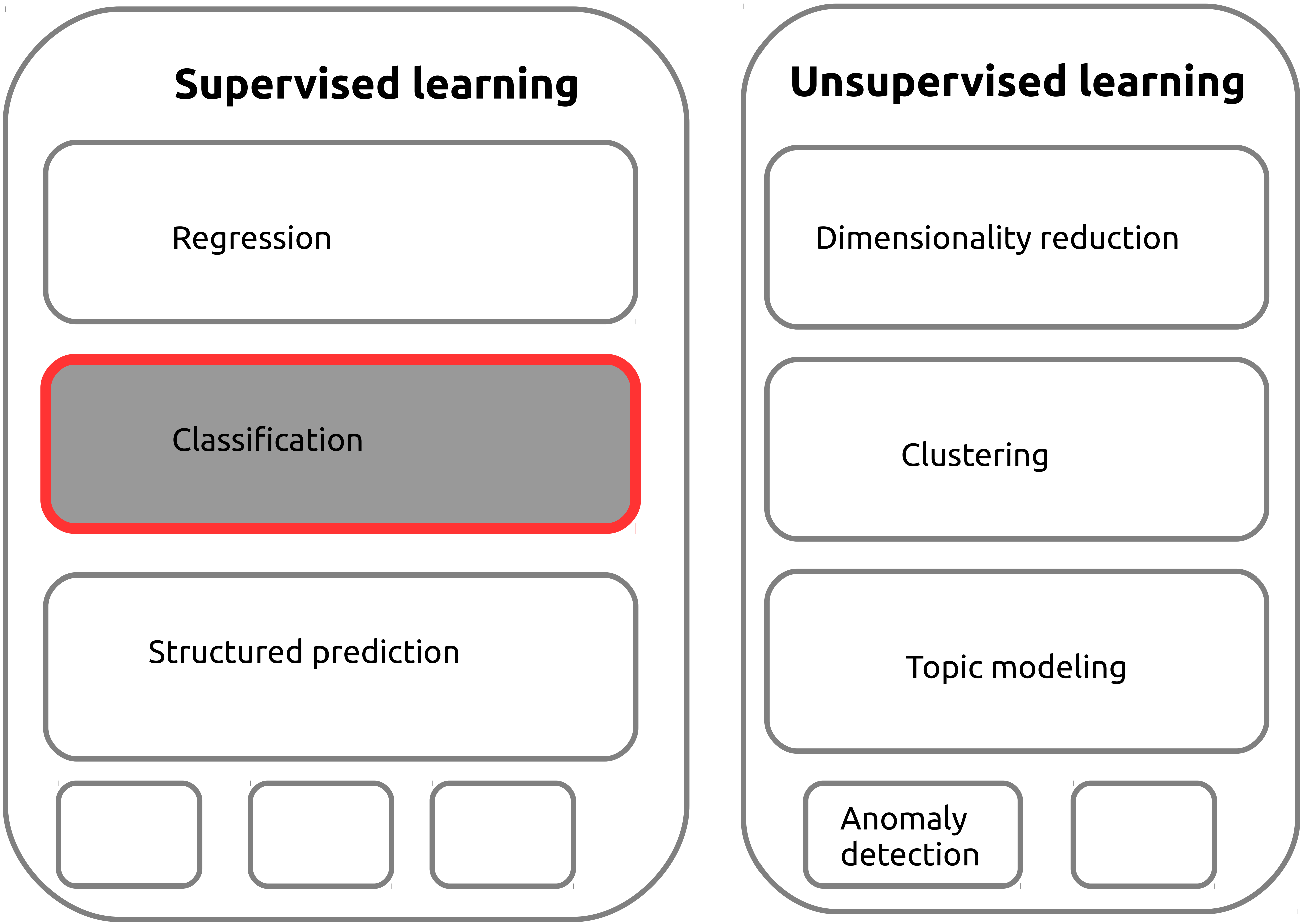
Naive Bayes

Grzegorz Chrupała
@gchrupala


Supervised learning
Regression
Classification
Structured prediction
Unsupervised learning
Dimensionality reduction
Clustering
Topic modeling
Anomaly detection

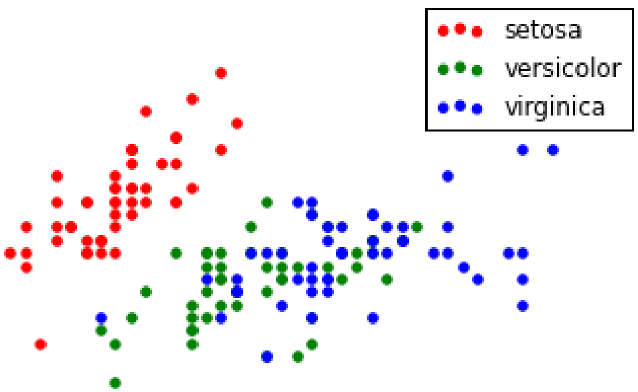
Reminder: How does KNN work?

