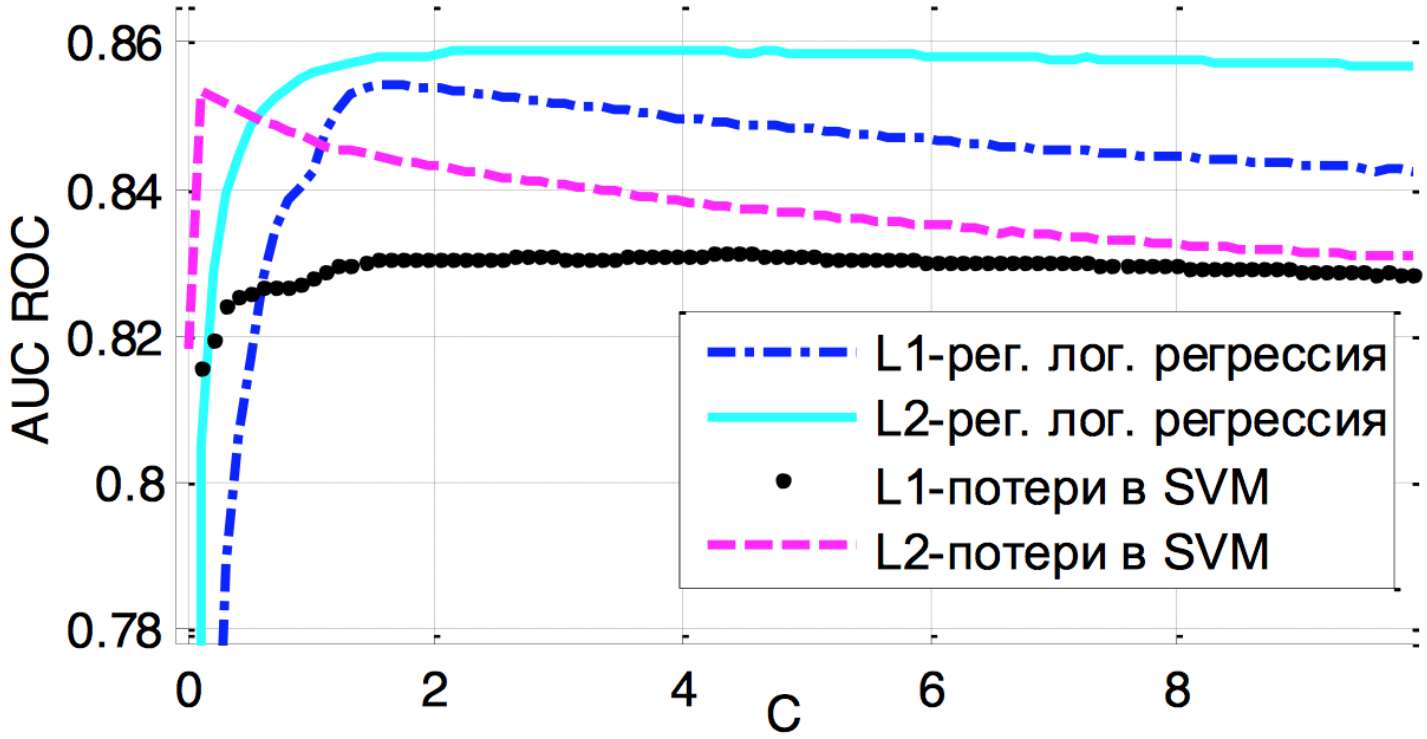

Рис.: Качество алгоритмов пакета LIBLINEAR после one-hot-кодирования признаков

4. Линейные методы

- Лучшее качество среди линейных алгоритмов показала логистическая регрессия с L2-регуляризацией
- Селекция признаков не проводилась
- При отборе признаков логистическая регрессия является лучшим алгоритмом для решения рассматриваемой задачи

Качество логистической регрессии на конъюнкциях разных порядков.

Порядок конъюнкции	1	2	3	4
ROC AUC	0.8591	0.8703	0.8713	0.8704

5. Байесовские алгоритмы и их обобщения

- Значения признаков f_j заменим на g_j - оценки принадлежности классу 1, полученные по j-ому признаку:

$$g_j = \begin{cases} \frac{|I_j(f_j) \cap Y_1|}{|I_j(f_j)|}, & f_j \in F_j \\ \Delta_j, & f_j \notin F_j \end{cases} \tag{4}$$

$I_j(f_j) = \{t \in \{1, 2, ..., m\} | f_j = f_{tj}\}$ - номера объектов, у которых значение j-ого признака равно f_j,
$Y_1 = \{t \in \{1, 2, ..., m\} | y_t = 1\}$ - номера объектов первого класса,
$F_j = \{f_{1j}, ..., f_{mj}\}$ - множество значений j-ого признака на обучении

- Часто на практике полагают:

$$g_j = \frac{|I_j(f_j) \cap Y_1| + \Delta_j * c}{|I_j(f_j)| + c}, \tag{5}$$

c - коэффициент регуляризации

5. Байесовские алгоритмы и их обобщения

- Обобщенный алгоритм: от признакового описания $(f_1, ..., f_n)$ перейдем к признаковому описанию $(\phi(g_1), ..., \phi(g_n), \gamma_1, ..., \gamma_n)$, где $\phi : R \to R$ - некоторая функция,

$$\gamma_j = \begin{cases} 0, & f_j \in F_j \\ 1, & f_j \notin F_j \end{cases} \tag{6}$$

5. Байесовские алгоритмы и их обобщения

Преимущества

- Можно использовать любые стандартные методы решения задачи
- Признаковое описание увеличивается незначительно
- Кодировки легко интерпретировать: первая группа признаков - оценка принадлежности к классам, вторая группа - индикаторы новых категорий
- Быстро удается построить алгоритм неплохого качества

- Переобучение
- Оценки вероятности могут быть некорректными из-за дефицита информации
  На практике обычно отбрасывают небольшие категории:

$$g_j = \begin{cases} \frac{|F_j \cap Y_1|}{|F_j|}, & |I(f_j)| \geq r \\ 0, & |I(f_j)| < r \end{cases} \tag{7}$$

- Не учитываются связи между признаками
  Признаки можно пополнять конъюнкциями, но при этом увеличивается число небольших категорий

