

Санкт-Петербургский государственный университет  
телекоммуникаций им. проф. М.А. Бонч-Бруевича

# Искусственный интеллект в сетях связи

Другие виды обучаемых систем

Выборнова А.И., доц. каф. ССиПД

СПб ГУТ)))

# Что такое машинное обучение?

**Машинное обучение** – это обеспечение для какой-либо системы возможности решать задачи, обучаться и изменяться без непосредственного программирования системы, а на основе опыта.

Машинное обучение служит для обучения нейронных сетей, но может применяться и к другим системам.

# Классификация

Классификация методов машинного обучения:

- Классификация/кластеризация:
  - Два варианта ответа (да или нет).
  - Больше двух вариантов ответа (человек, дерево, здание).
- Регрессия (ответ – любое число).

# Классификация

Классификация методов машинного обучения:

- Обучение с учителем:
  - Классификация.
  - Регрессия.
- Обучение без учителя:
  - Кластеризация.
- Обучение с частично известными ответами/метками («частично с учителем»).
- Обучение с подкреплением.

# Нормализация данных

**Нормализация** – это обработка данных с целью приведения разнородных данных к одному масштабу и/или одной форме распределения.

Min-max нормализация:

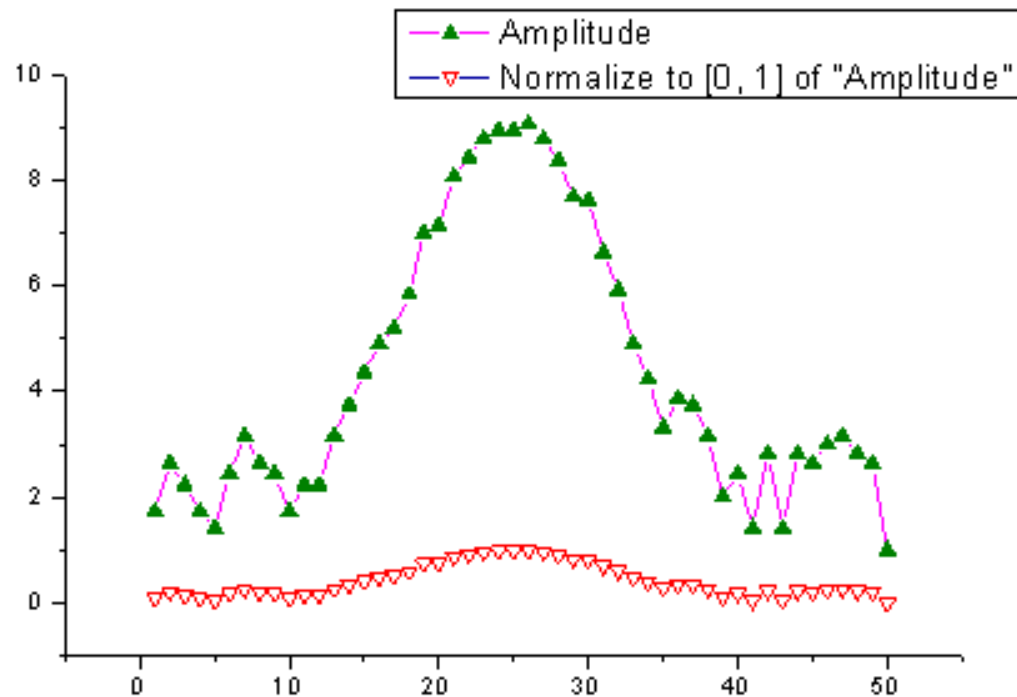
$$X_i = \frac{X_i - X_{min}}{X_{max} - X_{min}}$$

Z-scale нормализация:

