Отбор (селекция) признаков

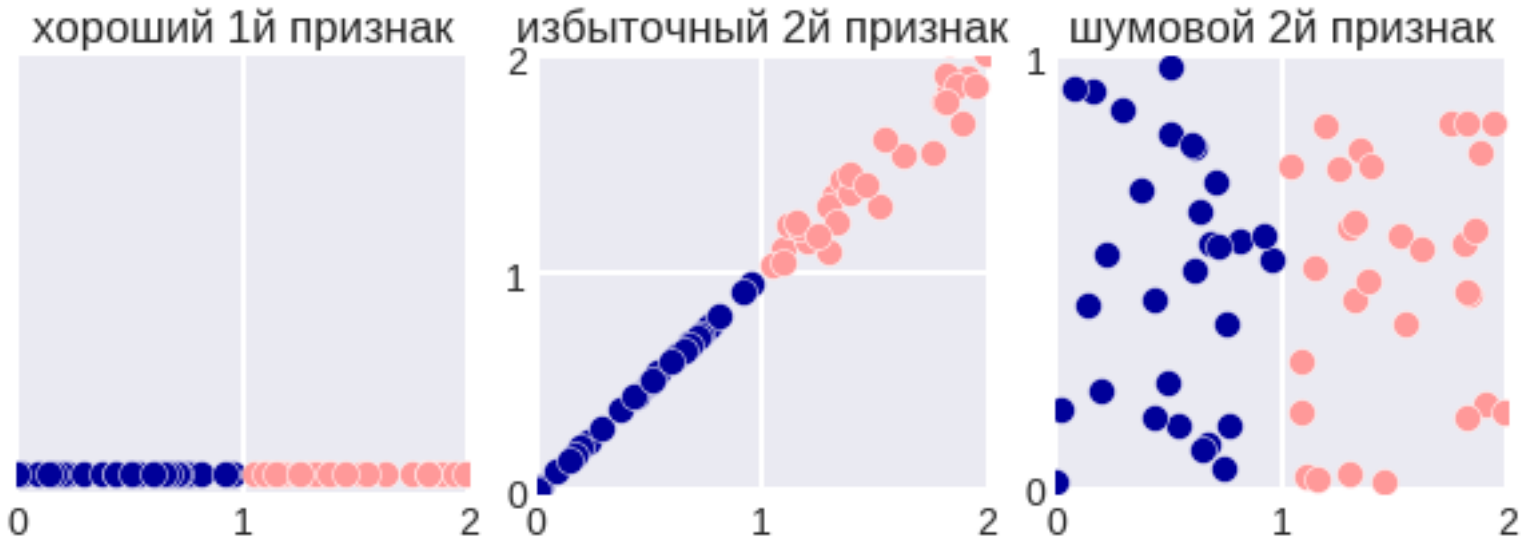
Дьяконов А.Г.

Московский государственный университет имени М.В. Ломоносова (Москва, Россия)

Отбор признаков (Feature Selection)

– нахождение оптимального подмножества признаков, в соответствии с некоторым критерием

– процесс удаления избыточных и нерелевантных признаков



Отбор признаков (Feature Selection)

Причины

- интерпретация
- скорость работы алгоритмов
- борьба с переобучением (корреляции для линейных методов)
- повышения качества (если много шума)

Классификация методов

Фильтры (filter methods) – не ориентированы на конкретные модели алгоритмов машинного обучения

Обёртки (wrapper methods) – ориентированны на конкретные модели алгоритмов машинного обучения

Встроенные (embedded methods) – являются частью методов МО

Нет волшебного алгоритма!