5. Байесовские алгоритмы и их обобщения

- $\phi(x)$-1-алгоритм - алгоритм, который решает задачу линейным методом в новом признаковом пространстве
- На практике в качестве $\phi : R \to R$ часто оказывается лучше использовать тождественные функции или фукнции вида $\phi(g) = g^k$
- Один из лучших алгоритмов: ϕ-2-алгоритм

$$L(f_1, ..., f_n) = \frac{\sum_{j=1, f_j \in F_j}^{n} w_j \phi(g_j)}{\sum_{j=1, f_j \in F_j}^{n} w_j} \quad (8)$$

Формула некорректна, если $f_j \notin F_j$ для всех $j \in \{1, 2, .., n\}$. В этом случае можно в качестве ответа выдавать $\frac{y_1 + ... + y_m}{m}$

- Качество обобщений байесовских алгоритмов

Порядок конъюнкции	1	2	3	4	5
x-1-алгоритм	0.8491	0.8814	0.8864	0.8878	0.8868
x-2-алгоритм	0.8475	0.8746	0.8801	0.8834	0.8842
$\log(x)$-1-алгоритм	0.8101	0.8234	0.8247	0.8242	0.8246
$\sqrt{x}$-1-алгоритм	0.8465	0.8726	0.8790	0.8809	0.8821

6. Методы, основанные на сингулярном разложении матрицы бинарных признаков

- Метод гребневой линейной регрессии
  Из усечённого сингулярного разложения $F' \approx U\Lambda V$,
  где F' - сильно разреженная матрица "объект-признак"
  можно использовать U как пизнаковую матрицу.
  Тогда можно решить задачу методом гребневой линейной регрессии:

$$Uw = Y$$

  где Y - целевой вектор, т.е.

$$w = (U^T U + \lambda I)^{-1} U^T Y$$

  I - единичная матрица λ - коэффициент регуляризации

6. Методы, основанные на сингулярном разложении матрицы бинарных признаков

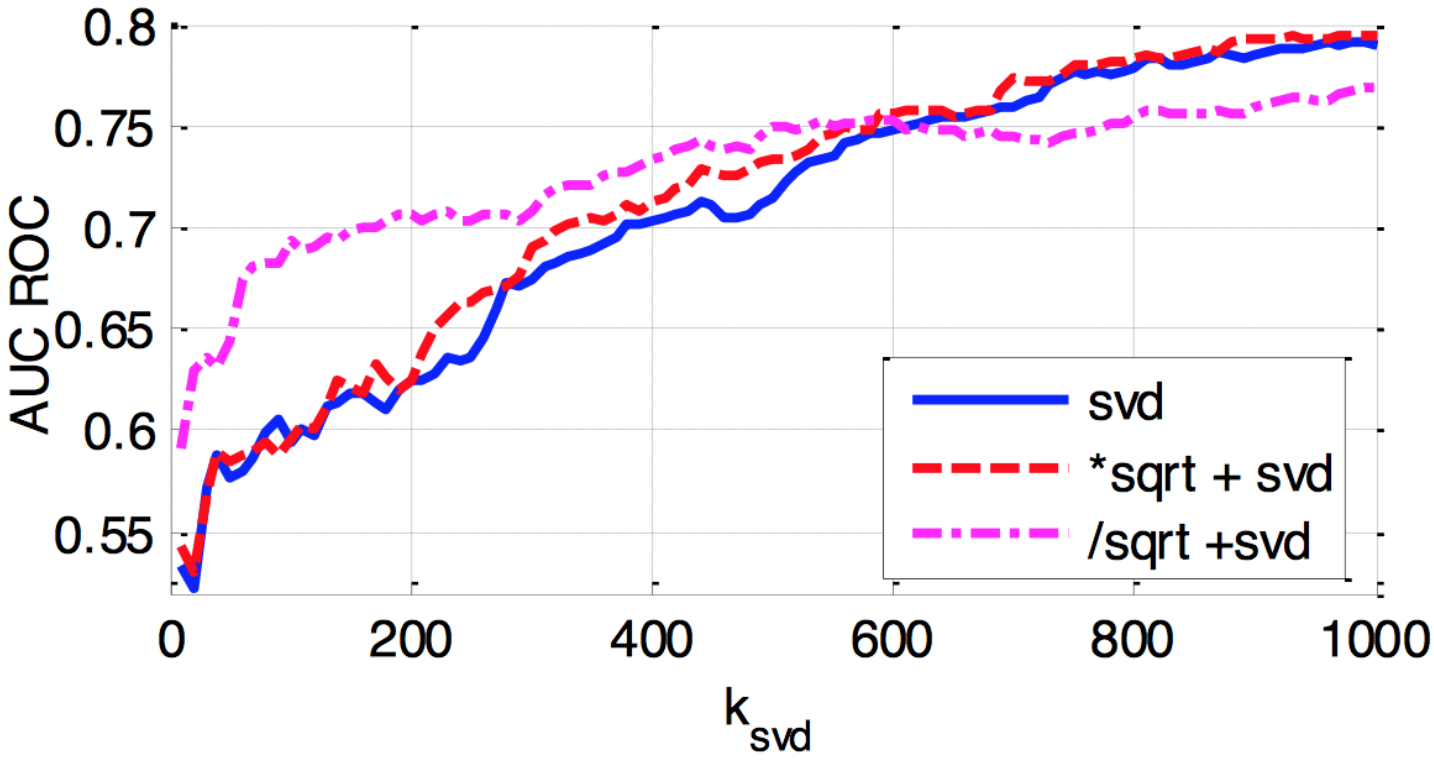

Рис.: Качество линейной регрессии после SVD от числа слагаемых в разложении.

