Lec 11

14 ноября 2024 г. 18:55

1. FP Growth
2. Scanning DB and counting frequencies of occurances of every object in say transactions. Prune those with less support level than minimum one. Than reorganize them by frequency
3.

FP Growth (Frequent Pattern-Growth Strategy)

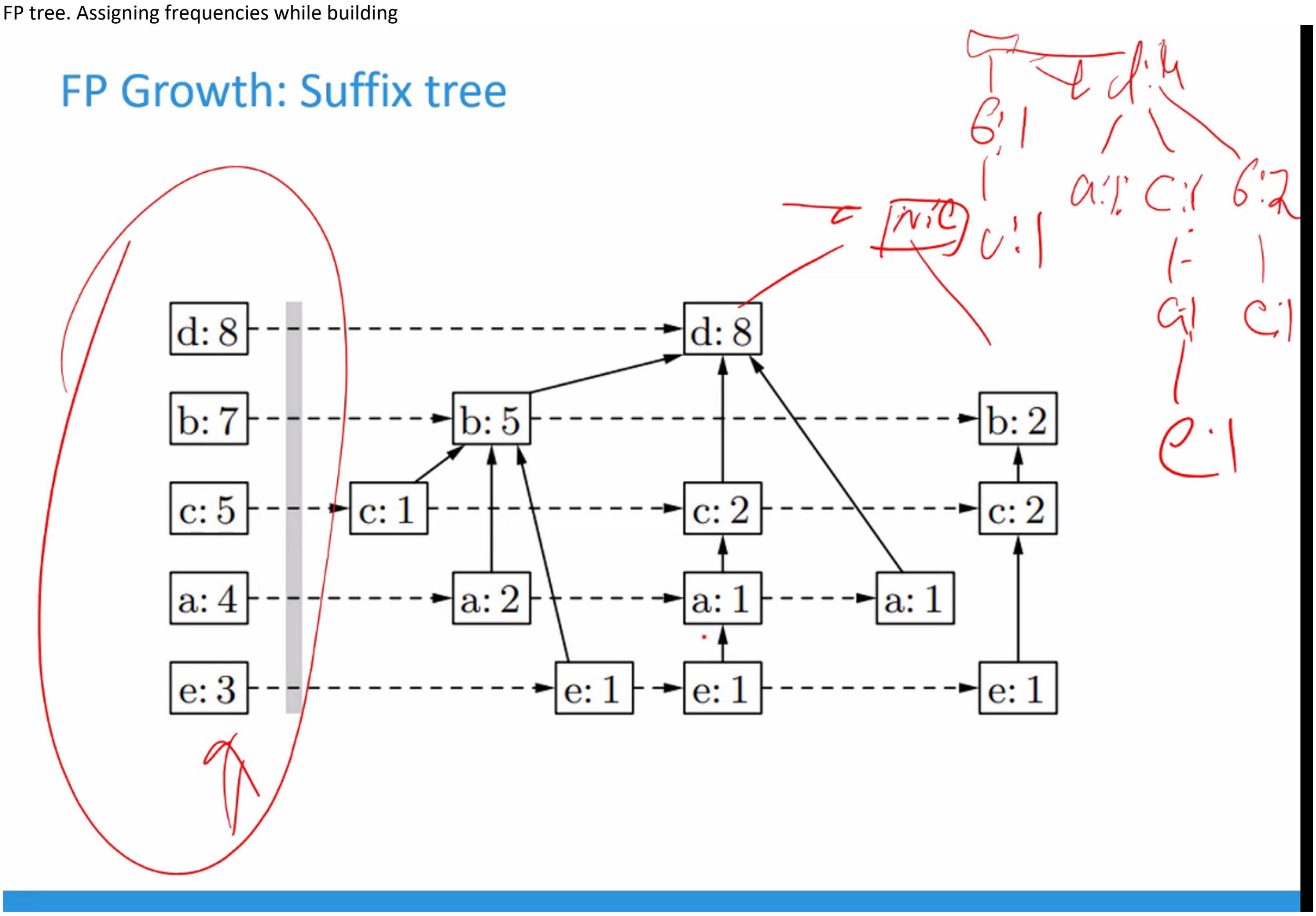
- Suffix trees (trie) construction with respect support