$$X_i = \frac{X_i - \mu}{\sigma}$$

$\mu$  - среднее,

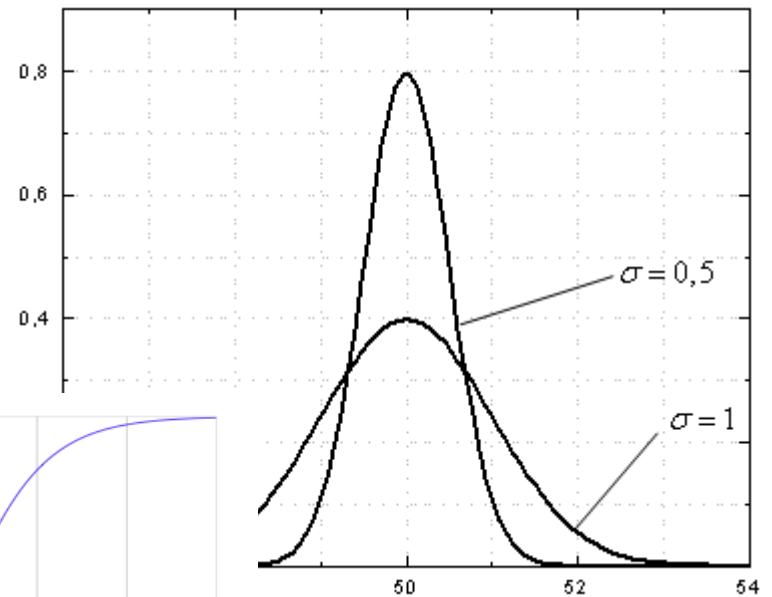
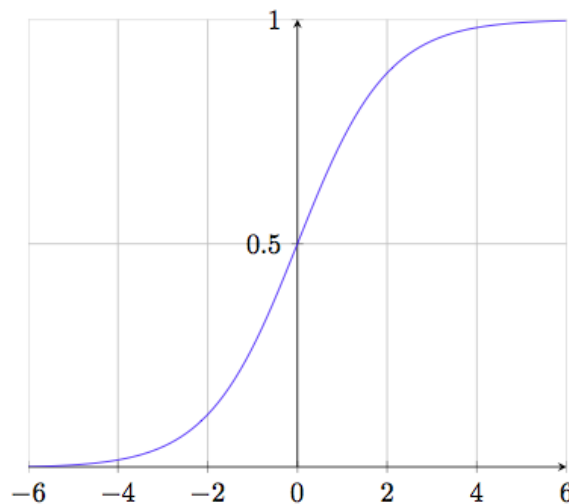
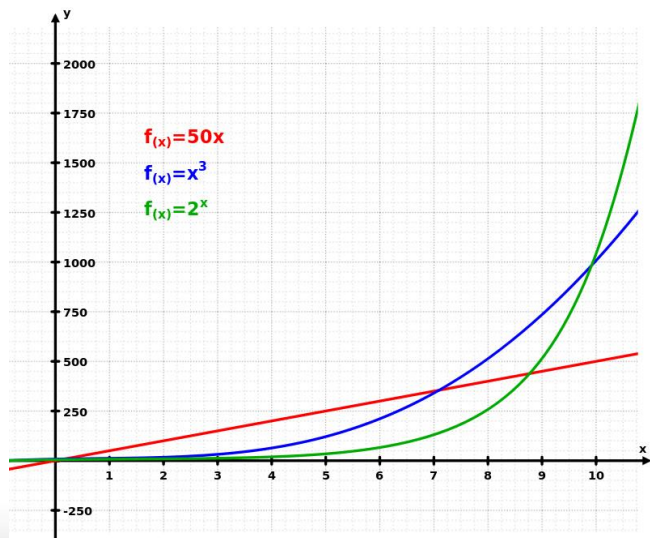
$\sigma$  - ср. кв. отклонение



# Регрессионный анализ – тоже ML

В качестве функции регрессии для данных, имеющих нелинейные зависимости, могут использоваться различные типы функций:

- Экспоненциальные.
- Логарифмические.
- Логистические.
- Гауссиана и др.



# Метод опорных векторов

Метод опорных векторов (Support Vector Machine) – это алгоритм классификации для многомерных данных с метками (известными ответами) при помощи гиперплоскости.

Размерность данных – это число параметров входных данных:

GNIperCapita	UrbanPopulation	FertilityRate	Tuberculosis	Undernourishment
680	26.282	4.843	340	26
5490	70.129	2.857	118	5
13480	91.604	2.322	30	5
4020	62.812	1.531	55	6.3
7600	54.355	2	102	5
1080	33.516	2.175	404	16.9
890	43.514	4.766	89	8.1
2870	68.107	2.968	187	16.6
7240	57.187	2.836	354	24.8