6. Методы, основанные на сингулярном разложении матрицы бинарных признаков

На рис. показано также качество линейной регрессии после предварительных нормировок (деление и умножение на корень суммы элементов в столбце) вида:

$$\eta(||u_{ij}||_{m\times n}) = ||\frac{u_{ij}}{v_j}||_{m\times n}$$

и

$$\mu(||u_{ij}||_{m\times n}) = ||u_{ij}v_j||_{m\times n}$$

где $v_j = \sqrt{\sum_{t=1}^{m} u_{tj}}$

6. Методы, основанные на сингулярном разложении матрицы бинарных признаков

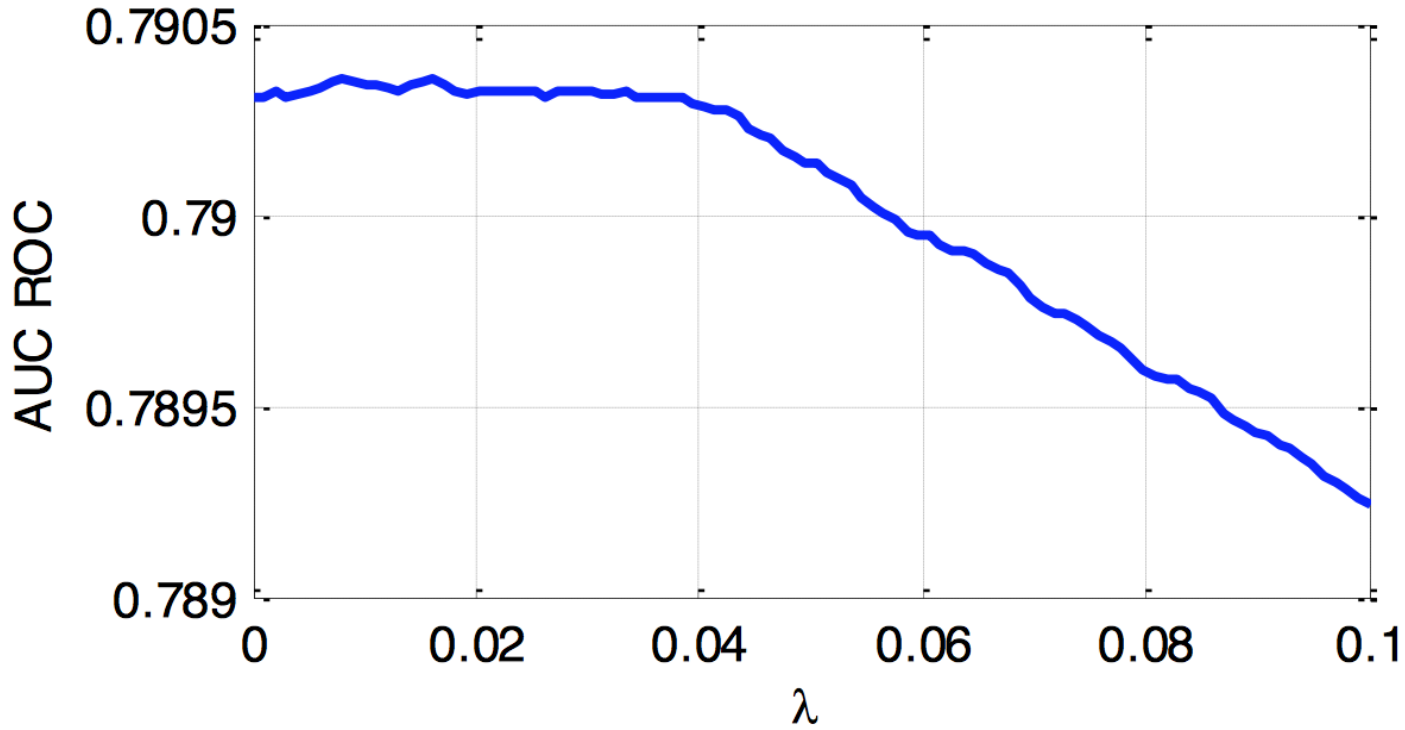

Рис.: Зависимость качества классификации от коэффициента регуляризации (при использовании 1000 слагаемых в SVD).

7. Методы, основанные на близости

- Оценка принадлежности классу

$$\Gamma_y(f_1, ..., f_n) = \frac{1}{N_y} \sum_{\Omega \in \Omega^*} \sum_{i: y_i = y} w^i w_\Omega B_\Omega((f_1, ..., f_n), (f_{i1}, ..., f_{in})), \quad (9)$$

где Ω^* – система опорных множеств: подмножеств множества признаков $\{1, 2, ..., n\}$,
w_Ω – вес опорного множества Ω,
w^i – вес i-го объекта из обучения,
N_y – нормирующий множитель,
$B_\Omega(f, g)$ – функция близости, которая оценивает сходство объектов f и g на опорном множестве Ω.

7. Методы, основанные на близости

- Ответ алгоритма

$$\sum_{i=1}^{m}\left(\frac{2y_i-1}{N_{y_i}}\sum_{\Omega\in\Omega^*}w_\Omega\prod_{j\in\Omega}I[f_j=f_{ij}]\right), I[f_j=f_{ij}]=\begin{cases}1, & f_j=f_{ij}\\ 0, & f_j\neq f_{ij}\end{cases} \quad (10)$$

- Немного изменим алгоритм

$$\frac{\sum_{i=1}^{m} r_i y_i}{\sum_{i=1}^{m} r_i}, r_i=\left(\sum_{\Omega\in\Omega^*}w_\Omega\prod_{j\in\Omega}I[f_j=f_{ij}]\right)^d \quad (11)$$

- Можно перейти к конъюнкциям: обозначим f_Ω - конъюнкция признаков с номерами из $\Omega\subset\{1,2,...,n\}$

$$r_i=\left(\sum_{\Omega\in\Omega^*}w_\Omega I[f_\Omega=f_{i,\Omega}]\right)^d$$

7. Методы, основанные на близости

- Параметры модели - w_Ω, d - настриваются методом покоординатного спуска.
- Качество метода, основанного на близости

Степень конъюкции	1	2	3	4
Качество	0.8681	0.8884	0.8900	0.8919

- Необходимо хранить всю обучающую выборку
- Долгая настройка параметров

8. Методы, основанные на тензорных разложениях

- Рассмотрим задачу с двумя категориальными признаками:

$$\begin{bmatrix} f_{11} & f_{12} \\ \vdots & \vdots \\ f_{m1} & f_{m2} \end{bmatrix} \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix}$$

учитывая, что первый признак принимает значения из $1, ..., n_1$, а второй из $1, ..., n_2$, задачу можно интерпретировать как:

$$Z = ||z_{ij}||_{n_1 \times n_2}$$

$$z_{f_{t1}, f_{t2}} = y_t$$

для всех $t \in 1, 2, ..., m$

8. Методы, основанные на тензорных разложениях

- Классификация объекта (f_1, f_2) эквивалентна определению значения элемента z_{f_1,f_2}
  Будем искать разложение $Z = UV$,
  где $U = ||u_{ij}||_{n_1 \times k}$
  $V = ||v_{ij}||_{k \times n_2}$
  минимизируя функционал

$$J = \sum_{t=1}^{m} e_t^2 + \lambda_1 \sum_{s=1}^{k} \sum_{i=1}^{n_1} u_{is}^2 + \lambda_2 \sum_{s=1}^{k} \sum_{j=1}^{n_2} v_{sj}^2$$