Фильтры

Рассмотрим дискретные признаки (принимают конечное число значений)

Фильтры

Энтропия:

$$H(X) = -\sum_{x_i \in X} p(x_i) \log p(x_i)$$

Условная энтропия (conditional entropy)

Энтропии, вычисленные

для фиксированных значений признаков:

$$H(Y \mid X) = \sum_{x_i \in X} p(x_i) H(Y \mid X = \{x_i\}) =$$

$$= \sum_{x_i \in X} p(x_i) \sum_{y_j \in Y} p(y_j \mid x_i) \log(p(y_j \mid x_i))$$

Взаимная информация (Information Gain, Mutual Information)

Насколько более чётко определена Y, если знаем X

$$I(Y, X) = H(Y) - H(Y \mid X) =$$

$$= \sum_{y_j \in Y} \sum_{x_i \in X} p(x_i, y_j) \log \frac{p(x_i, y_j)}{p(x_i)(y_j)}$$

Фильтры

$$MI = \sum_{x_i \in X} \sum_{y_j \in Y} p(x_i, y_j) \log \frac{p(x_i, y_j)}{p(x_i) p(y_j)}$$

Для независимых признаков = 0

Предпочитает выбирать признаки с большим числом значений

Фильтры

Ожидаемая вероятность в предположении независимости:

$$P(A \cap B) = P(A) \cdot (B)$$

	Y=0	Y=1	
X=0	6	4	Σ=10
X=1	14	16	Σ=30
	Σ=20	Σ=20	Σ=40

$$\text{expected} = \frac{10}{40} \cdot \frac{20}{40} \cdot 40 = 5$$

$$\chi^2 = \sum \frac{(\text{observed} - \text{expected})^2}{\text{expected}}$$

Фильтры

Разные статистики признаков:

1. Низкая оценка дисперсии – почти константный признак

2. t-оценка (для задачи с 2 классами)

$$\frac{|\mu_1 - \mu_2|}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}}$$

3. Хи-квадрат (см. выше)

$$\chi^2 = \sum_i \sum_j \frac{(n_{ij} - \mu_{ij})^2}{\mu_{ij}}, \mu_{ij} = \frac{\sum_t n_{it} \sum_t n_{tj}}{n}$$

Фильтры

4. Корреляция между признаками
корреляция с целевым $\Rightarrow$ хороший
корреляция с другим $\Rightarrow$ один можно удалить

5. Использование других мер качества
AUC-ROC-признака

Фильтры

+ сложность линейно зависит от числа признаков

– не учитываем алгоритм

– оцениваем отдельные признаки (дальше исправим)

Jundong Li, Kewei Cheng, Suhang Wang, Fred Morstatter, Robert P. Trevino, Jiliang Tang, Huan Liu Feature Selection: A Data Perspective

Обёртки

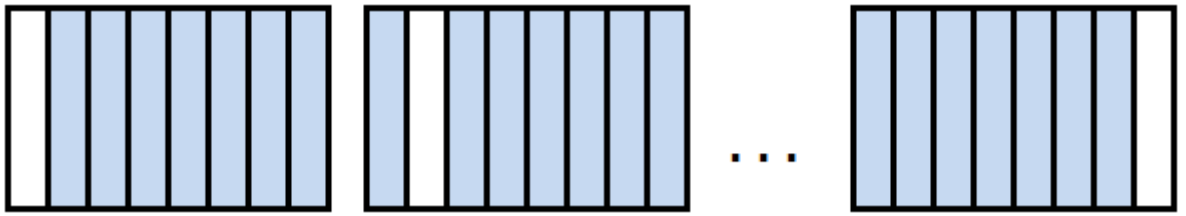
Запускаем алгоритм на наборе признаков

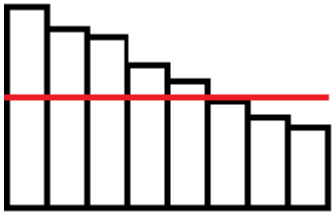
1	1	1	1	0	1	1	1
3	1	1	1	0	2	1	1
2	1	2	2	3	1	1	2
1	1	2	2	2	0	2	2
0	2	1	3	1	1	1	1
1	2	1	3	1	1	1	1
1	2	2	1	1	0	1	3