# Метод опорных векторов

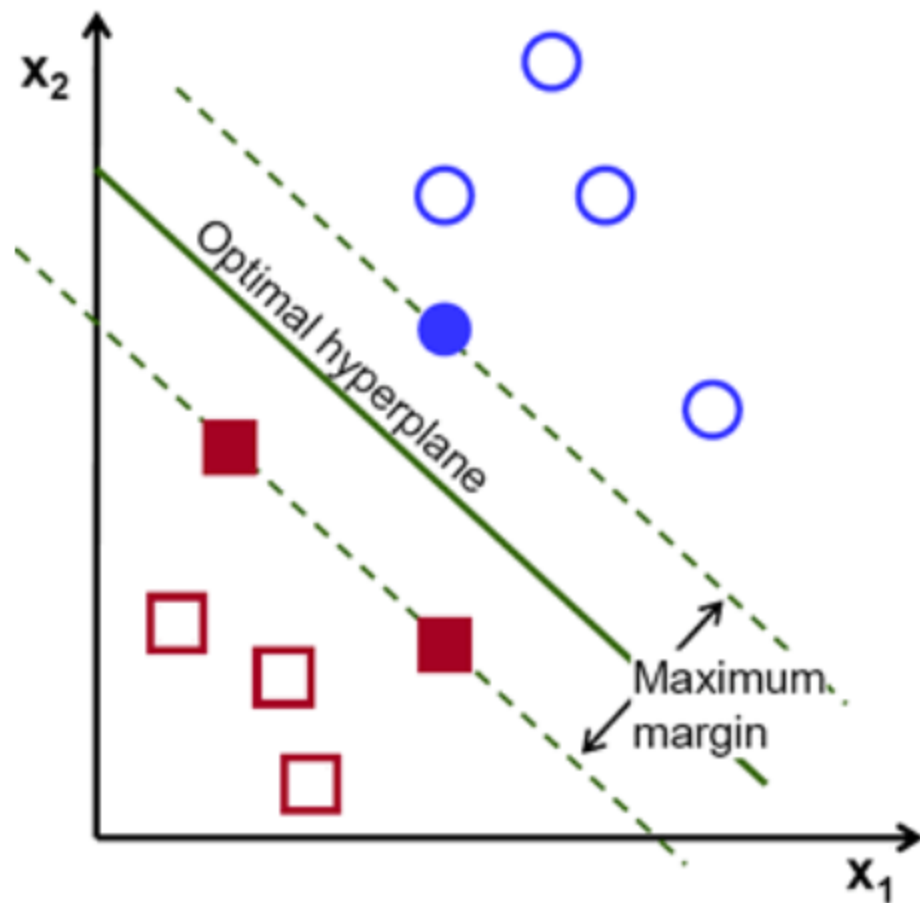
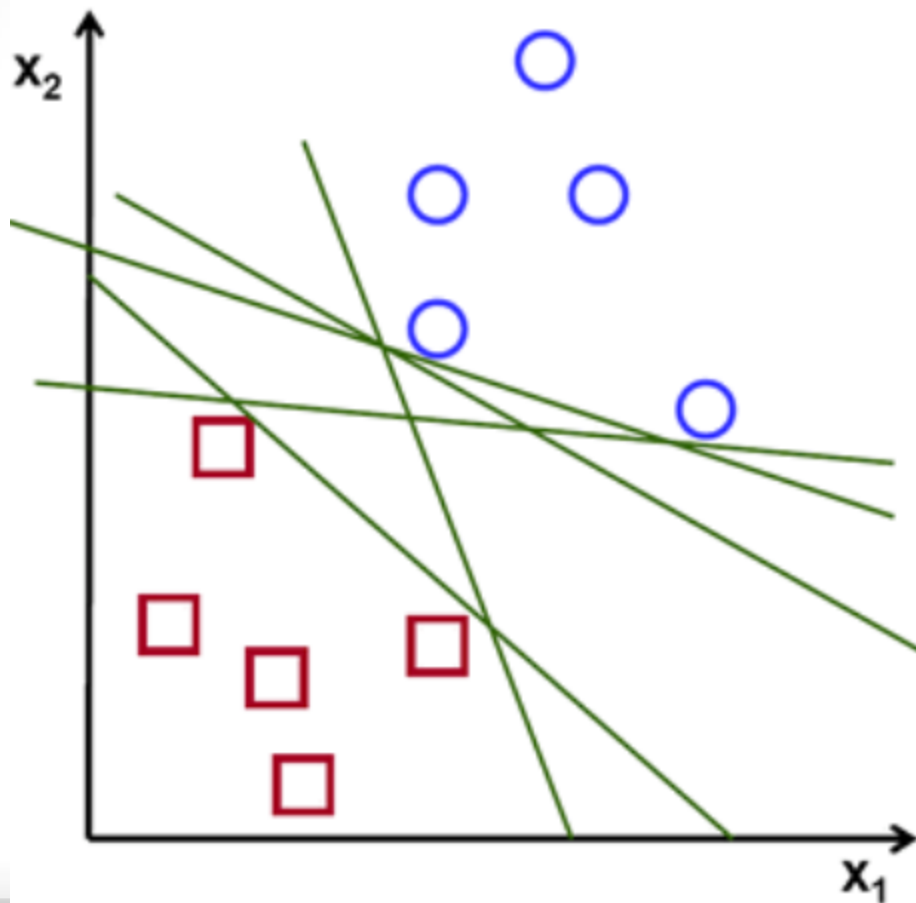
Гиперплоскость – подпространство с размерностью на единицу меньше, чем пространство, в котором находится гиперплоскость (для плоскости – прямая, для трехмерного пространства – плоскость и т.д.).

Метод опорных векторов оперирует только одной гиперплоскостью, поэтому может разделить пространство только на две части, т.е. пригоден в изначальном виде только для классификаций не два класса (two-class).