a d f
a c d e
b d
b c d
b c
a b d
b d e
b c e g
c d f
a b d

d	8
b	7
c	5
a	4
e	3
f	2
g	1

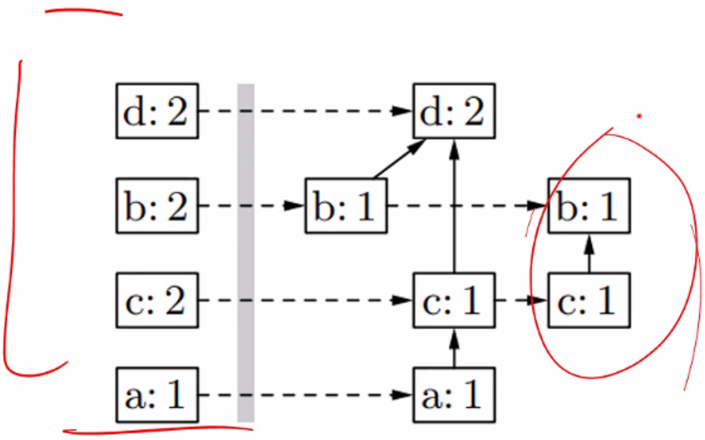
d a
d c a e
d b
d b c
b c
d b a
d b e
b c e
d c
d b a

4. FP tree. Assigning frequencies while building
5.



6. Haven't caught the slide, but the idea is that we are building the projection from bottom to up from the object we want to imply. In next picture there is a projection of rules that implies e. Frequencies are reassigned

Patterns generation

1. 

Let min support to be 2 transactions (it was 3, but I'm lazy to redraw picture ☺):

d -> e

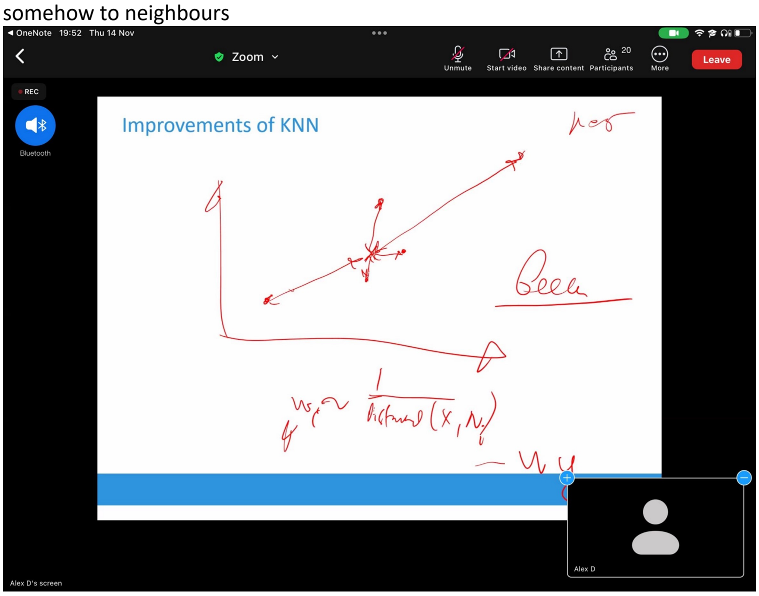
b -> e

c -> e

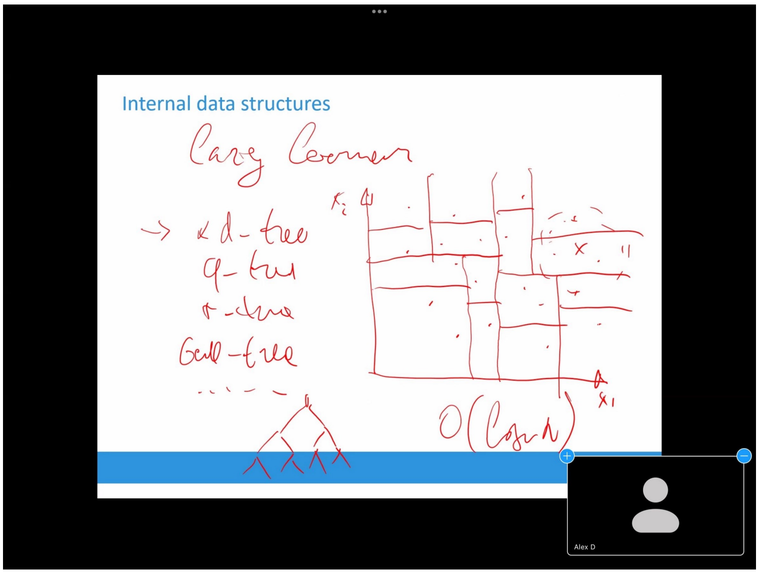
2.