  где

$$e_t = \sum_{s=1}^{k} (u_{f_{t1},s} v_{f_{s},t2}) - y_t$$

8. Методы, основанные на тензорных разложениях

- Обычно используется два подхода минимизации:
  - Метод стохастического градиента (stochastic gradient descent)
  - Чередующиеся минимизации среднеквадратичной ошибки (alternating least squares)

8. Методы, основанные на тензорных разложениях

- Чередующиеся минимизации среднеквадратичной ошибки
  Метод основан на последовательной фиксации одной из матриц: $U = ||u_{ij}||_{n_1 \times k}$ или $V = ||v_{ij}||_{k \times n_2}$
- Выпуклая задача оптимизации
- Сходится медленнее, чем метод стохастического градиента

8. Методы, основанные на тензорных разложениях

- Метод стохастического градиента
  Метод основан на итерационном изменении настраиваемых параметров в направлении антиградиента.

$$\frac{\partial J}{\partial u_{is}} = 2 \sum_{t:f_{t1}=i} e_t v_{s,f_{t2}} + 2\lambda_1 u_{is}$$

$$\frac{\partial J}{\partial v_{sj}} = 2 \sum_{t:f_{t2}=j} e_t u_{f_{t1},s} + 2\lambda_2 v_{sj}$$

$U = ||u_{ij}||_{n_1 \times k}$ и $V = ||v_{ij}||_{k \times n_2}$ - случайные матрицы (начальное приближение), которые потом пересчитываются по формулам:

$$u_{is} = u_{is} - \alpha \sum_{t:f_{t1}=i} e_t v_{s,f_{t2}} - \lambda_1 u_{is}$$

$$v_{sj} = u_{sj} - \alpha \sum_{t:f_{t2}=j} e_t u_{f_{t1},s} - \lambda_2 v_{sj}$$

8. Методы, основанные на тензорных разложениях

- Аналогично в общем случае:
  Заданы n категориальных признаков, можно считать, что задана информация о m элементах многомерной матрицы размера $n_1 \times n_2 \times ... \times n_n$ – тензора n-го порядка.
  Нужно найти матрицы $U(r) = ||u_{ij}^r||_{n_r \times k}$ $r \in 1, ..., m$, чтобы минимизировать

$$J = \sum_{t=1}^{m} e_t^2 + \lambda_1 \sum_{r=1}^{n} \lambda_r \left(\sum_{i,j} (u_{ij}^r)^2 \right)$$

где $e_t = \sum_{s=1}^{k} \prod_{r=1}^{n} (u_{f_{r1},s}^r) - y_t$
Формулы для пересчета параметров:

$$v_{ij}^r = u_{ij}^r - \alpha \left(e_t \prod_{d=1, d \neq r}^{n} u_{f_{td},s}^d - \lambda u_{ij}^r \right)$$

8. Методы, основанные на тензорных разложениях

- Результат применения этого алгоритма в задаче Employee Access Challenge

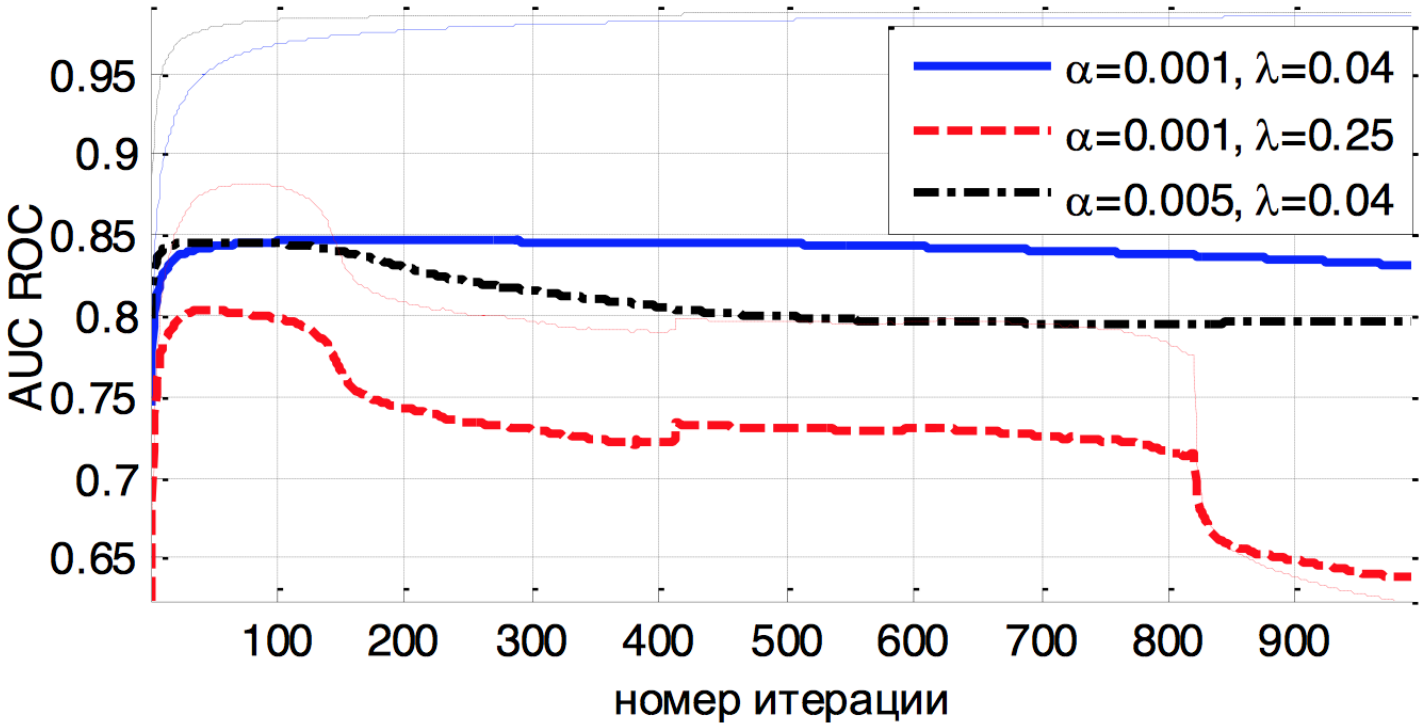

Рис.: Качество на обучении (тонкие графики) и контроле (толстые) методом, основанном на тензорном разложении