# Метод опорных векторов

Случай линейной делимости данных:



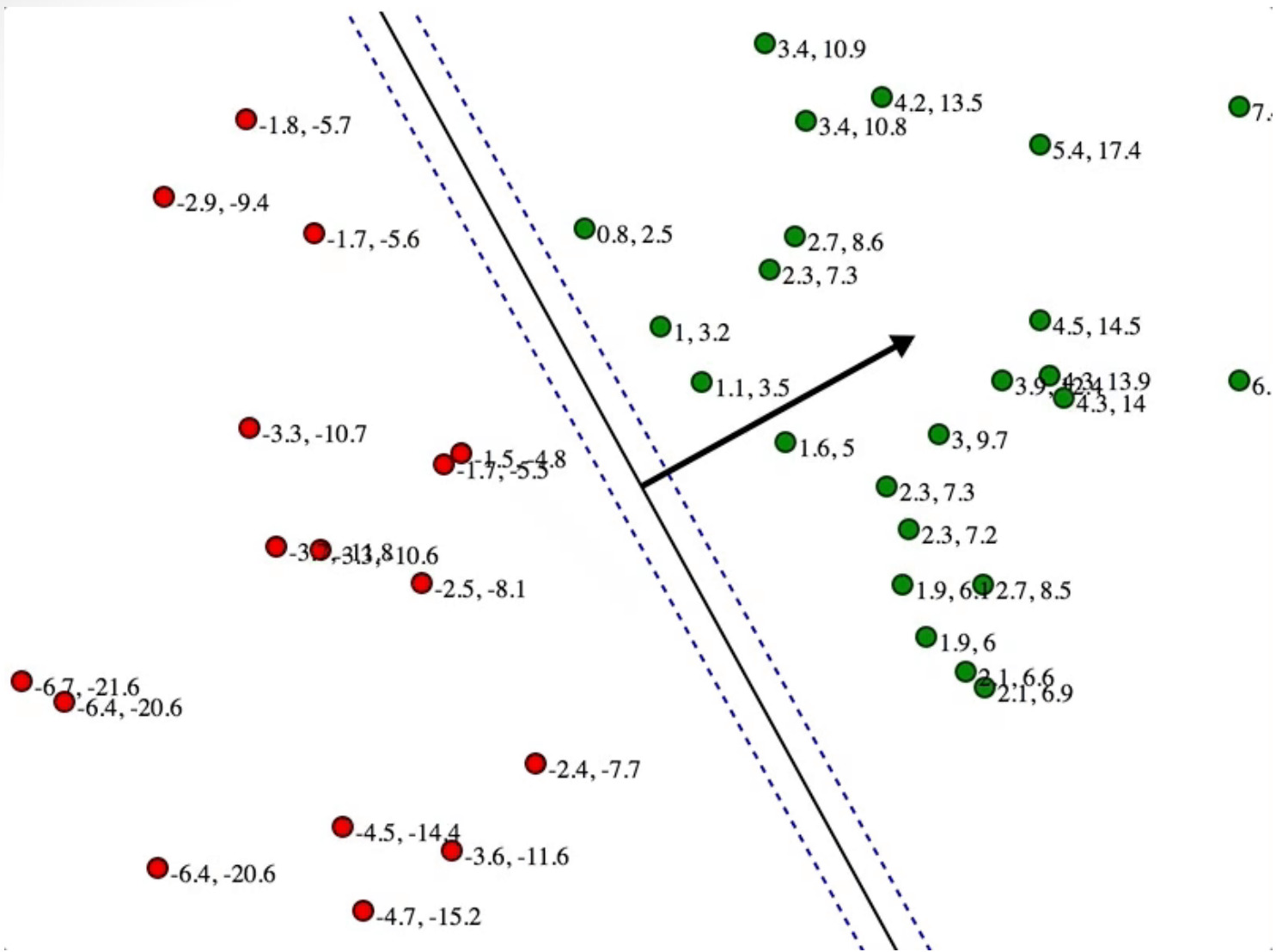
# Метод опорных векторов

В случае линейной разделимости данных для каждой гиперплоскости, разделяющей два класса, строятся опорные векторы – гиперплоскости, параллельные изучаемой, и проходящие через ближайшие к ней точки данных.

Далее решается задача оптимизации, а именно – максимизации расстояния между опорными векторами (зазора, margin).

Гиперплоскость, для которой зазор между опорными векторами максимален, и будет оптимальной гиперплоскостью, разделяющей данные на два класса.