$$Q(\{f_1,\ldots,f_k\}) = Q(A(X[:,[f_1,\ldots,f_k]],y))$$

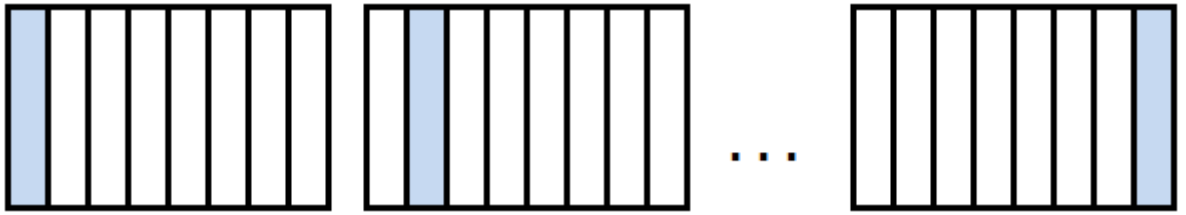
Обёртки на практике

Исключение (перестановка) по одному



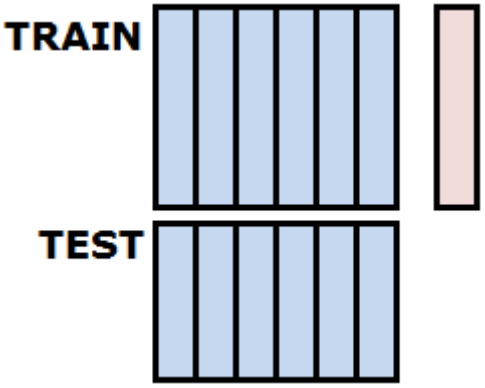


Качество по отдельным признакам

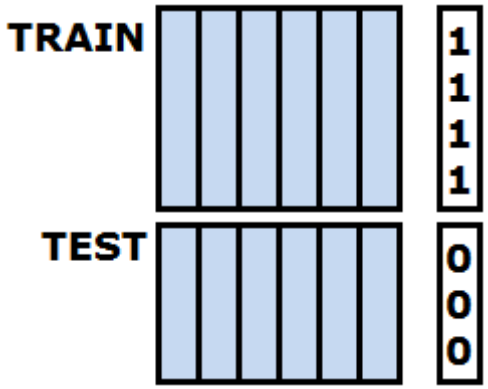


Стабильность признака

Исходная задача



Новая задача

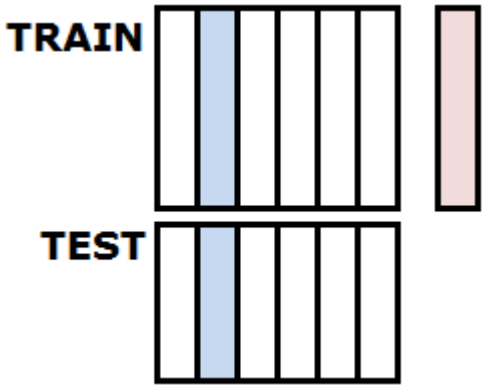


AUC ROC

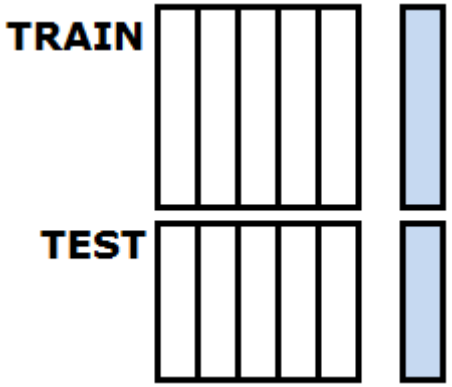
- **Получаем оценку схожести обучения и контроля**
- **Важности признаков – оценки их нестабильности**
- **Отбор признаков – поиск стабильного признакового пространства**

Зависимость (выводимость) признака

Исходная задача



Новая задача



Функционал?