FP Growth vs Apriori

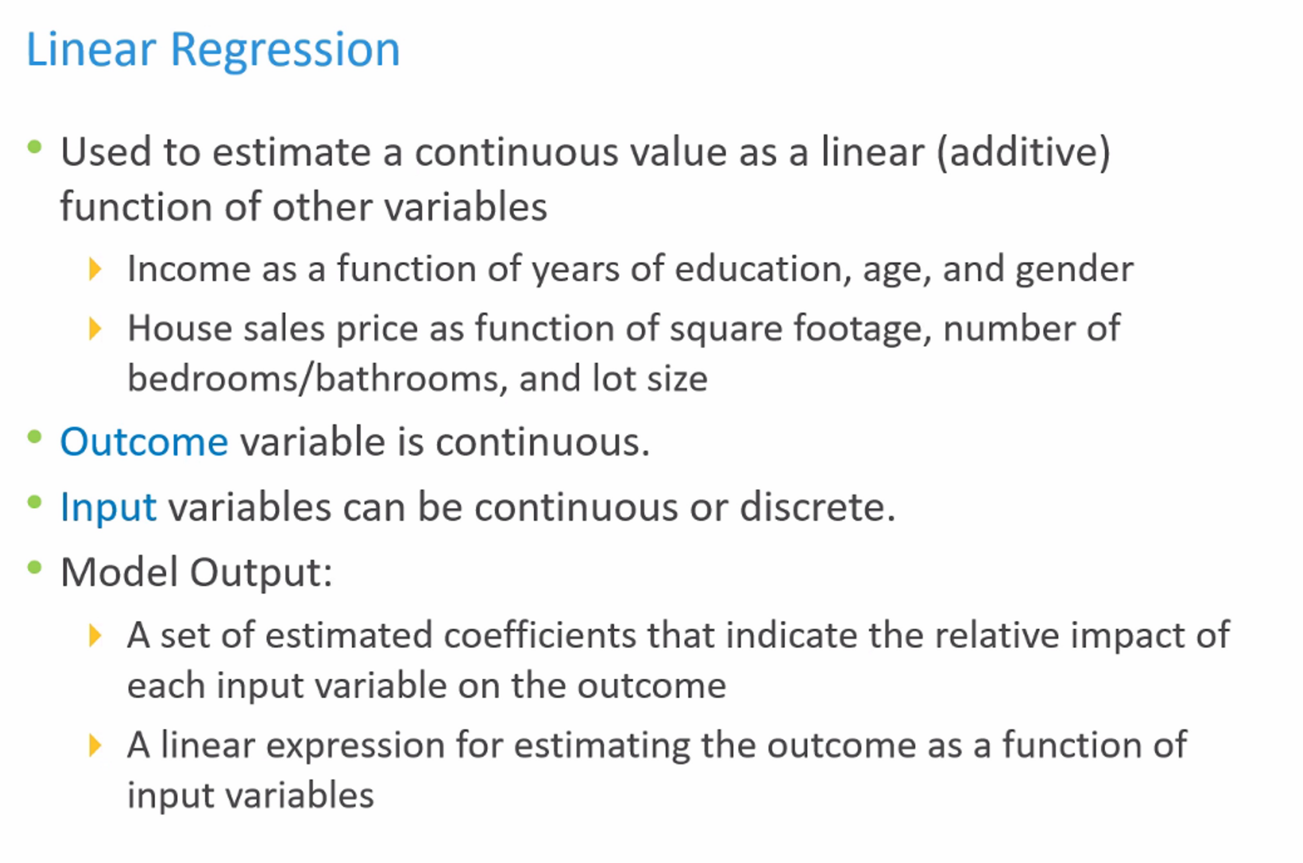
FP Growth	Apriori
Pattern Generation	
FP growth generates pattern by constructing a FP tree	Apriori generates pattern by pairing the items into singletons, pairs and triplets.
Candidate Generation	
There is no candidate generation	Apriori uses candidate generation
Process	
The process is faster as compared to Apriori. The runtime of process increases linearly with increase in number of itemsets.	The process is comparatively slower than FP Growth, the runtime increases exponentially with increase in number of itemsets
Memory Usage	
A compact version of database is saved	The candidates combinations are saved in memory

3. Classification. KNN - k neighbours
4. On a map of features looking for k neighbours and deciding what class does it belong to. If it's neighbour is cat, than it moves like cat and walks like cat than it is a 🐱
5. But some neighbours can be closer than others, so it is vital to consider assigning weight somehow to neighbours
6. 