- Remember training examples
- When new example arrives
  - Find nearby points
  - Return their majority class

Naive Bayes classifier

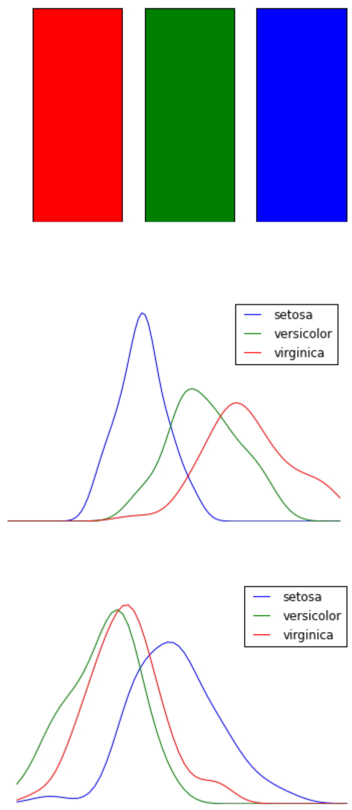
- Estimates probabilities of classes and features from training data
- When new example arrives
  - Computes and the most probable class for its features
  - (while making a naive assumption)

K-NN



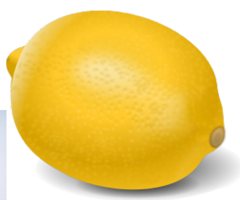
setosa
versicolor
virginica

NB



setosa
versicolor
virginica
setosa
versicolor
virginica

Toy example



Shape	**Color**	***Target***
Round	Green	Lime
Round	Yellow	Lemon
Round	Green	Apple
Round	Red	Apple
Long	Yellow	Banana
Long	Green	Banana

New example

Shape	Color
Round	Green

What are the probabilities of the classes, given this information about the fruit?

P(Banana | Shape=Round, Color=Green) = ?

P(Apple | Shape=Round, Color=Green) = ?

P(Lemon | Shape=Round, Color=Green) = ?

P(Lime | Shape=Round, Color=Green) = ?

If we knew, we could chose the most probable

Estimate probabilities of classes

- Our estimates → relative counts

  P(Lime) = 1/6
  P(Lemon) = 1/6
  P(Apple) = 1/3
  P(Banana) = 1/3

Can use counts from the data directly?

- In this toy example, yes
- But what if we had many more features:
  - P(Lime | Shape=Round, Color=Green,
    Weight=200g, Diameter=7cm,
    Taste=Sour, Texture=Smooth,
    ....)
  - Not many other examples exactly matching these features

Naive Bayes trick

- The NB classifier uses two ideas
  - Invert probabilities using Bayes Law
  - Assume features are independent

Properties of probabilities

- Conditional probability

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

- Chain rule

$$P(A,B) = P(A|B)P(B)$$

Bayes Law

Combines definition of conditional probability with chain rule to invert direction of conditioning

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

Concretely

$$P(\text{Lime}|x_1, x_2) = \frac{P(x_1, x_2|\text{Lime})P(\text{Lime})}{P(x_1, x_2)}$$

Did it help?

- We still need to estimate probabilities of complex things like

$$P(x_1, x_2 | \mathrm{Lime})$$

Let's make a naive assumption

- Two events A and B are **conditionally independent** if

$$P(A, B|C) = P(A|C)P(B|C)$$

- Let's assume features are independent given the class

$$P(x_1, x_2|\mathrm{Lime}) = P(x_1|\mathrm{Lime})P(x_2|\mathrm{Lime})$$

Putting it together

$$P(\text{Lime}|x_1, x_2) = \frac{P(x_1|\text{Lime})P(x_2|\text{Lime})P(\text{Lime})}{P(x_1)P(x_2)}$$