9. Методы, основанные на кодировках признаков

- Случайная кодировка

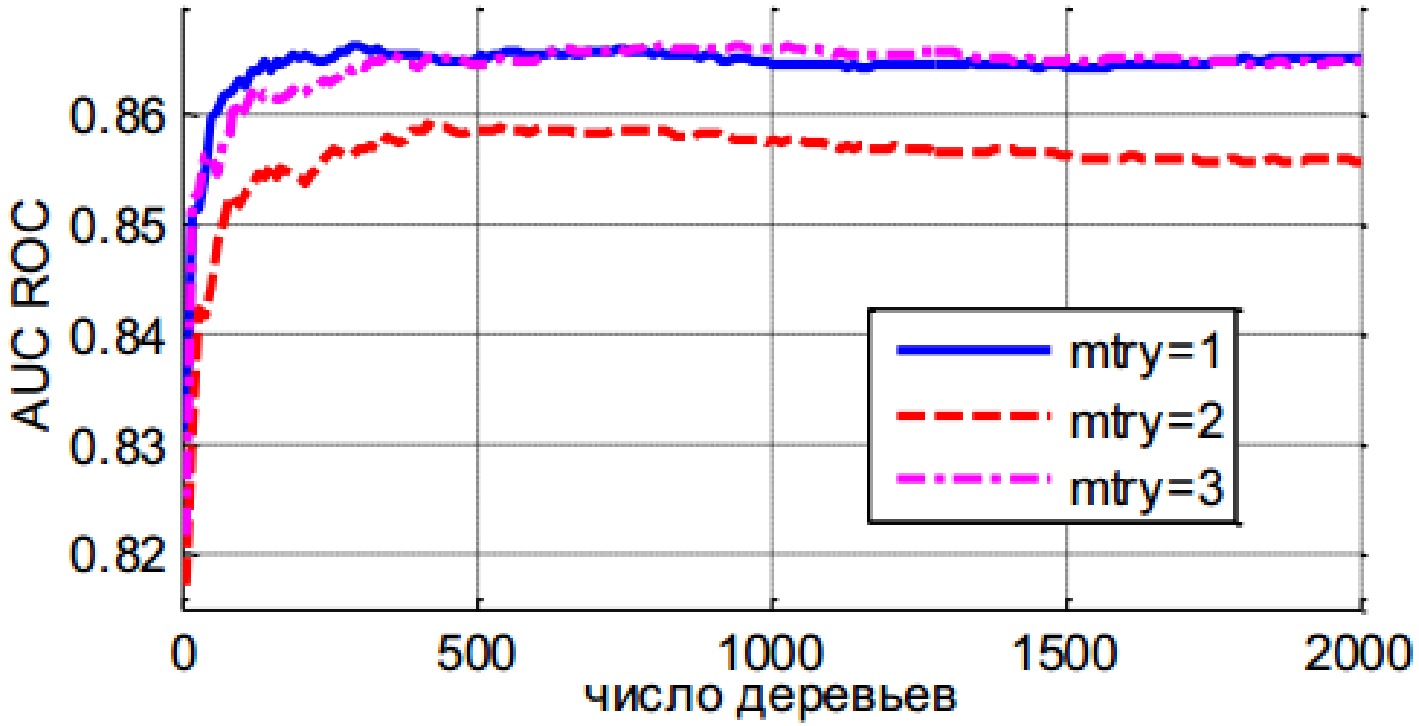

Рис.: Качество случайного леса от числа деревьев при случайных кодировках

9. Методы, основанные на кодировках признаков

- Случайная кодировка

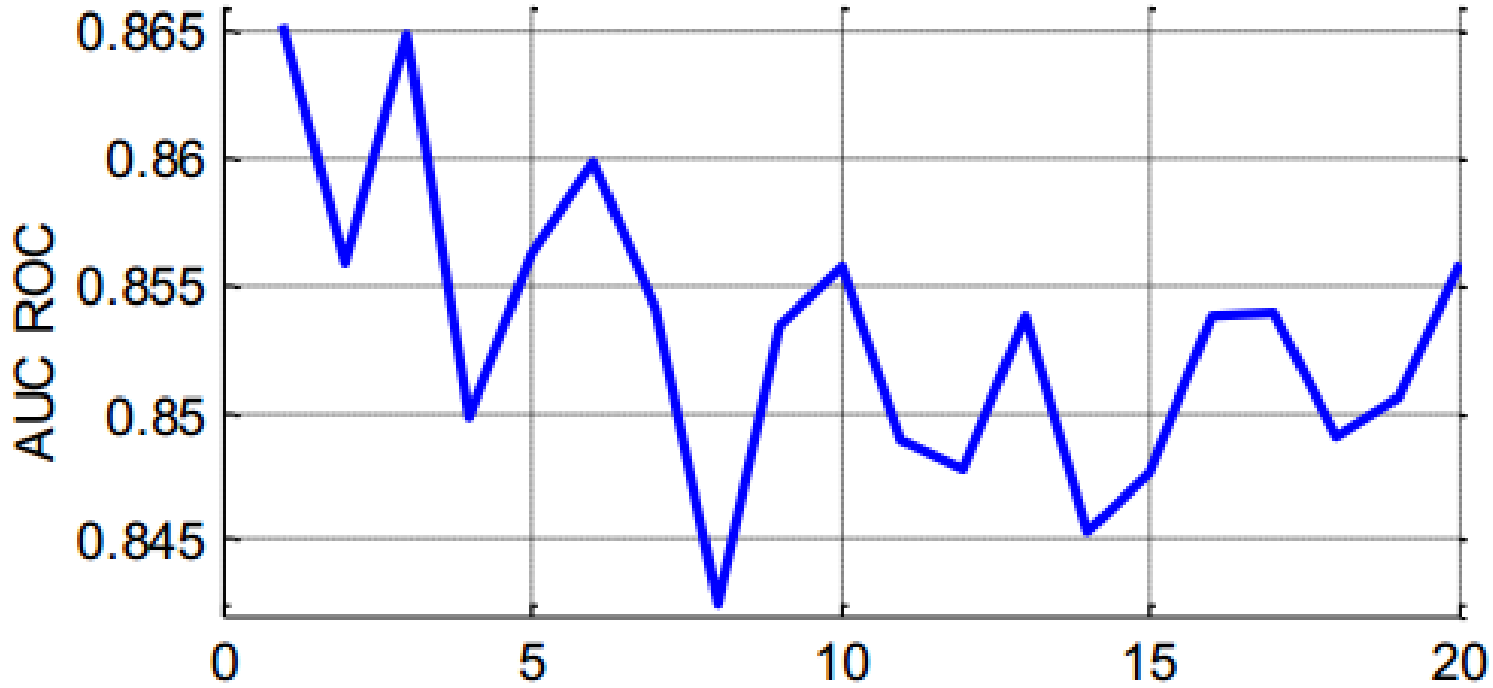

Рис.: Качество случайного леса от параметра mtry при случайных кодировках

9. Методы, основанные на кодировках признаков

- Категориальные + Вещественные
  Ф - некоторое множество вещественных функций, в котором для произвольного натурального числа k есть ровно одна функция k переменных.
  При этом все функции симметричные, т.е. для любой функции $\varphi \in$ Ф от k переменных
  $\varphi(x_1, ..., x_k) = \varphi(x_{\sigma(1)}, ..., x_{\sigma(k)})$
  для любой перестановки σ.
  Пример: множество сумм
  $\varphi(x_1, ..., x_k) = x_1 + ... + x_k$
  Для кодирования значения f_j j-го категориального признака выбираем
  $I = \{t \in \{1, 2, ..., m\} | f_{tj} = f_j\}$,
  вещественный признак - s-ый
  Кодируем значение f_j значением подходящей функции из Ф (т.е. функции от $|I|$ переменных) от значений $f_{is}, i \in I$

9. Методы, основанные на кодировках признаков

- Категориальные
  Первый признак - кодируемый, $\{1, 2, ..., n_1\}$
  Второй признак - кодирующий, $\{1, 2, ..., n_2\}$
  $P = ||p_{ij}||_{n_1 \times n_2} : p_{ij} = |\{t \in \{1, 2, ..., m\} | f_{t1} = i, f_{t2} = j\}|$
  Сделаем неполное сингулярное разложение матрицы $P \approx U \Lambda V$ (первые k слагаемых)
  Получаем k различных кодировок: в t-ой кодировке заменяем значение i на it-ый элемент матрицы U.