7. But how it learns?
8. Very often it doesn't really learn. Lazy learner strategy just seek for k neighbours. But sometimes search trees are used for more efficient search of closest k neighbours

9. 

10. Linear regression
11. Prediction or classification is based on some linear equation.

12. 

13.

Linear Regression Model

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p + \varepsilon$$

where y is the outcome variable

x_j are the input variables, for $j = 1,2,\ldots,p$

β_0 is the value of y when each x_j equals zero

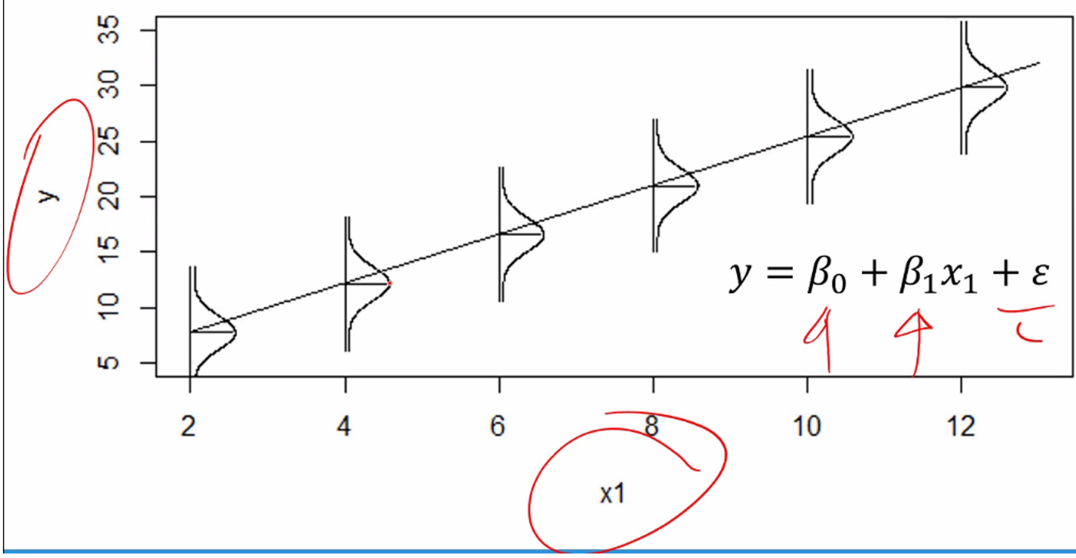
β_j is the change in y based on a unit change in x_j

$\varepsilon \sim N(0, \sigma^2)$ and the ε's are independent of each other

14.

Example: Linear Regression with One Input Variable

- x_1 - the number of employees reporting to a manager
- y - the hours per week spent in meetings by the manager



Classes are basically types of vector. For example, (8, 1, 0, 0) is a class of finance manager

1.

Representing Categorical Attributes

$$y = \beta_0 + \beta_1 employees + \beta_2 finance + \beta_3 mfg + \beta_4 sales + \varepsilon$$

Possible Situation	Input Variables
Finance manager with 8 employees	(8,1,0,0)
Manufacturing manager with 8 employees	(8,0,1,0)
Sales manager with 8 employees	(8,0,0,1)
Engineering manager with 8 employees	(8,0,0,0)

- For a categorical attribute with *m* possible values
  - Add *m-1* binary (0/1) variables to the regression model
  - The remaining category is represented by setting the *m-1* binary variables equal to zero

2.