Simplification

- We can ignore the denominator. Why?

$$P(\mathrm{Lime}|x_1, x_2) = \frac{P(x_1|\mathrm{Lime})P(x_2|\mathrm{Lime})P(\mathrm{Lime})}{P(x_1)P(x_2)}$$

$$P(\mathrm{Lemon}|x_1, x_2) = \frac{P(x_1|\mathrm{Lemon})P(x_2|\mathrm{Lemon})P(\mathrm{Lemon})}{P(x_1)P(x_2)}$$

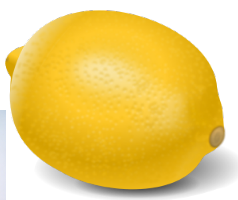
The denominator is the same in all both cases so it does not affect which class is more probable

Naive Bayes

$$y_{\text{pred}} = \arg\max_y P(y) \prod_{j=1}^{J} P(x_j|y)$$

where y is a class and x_j a feature

Toy example



Shape	**Color**	***Target***
Round	Green	Lime
Round	Yellow	Lemon
Round	Green	Apple
Round	Red	Apple
Long	Yellow	Banana
Long	Green	Banana

What are class probabilities for

Shape	Color
Round	Green

- P(Lime)P(Round,Green|Lime) =

  P(Lime)P(Round|Lime)P(Green|Lime)

  - = 1/6 × 1 × 1 = 1/6
- P(Lemon)P(Round,Green|Lemon) = P(Lemon)P(Round|Lemon)P(Green|Lemon)
  - = 1/6 × 1 × 0 = 0

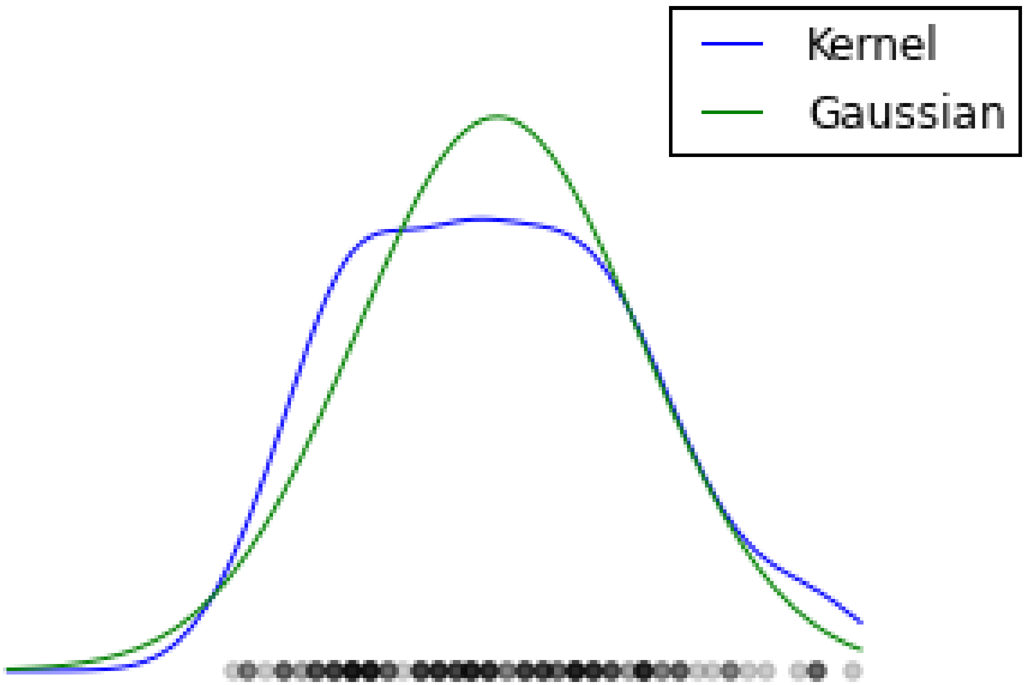
Smoothing

- Using estimates of exact zero
  - too strict
- Smoothing
  - distributing probabilities more evenly
- Simple method
  - add 1 to all counts

- What are the class probabilities for the toy example using add-1 smoothing?

What about continuous features?

- Estimate feature densities
- Parametric
  - Gaussian
- Non-parametric
  - Kernel density



Kernel
Gaussian

Summary

- Naive Bayes
  - Probabilistic classification
- Bayes Law
- Independence assumption
- Estimating probabilities
  - smoothing