# Метод опорных векторов



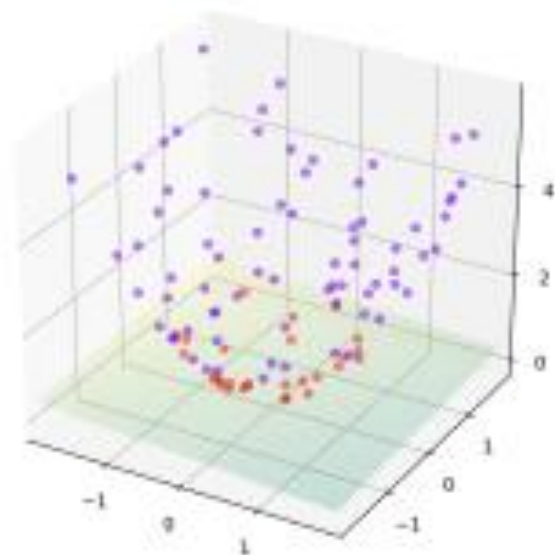
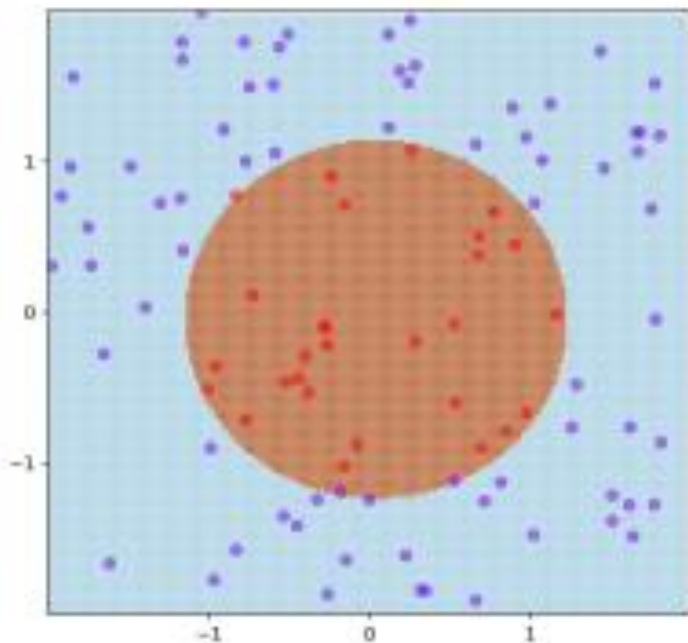
# Метод опорных векторов

Более жизненный случай – два класса не могут быть разделены гиперплоскостью (например, прямой для двумерного пространства), ТО есть не делимы линейно.

Решение – перевести данные в пространство большей размерности (пространство признаков, feature map) при помощи **метода ядра**, и там попытаться разделить при помощи гиперплоскости описанным ранее методом опорных векторов.

# Метод опорных векторов

Методы ядра (ядерный метод, kernel trick)



Ядро в данном случае задается функцией  
 $\varphi((a, b)) = (a, b, a^2 + b^2)$

# Метод опорных векторов

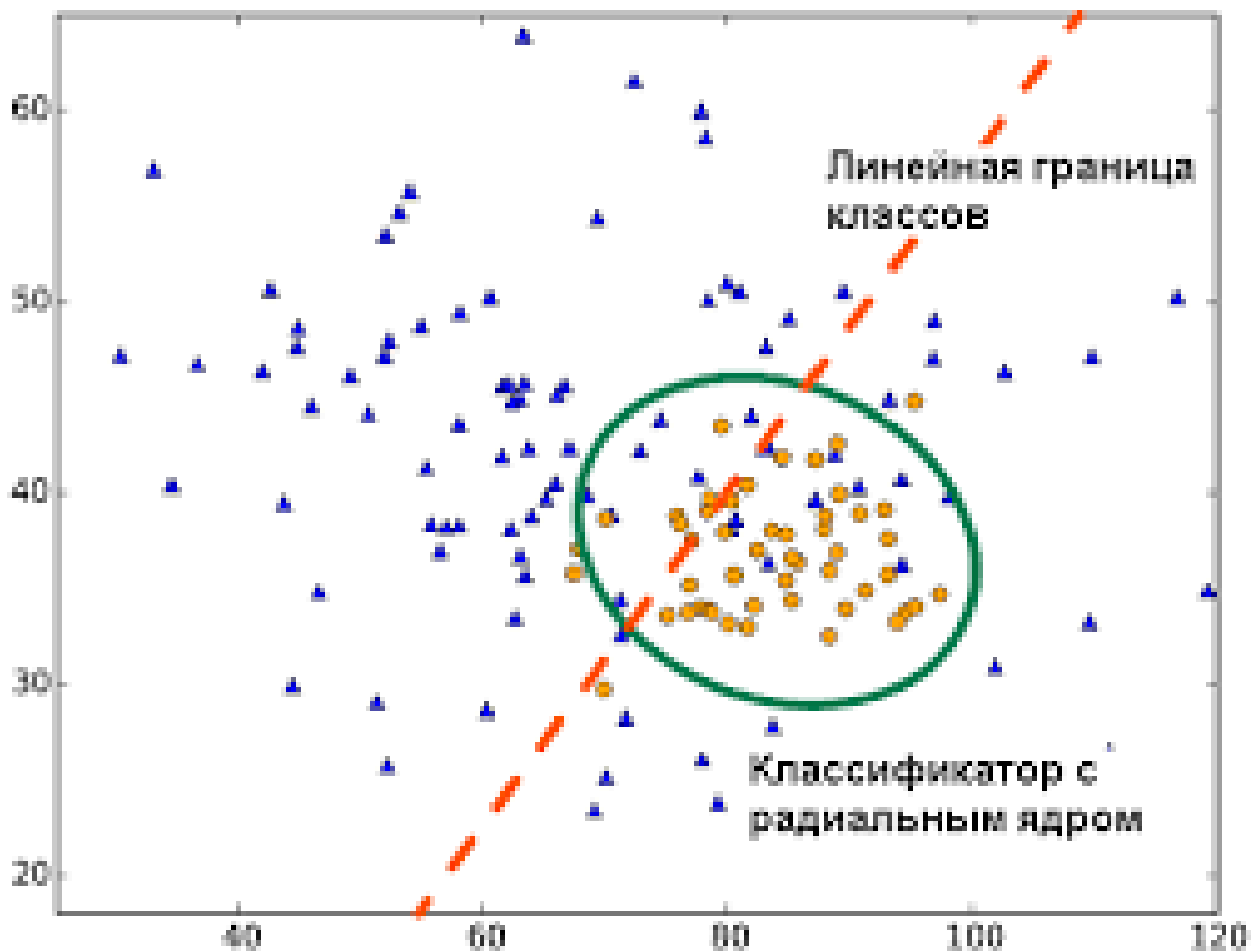
Трюк – потому что вместо вычисления координат всех точек в новом пространстве (трудоемко), вычисляются только скалярные расстояния между ними.