Линейная регрессия как задача оптимизации

$$\min_Q 1/(2m)\sum_i^m (h_Q(x^{(i)}) - y^{(i)})^2$$

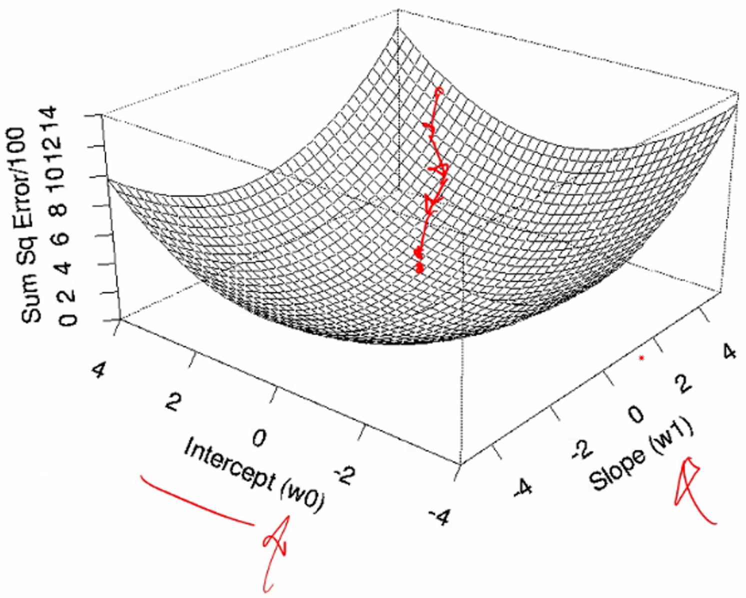
- квадратичная функция ошибки

$J(Q) = 1/(2m) \sum_i^m (h_Q(x^{(i)}) - y^{(i)})^2$

minimize $J(Q)$

1. Moving towards anti-gradient, since gradient shows the vector of maximum increase

Иллюстрация J(Q)

2. 

3.

Алгоритм градиентного спуска (2)

while (|oldJ(Q) – J(Q)| > ε)

{

oldJ(Q) = J(Q)

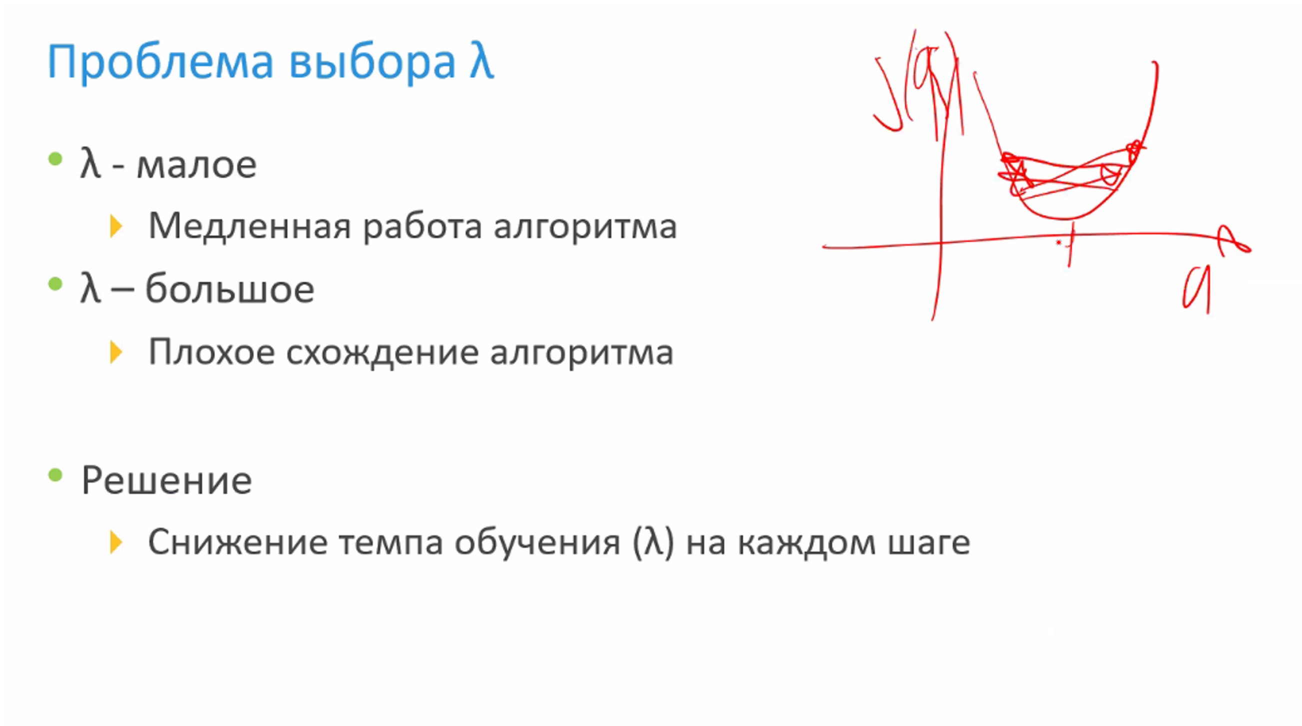
$Q_0 = Q_0 - \lambda / m \sum_i^m (h_Q(x^{(i)}) - y^{(i)}))$