- **Получаем оценку зависимости признака от остальных**
- **Можем последовательно удалять лишние признаки**
- **Можно добавлять признаки к базовым**

Отбор признаков как задача глобальной оптимизации

1	1	1	1	0	1	1	1
3	1	1	1	0	2	1	1
2	1	2	2	3	1	1	2
1	1	2	2	2	0	2	2
0	2	1	3	1	1	1	1
1	2	1	3	1	1	1	1
1	2	2	1	1	0	1	3

0	1	1	1	0	1	0

Максимизация функции

$$f : \{0,1\}^n \to \mathbf{R}$$

Решение задач глобальной оптимизации

Заведомо нет лучшего алгоритма

Пример функции с точечным носителем

1. Перебор
2. Направленный поиск
3. Стохастическая оптимизация

Полный перебор

Может не завершиться

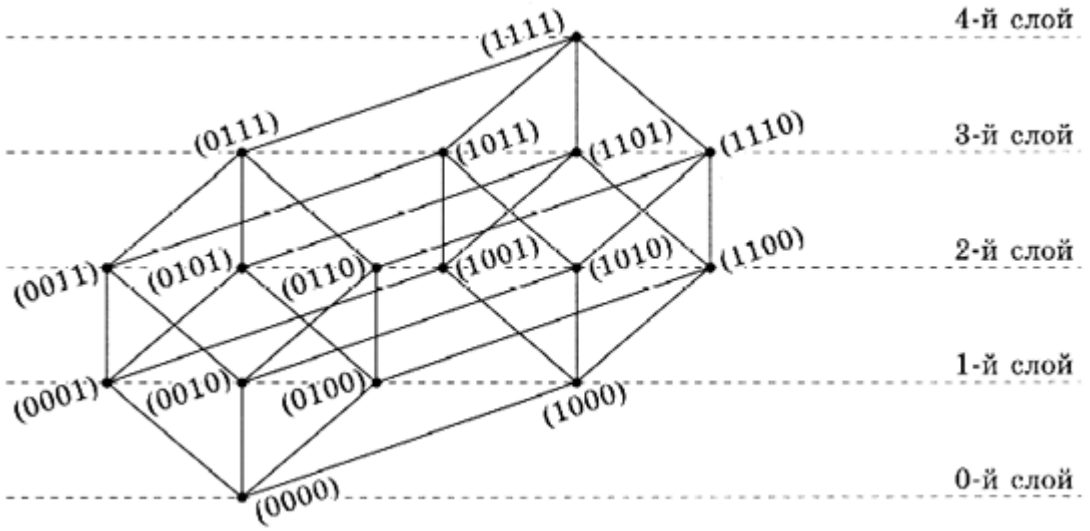
Грамотно организовать:

- **сначала потенциально лучшие точки**
- **устанавливать свойства функции (монотонность, несущественность, эквивалентность переменных и т.п.)**

Пример

- **Перебор всех троек признаков**
- **Удалить те, которые не попали в хорошие подпространства**

Направленный поиск

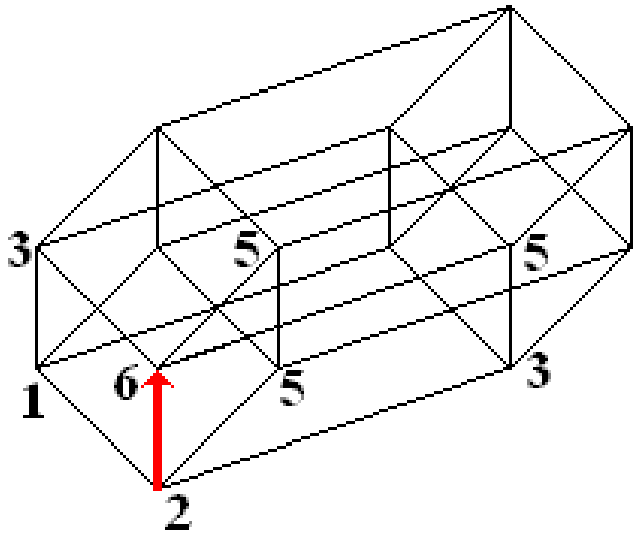


Точки для перебора выбираем из окрестности уже исследованных точек

- **градиентный алгоритм**
- **симуляция отжига**
- **метод луча (beam search)**
- **локальный поиск**