Ядро  $\phi((a, b)) = (a, b, a^2 + b^2)$     Расстояние  $K(x, y) = x \cdot y + x^2 y^2$

Существуют различные функции ядра, применяемые для разных целей.

# Метод опорных векторов

Другое представление применения ядра:





# Метод опорных векторов – плюсы и минусы

Преимущества:

- Часто самый быстрый.
- Максимальная ширина полосы -> более точная классификация после обучения.

Недостатки:

- Требуется нормализация и чувствителен к шумам данных.
- Нет единого подхода к выбору ядра в случае нелинейной разделимости.

# Деревья решений

Дерево решений (decision tree) – метод классификации.

Определение, выжил ли человек на Титанике:



# Деревья решений

Листья дерева берутся из «меток», то есть заданных типов правильных ответов.

Узлы дерева – критериальные переменные (входные данные), которые сравниваются с некоторыми пороговыми значениями, определяемыми в процессе обучения.

# Деревья решений

Каким образом происходит построение дерева на имеющихся данных для обучения?

Существует несколько алгоритмов построения деревьев.

В общем случае построение начинается с корня дерева. Выбирается параметр  $A$ , например, с наименьшей энтропией (метод Шеннона) и становится следующим узлом дерева. От этого узла создается  $n$  потомков по числу вариантов параметра  $A$ . Каждому потомку ставится в соответствие множество входных данных с соответствующим значением параметра  $A$ .

# Деревья решений

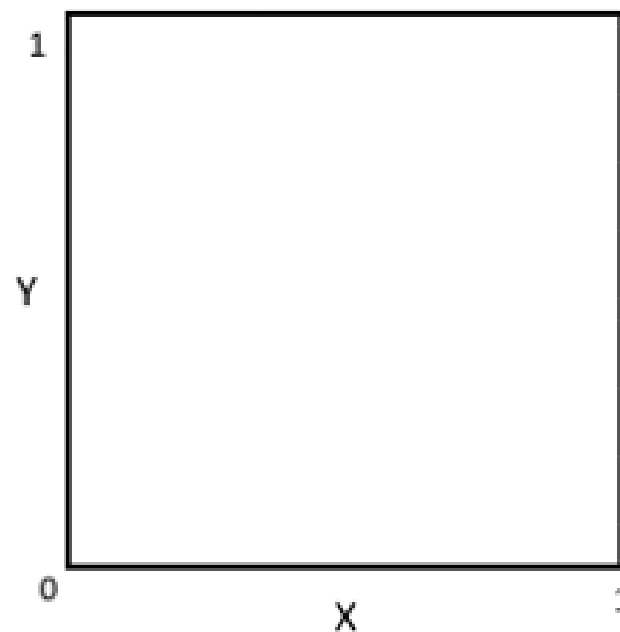
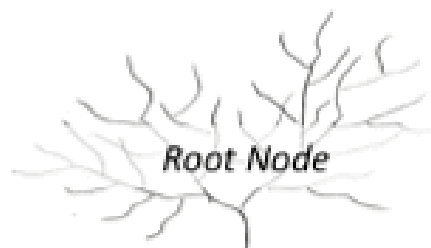
Процесс повторяется рекурсивно, пока множество параметров каждой ветви:

- не станет пустым (тогда эта ветвь становится листом, в качестве ответа выбирается наиболее частая метка в множестве входных данных непосредственного предка этой вершины),

ЛИБО

- все метки подмножества входных данных этой ветки не будут иметь одну и ту же метку (тогда эта точка становится листом с соответствующей меткой).