$Q_k = Q_k - \lambda / m \sum_i^m (h_Q(x^{(i)}) - y^{(i)})) \, x^{(i)}_k$

foreach k=1…n

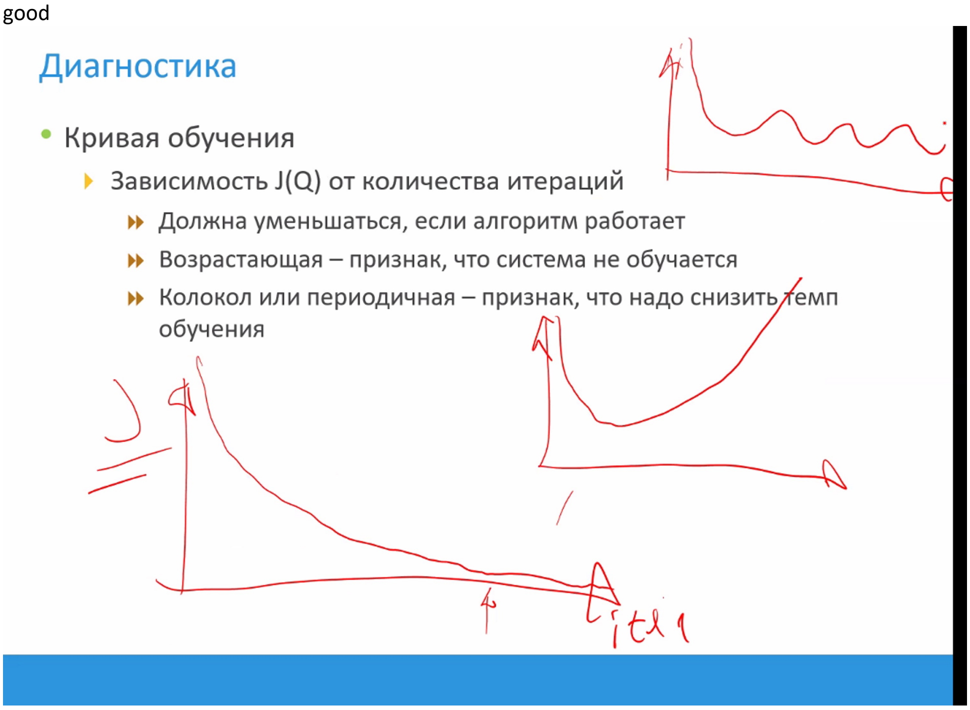
}

4.



5. First picture - wrong lambda coefficient chosen, overthrowing. Second - wrong algo, third - good

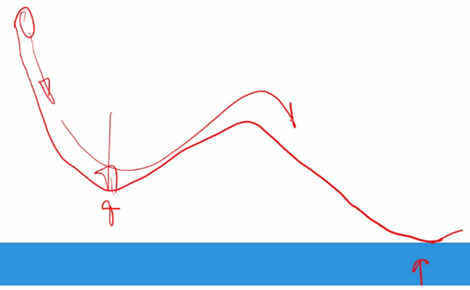
6.



7.

Проблема локальных минимумов

- Решение
  - Использование более сложных алгоритмов ☺
- Пример
  - Стохастический градиентный спуск
  - Метод имитации отжига (simulated annealing)
  - …



Yey physics

8.

Имитация отжига

- Идея:
  - При высокой температуре более вероятны случайные переходы в другие состояния (можно «выпрыгнуть» из потенциальной ямы / локального минимума)
  - Температура постепенно снижается до 0
- Различные варианты
  - λ(t) + градиентный спуск
  - Случайный переход Δ(t)
    - Со сравнением / без сравнения со старыми значениями
  - Градиентный спуск + случайный переход с вероятностью p(t)

9. The more the temperature the highly likely are transitions to other states
10. Also there is a problem of different scale. When variables are of very different scale. Than we can observe combe(ovrag in russian)
11. The solution is normalization or rescaling

12.

Проблема переобучения

- Общая для машинного обучения
- Суть:
  - Можно обучить систему так, что она будет идеально работать на тренировочных данных, но не будет полезной моделью, т.к. не будет обладать классифицирующей или предсказательной способностью на данных, которые модель еще «не видела»
- Пример:
  - Чем выше степень полинома, тем более точно кривая пройдет, через базовые точки, но и тем более экстремально она будет себя вести

13. How to prevent?
14. Cut the number of input variables to only those that have the biggest impact. Others will only idealize the curve

15.

Регуляризация

- Идея:
  - Штрафовать модель за большие значения коэффициентов
- Два часто используемых подхода
  - Ridge (L2)
  - Lasso (L1)

16. Logistic regression