Направленный поиск

Градиентный алгоритм

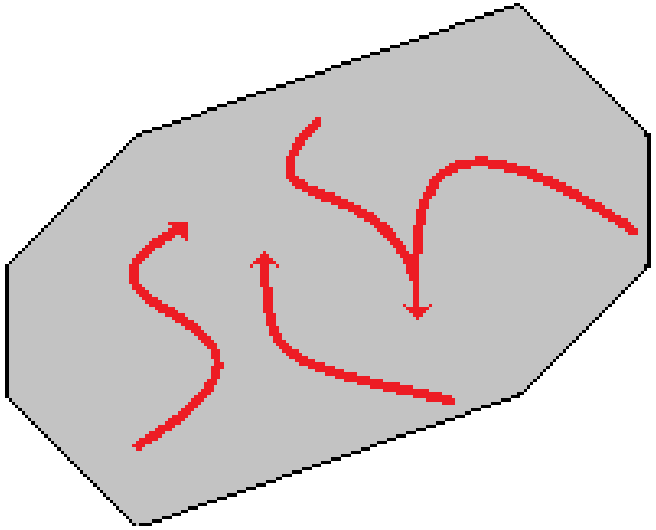


1. Начинаем со случайной точки
2. Ищем в окрестности текущей точки наибольшее значение
3. Переходим в соответствующую точку

Останавливаемся в локальном максимуме

Направленный поиск

Градиентный алгоритм (усовершенствования)



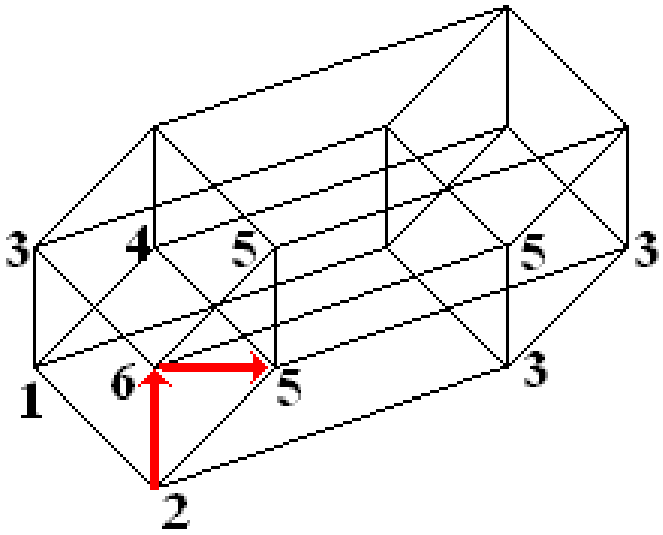
1. Перезапуски
2. Параллельные запуски с переключением на перспективные ветки
3. Продолжать движение в локальных максимумах – **симуляция отжига**

$$\exp([f(\tilde{z}^t) - f(\tilde{z})]/T)$$

4. Идти в сторону k лучших (запоминать, что посетили)
5. Также собирать информацию о функции