- Категориальные

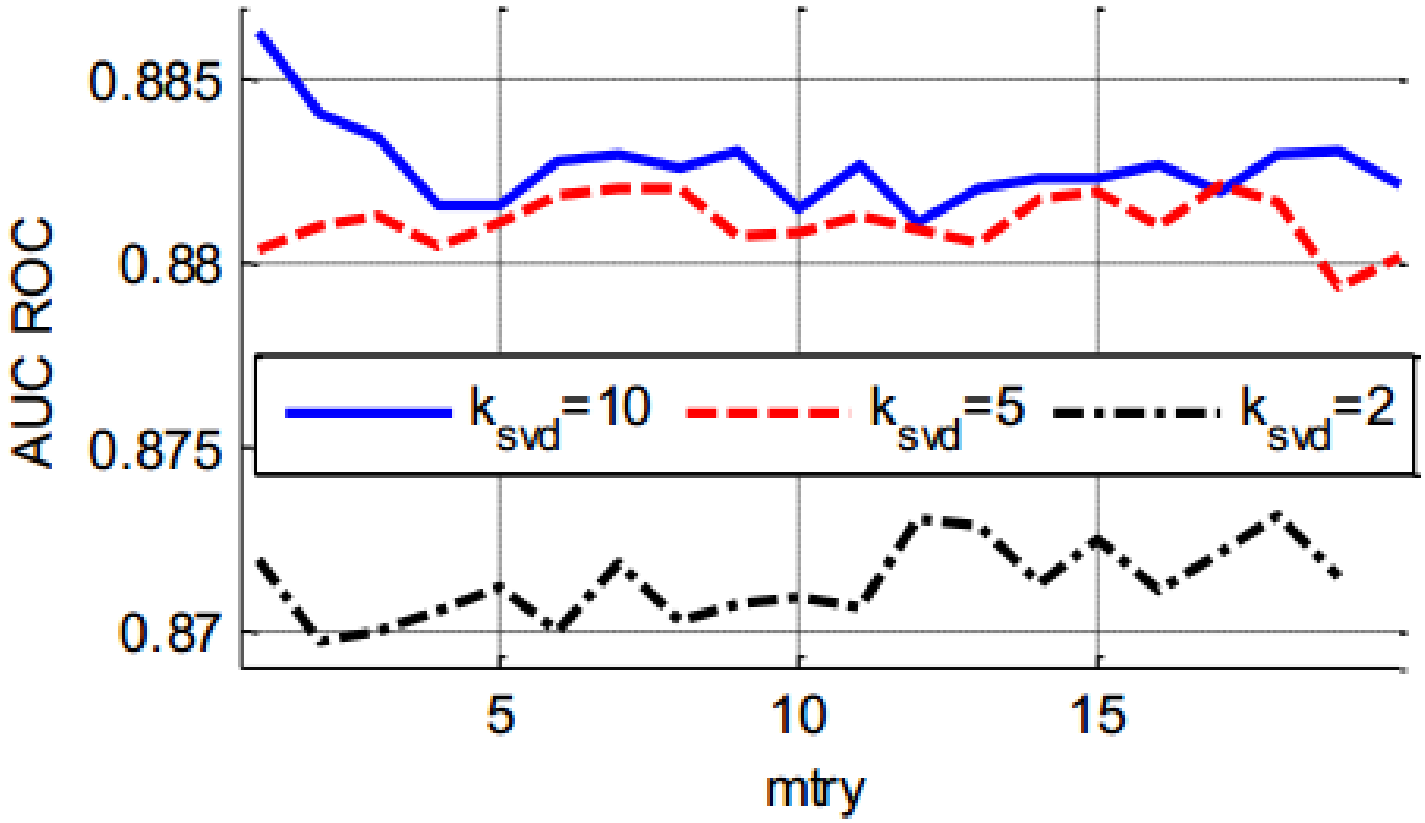

Рис.: Зависимость качества от параметра mtry случайного леса

9. Методы, основанные на кодировках признаков

- Категориальные

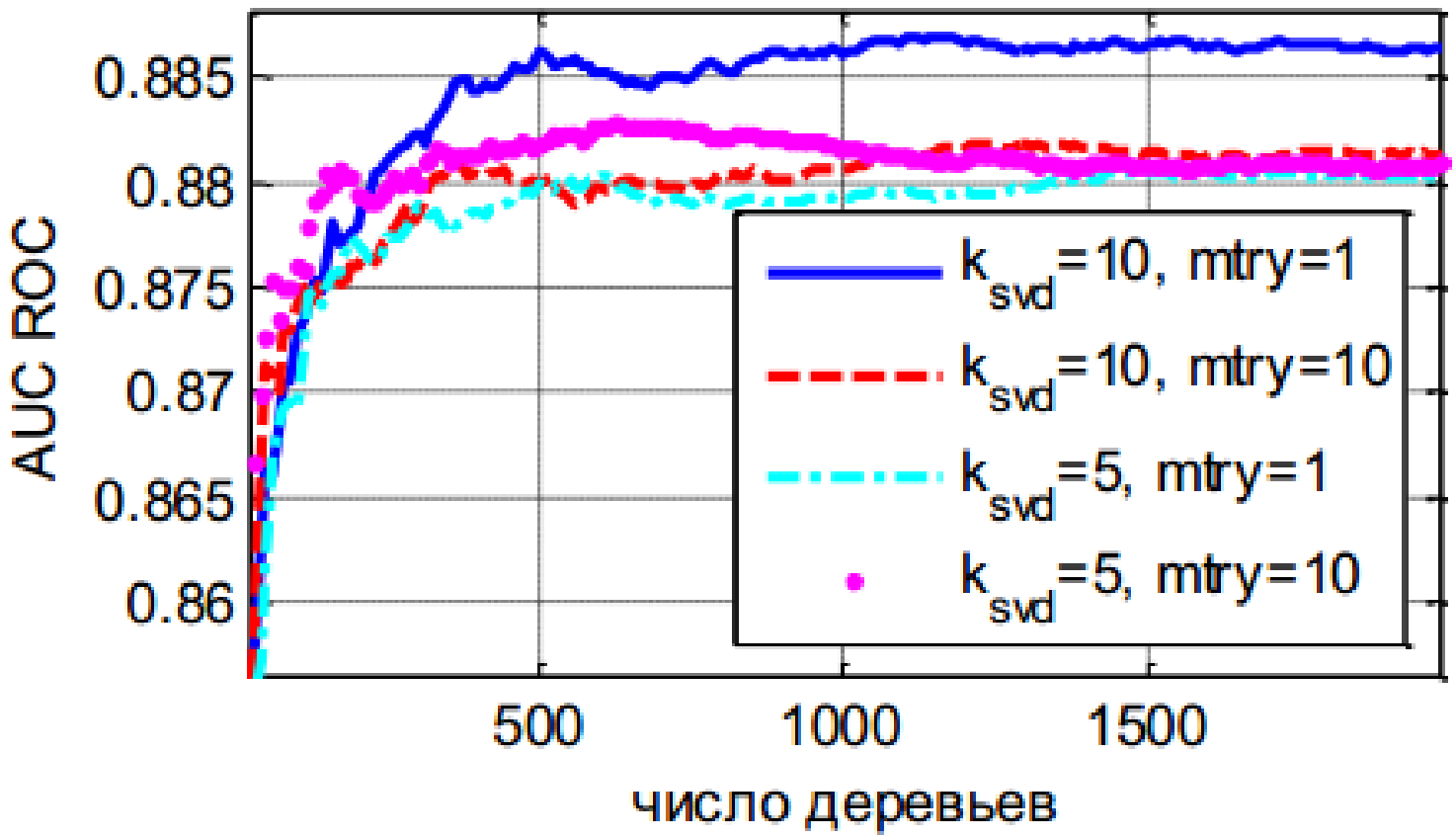

Рис.: Зависимость качества от числа деревьев в случайном лесе

10. Выпуклые комбинации алгоритмов

алгоритмы	4	5	7	8	9
4	0.8713	0.8914	0.8919	0.8731	0.8879
5		0.8878	0.8972	0.8884	**0.8974**
7			0.8919	0.8924	0.8919
8				0.8453	0.8872
9					0.8863

Рис.: Качество лучших линейных комбинаций

алгоритмы	4	5	7	8	9
4	1	**0.9523**	0.9543	0.9696	0.9568
5		1	0.9989	0.9858	0.9983
7			1	0.9866	0.9989
8				1	0.9880
9					1