# Деревья решений



For more tutorials: [annalyzin.wordpress.com](http://annalyzin.wordpress.com)

# Деревья решений - усовершенствования

Существует ряд усовершенствований классического алгоритма:

- **CART** – classification and regression tree – возможность не только классификации, но и регрессии.
- **Обрезка деревьев (pruning)** – удаление веток с низким вкладом в результат, для избегания эффекта переобучения и уменьшения объема вычислений.
- **Бэггинг (bagging)** – из изначального набора данных выбирается несколько поднаборов (причем данные из начального набора в поднаборах могут встречаться несколько раз или не встречаться вообще). На каждом поднаборе обучается свое дерево, далее выбирается наиболее удачное.



# Деревья решений - усовершенствования

Существует ряд усовершенствований классического алгоритма:

- **Случайный лес** (random forest) – частный случай бэггинга, где все деревья оставляются, а для принятия окончательного решения при использовании леса выбирается то, что чаще всего встретилось среди деревьев леса.
- В некоторых случаях для обучения разных деревьев леса используются не только разные поднаборы данных, но и разные комбинации параметров в этих данных.

# Деревья решений - усовершенствования

A person with long black hair, wearing a black leather vest and pants, stands in a lush green forest. They are pointing their right hand towards a tree. The forest floor is covered in moss and ferns, and the trees are tall and thin.

**WELCOME TO THE FOREST**

**HERE'S A RANDOM TREE**

# Деревья решений - усовершенствования

Существует ряд усовершенствований классического алгоритма:

- Использование вместо дерева **ориентированного ациклического графа (DAG, directed acyclic graph)**.

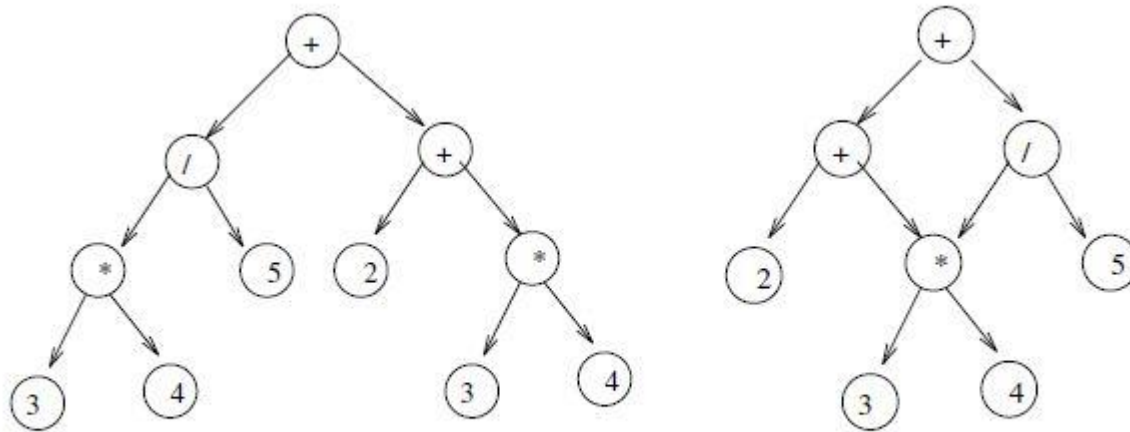


Figure 5.17: Expression  $2 + 3 * 4 + (3 * 4) / 5$  as a tree and a DAG



# Деревья решений - усовершенствования

Существует ряд усовершенствований классического алгоритма:

- Лес из ориентированных ациклических графов вместо деревьев называется Random Jungle



# Деревья решений – плюсы, минусы, подводные камни

Преимущества:

- Создает «белый ящик» вместо черного, как в других алгоритмах, то есть очевидны критерии принятия решения, и их можно попробовать интерпретировать.
- Не требует нормализации, в целом менее требователен к подготовке данных.

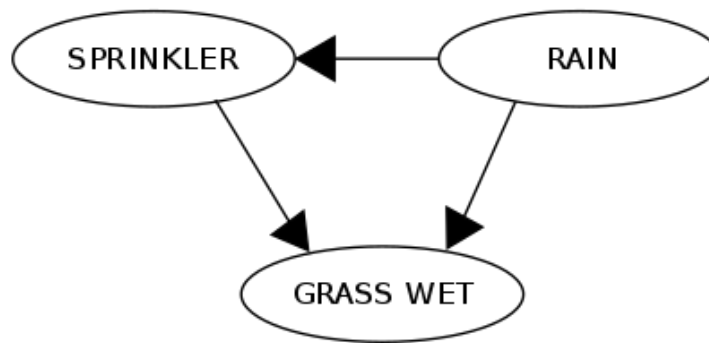
■ Недостатки:

- Без отсечения ветвей легко переобучается.
- Небольшие изменения в данных могут существенно менять вид дерева, поэтому плохо подходит для работы с изменяющимися данными.
- Могут быть сложности с поиском глобально-оптимального решения.

# Байесовская сеть

Байесовская сеть (Bayesian network) – граф, в вершинах которого находятся входные параметры, а ребра показывают вероятностную зависимость между этими параметрами.