Направленный поиск

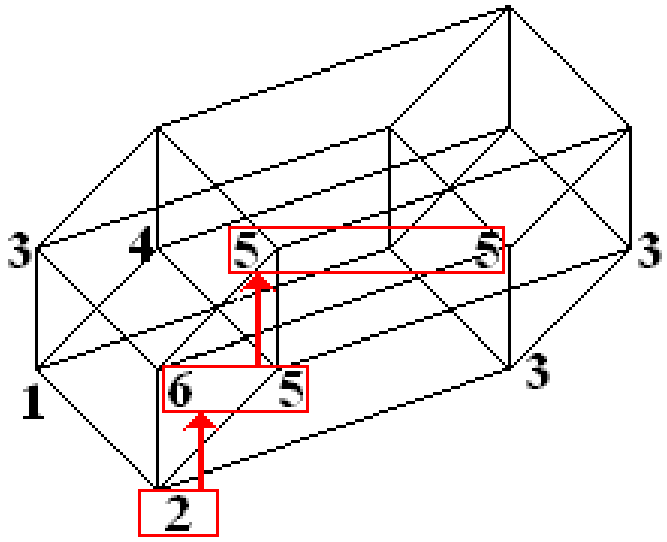
Метод луча



Храним k лучших точек

Направленный поиск

Локальный поиск



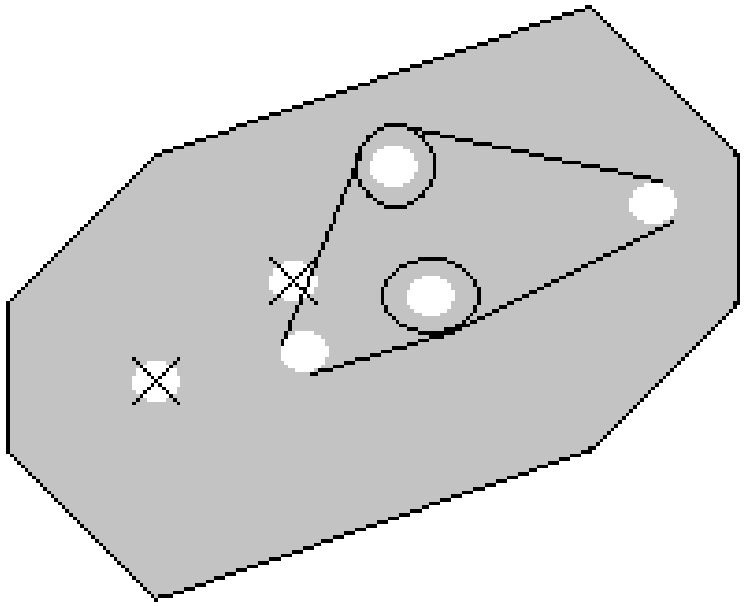
1. Стартуем с {(0,0,…,0)}

2. Среди соседей текущего множества выбираем k лучших соседей верхнего уровня

Стохастическая оптимизация

Генетический алгоритм

1. Инициализация.
2. Селекция.
3. Скрещивание (размножение).
4. Мутации.
5. Переход к п. 2.



1010101110010
0110100101011

1010101101011

Стохастическая оптимизация

Генетический алгоритм (усовершенствования)

Селекция

- смерть от старения
- отбор по вероятности (оценка ~ вероятность смерти)
- смерть в боях (турниры)
- приход чужаков
- параллельно живущие популяции

Скрещивание

- разные схемы кроссовера
- разный выбор для скрещивания (все по парам, с вероятностями)
- алгоритм с постоянным числом индивидов (дети вместо родителей)
- конвейерная версия (0.1 – выживает, 0.9 – случайная пара переносит потомков)

Стохастическая оптимизация

Генетический алгоритм (усовершенствования)

Мутация

- лучшие не мутируют (элитаризм)
- вероятность мутации выше, если нет улучшений
- генетика + градиент