Рис.: Корреляция ответов алгоритмов

10. Выпуклые комбинации алгоритмов

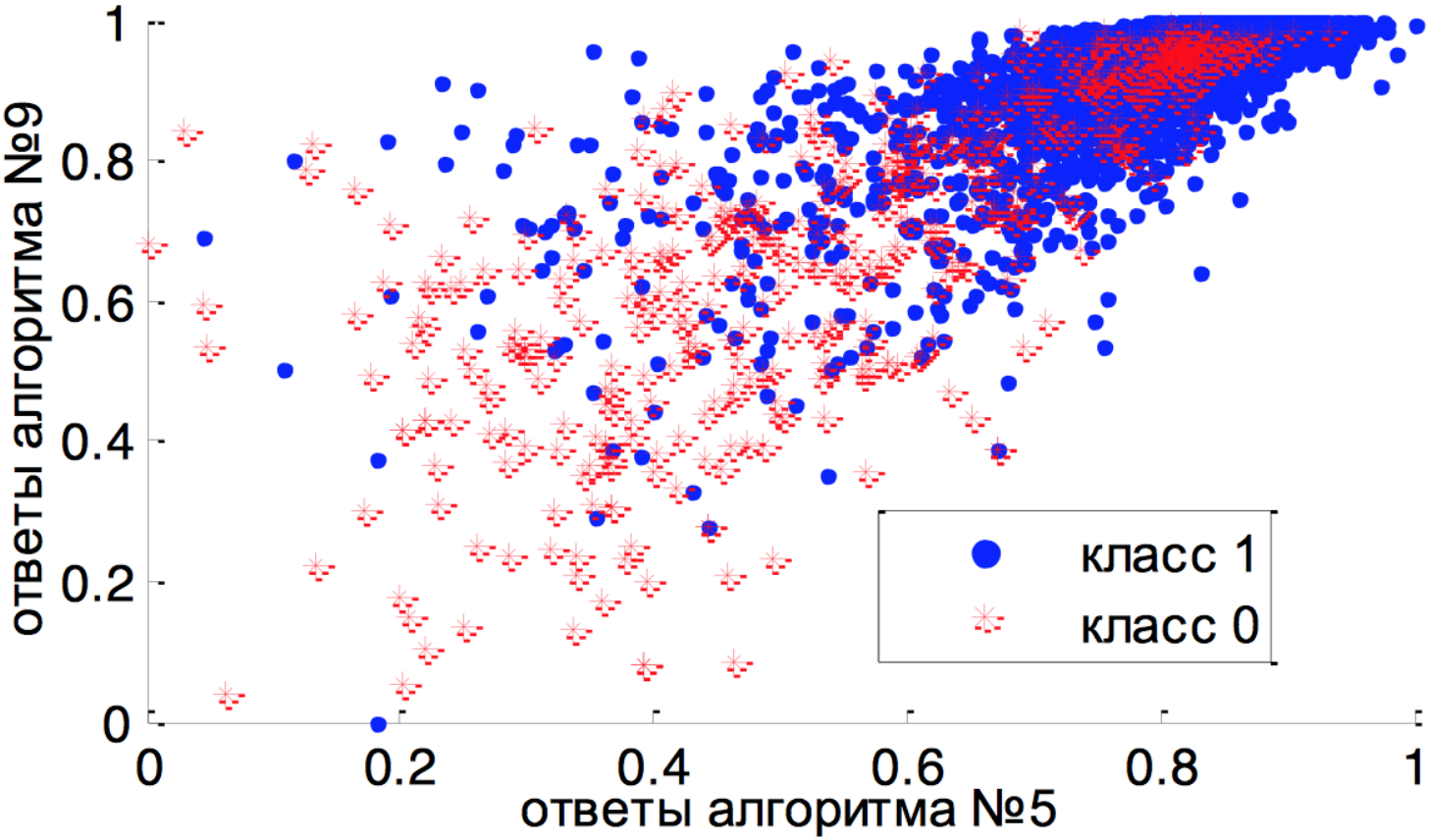

Рис.: Ответы двух алгоритмов

10. Выпуклые комбинации алгоритмов

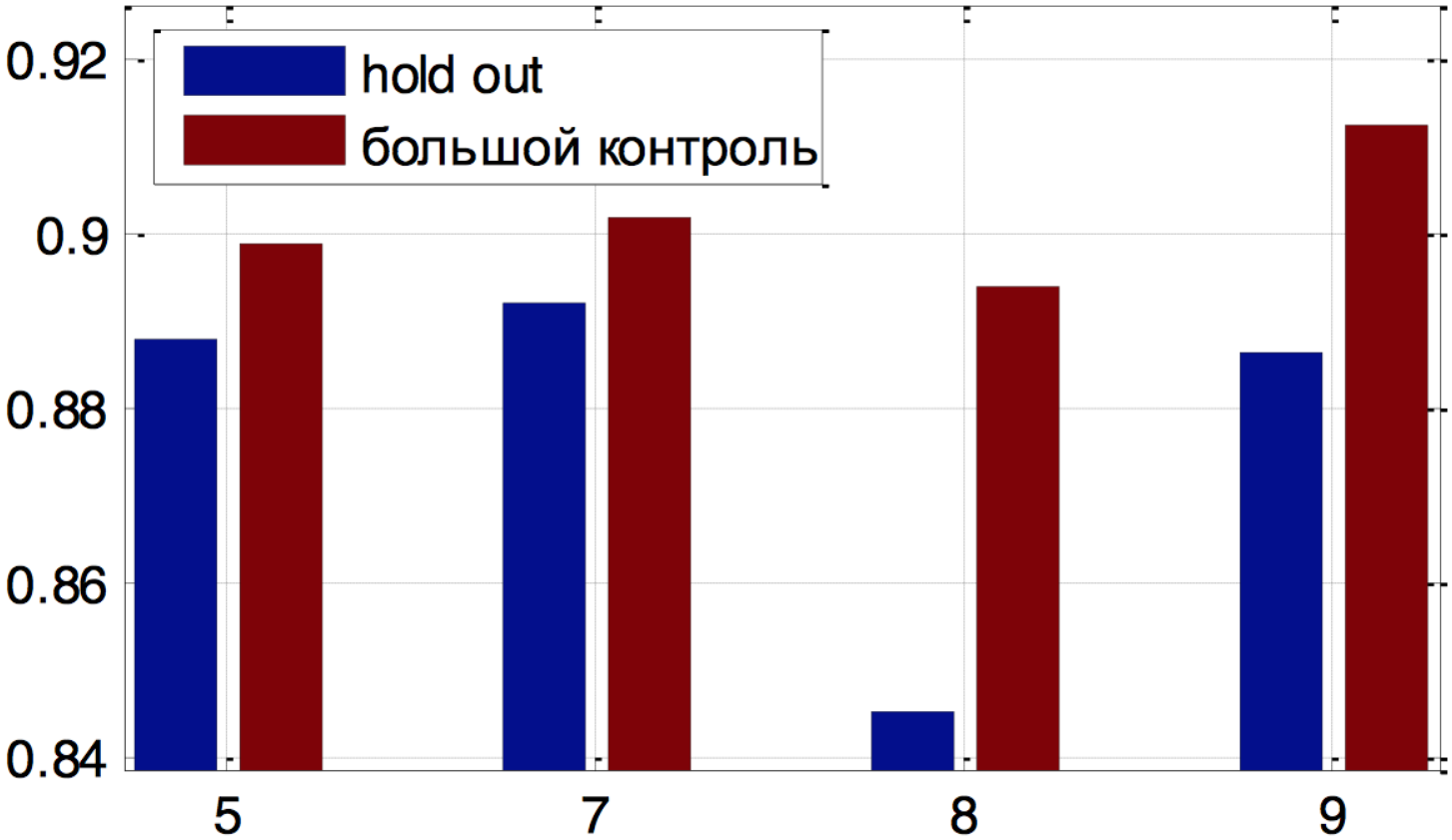

Рис.: Качество при проверке на переобучение