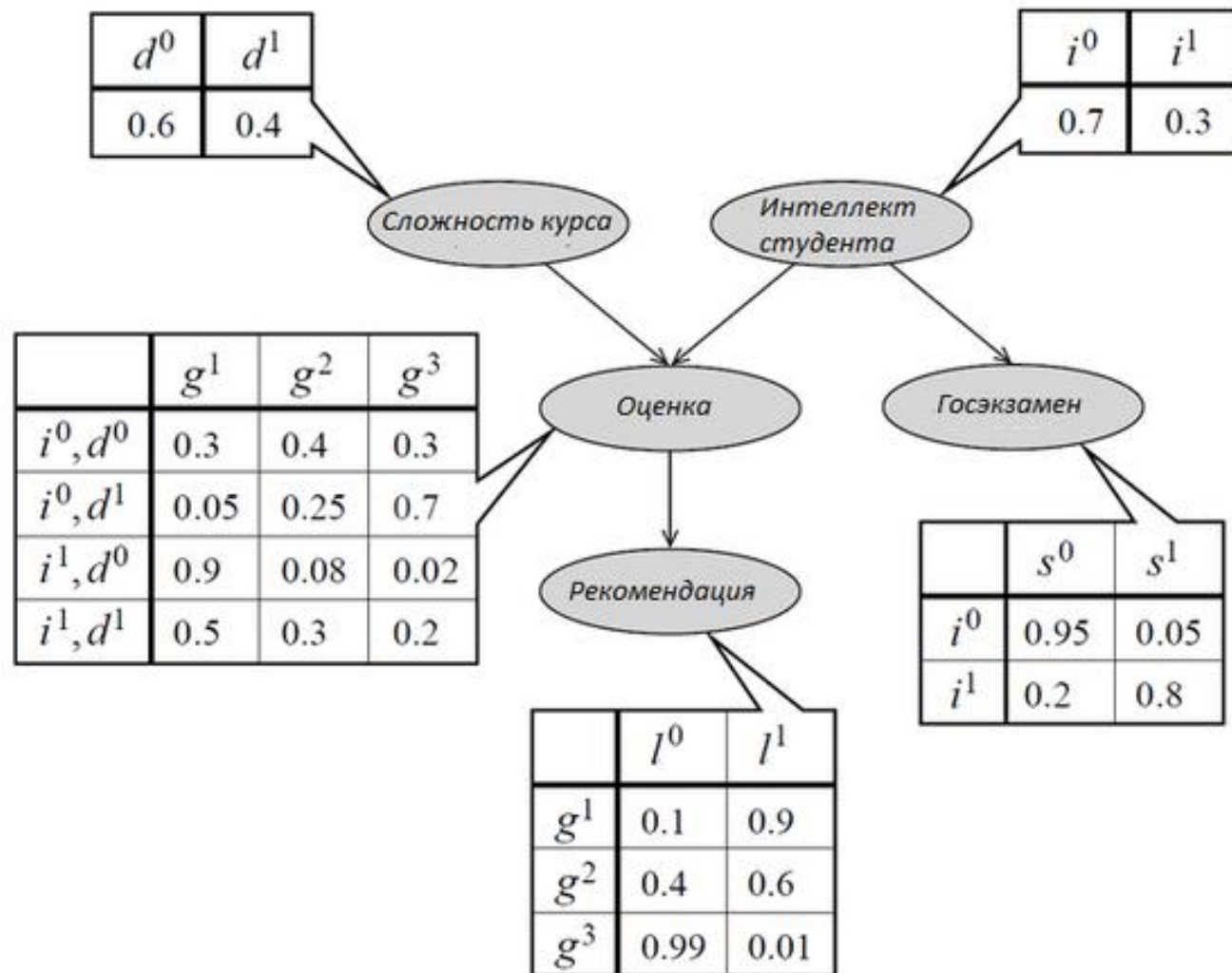
		SPRINKLER	
		T	F
RAIN	F	0.4	0.6
	T	0.01	0.99



		RAIN	
		T	F
		0.2	0.8

		GRASS WET	
		T	F
SPRINKLER	F	0.0	1.0
	T	0.8	0.2
RAIN	F	0.9	0.1
	T	0.99	0.01

# Байесовская сеть





# Байесовская сеть

Формула Байеса:

$$P(A | B) = \frac{P(B | A) P(A)}{P(B)},$$

где

$P(A)$  — априорная вероятность гипотезы  $A$  (смысл такой терминологии см. ниже);

$P(A | B)$  — вероятность гипотезы  $A$  при наступлении события  $B$  (апостериорная вероятность);

$P(B | A)$  — вероятность наступления события  $B$  при истинности гипотезы  $A$ ;

$P(B)$  — полная вероятность наступления события  $B$ .

# Байесовская сеть

Обучение Байесовской сети в простейшем случае возможно чисто статистически – по частоте встречаемости.

В более сложных случаях берется множество вариантов таблиц сети (гипотез) и любым методом оптимизации выбирается гипотеза с наибольшей вероятностью, что данные соответствуют той гипотезе.