Кодирование особей

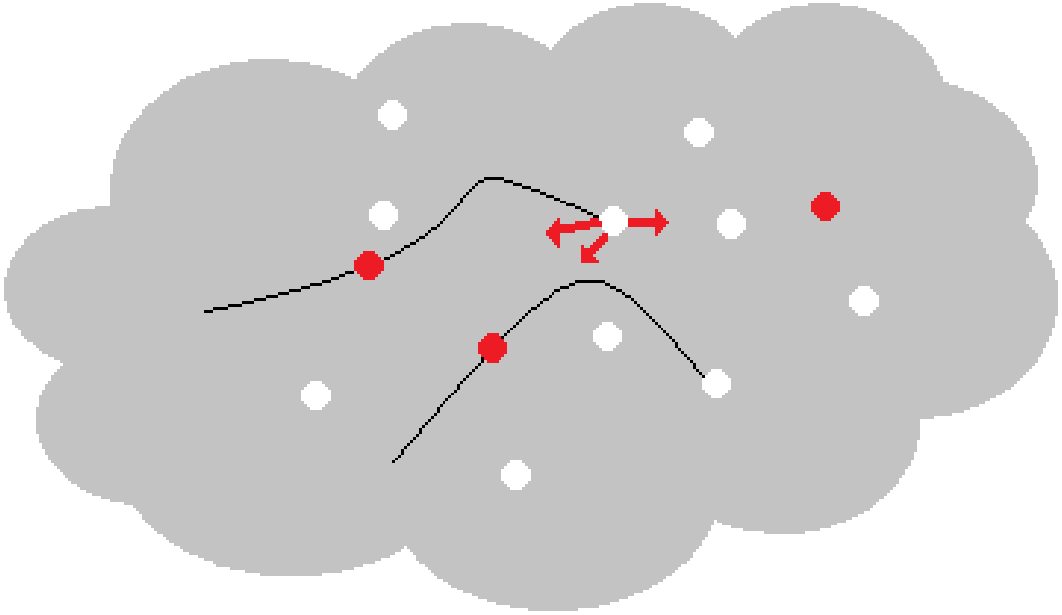
Число	Стандартный код	Код Грея
0	000	000
1	001	001
2	010	011
3	011	010
4	100	110
5	101	111
6	110	101
7	111	100

Стохастическая оптимизация

Роевой алгоритм

$$f : \mathbf{R}^n \to \mathbf{R}$$

- К своему максимуму
- К максимуму роя
- К максимуму подруги



$$x_{t+1}^i = x_t^i + \alpha(m^i - x_t^i) + \beta(m - x_t^i) + \gamma(m^j - x_t^i)$$

Стохастическая оптимизация

Полный перебор	**Высокая точность при экспоненциальном времени работы**
Направленный перебор	**Приемлемая точность, простая реализация, быстрая работа.** **Не полный перебор пространства.**
Стохастический перебор	**Хорошо работает при удачном выборе всех параметров, простой в реализации и модификации, избегает локальных максимумов.**

Sean Luke Essentials of Metaheuristics. — Lulu, 2009. — 235 p.

Стохастическая оптимизация

«Exploration vs. Exploitation»

задача исследования

Просмотреть как можно больше (новых) точек из всего пространства поиска

задача использования

Не пропустить хорошее решение и по максимуму использовать уже полученную информацию

Изменение параметров:

- радиус окрестности в градиентном алгоритме
- вероятность мутации
- ...

Встроенные методы

Линейная регрессия

$$Xw = y$$

Решение линейной регрессии

$$\| Xw - y \|^2 \to \min$$

Регуляризация по Тихонову

$$\| Xw - y \|^2 + \lambda \| w \|^2 \to \min$$

LASSO

$$\| Xw - y \|^2 + \lambda_2 \| w \|^2 + \lambda_1 \| w \| \to \min$$

```
from sklearn.linear_model import Ridge
clf = Ridge(alpha=1.0)
clf.fit(X, y)
```

Нормализация признаков!

Встроенные методы

Оценка важности в случайном лесе

1) Насколько уменьшает ошибку леса

2) Ухудшение на OOB при перемешивании значений

Ещё способы уменьшить число признаков…

Уменьшение размерности