Small Data

CMPT 733

Spring 2017

Apala Guha

Why Small Data?

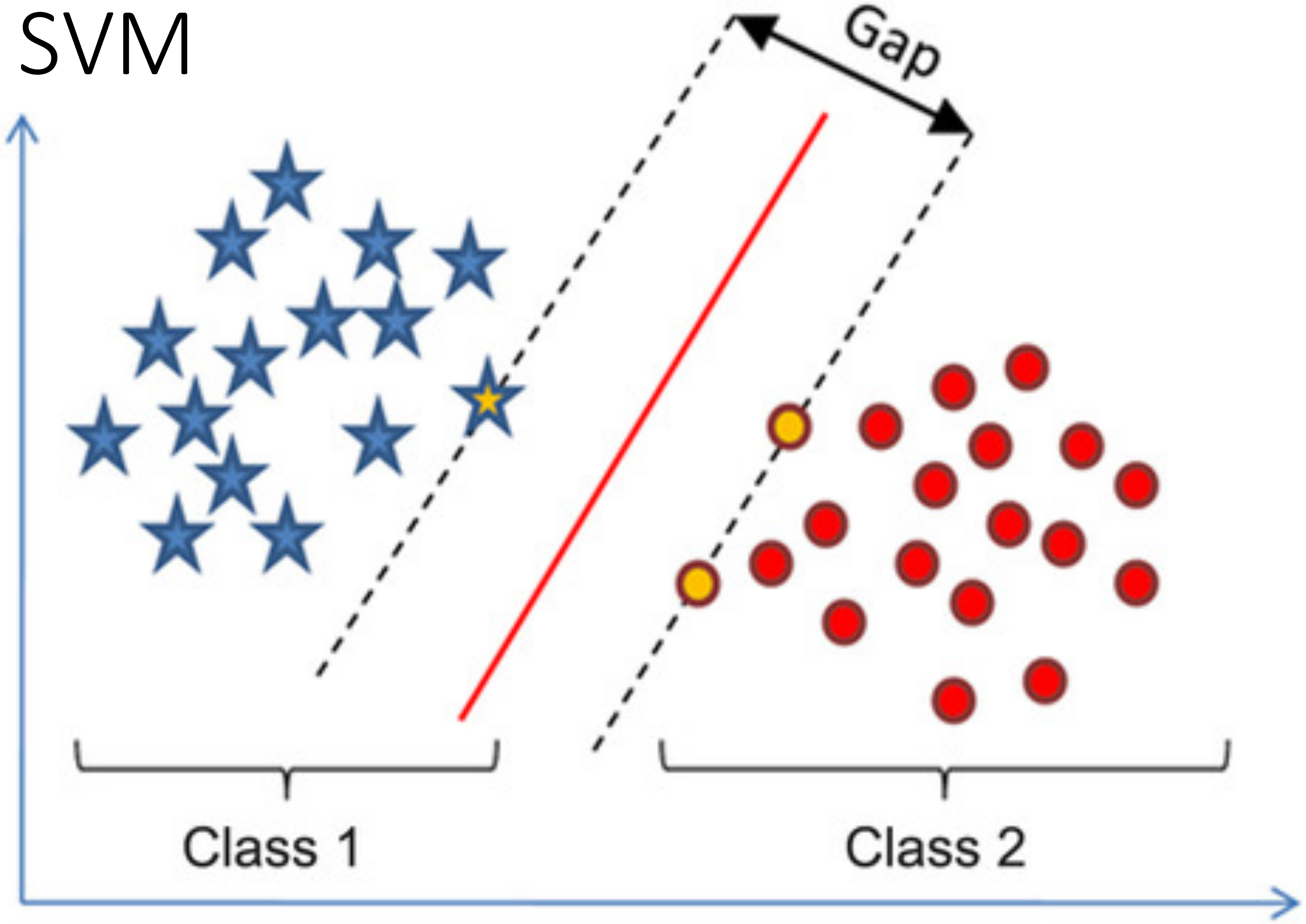
- Often our dataset fits on a single node, and we can avoid the communication and synchronization costs of big data

- Typical numbers:
  - 1 compute: 1 ns
    (West Mall to Cornerstone)
  - DRAM access: 100 ns
    (Squamish to Cornerstone)
  - Disk access: 150,000 ns
    (Halfway to moon)
  - Datacenter round trip: 500,000 ns
    (?)

- classification
  - Nearest Neighbor
  - Logistic Regression
  - Support Vector Machines
  - Decision Trees
  - Neural Networks

Support Vector Machines

Linear SVM

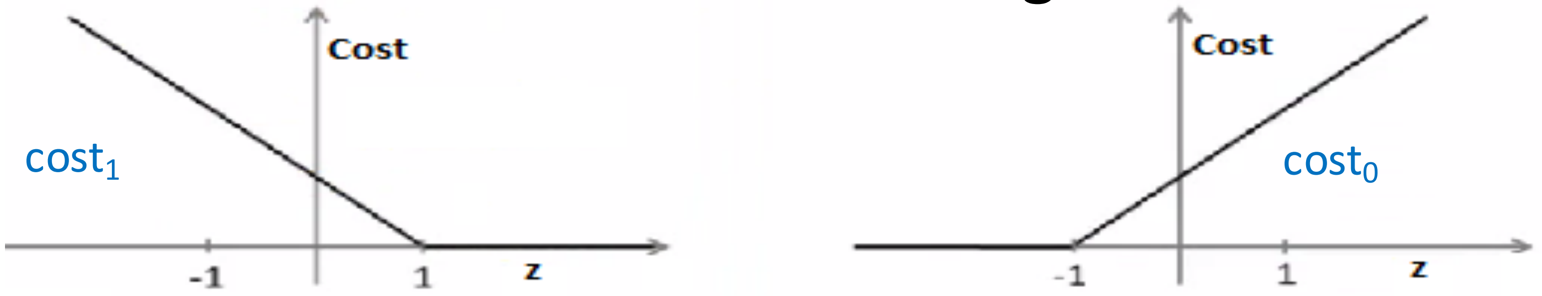


Linear SVM

- training example: (X, y)
- weight vector: W
- prediction: W^TX
- cost: y * $cost_1(W^TX)$
  + (1 - y) * $cost_0(W^TX)$
- regularization: C (inverse of ƛ)

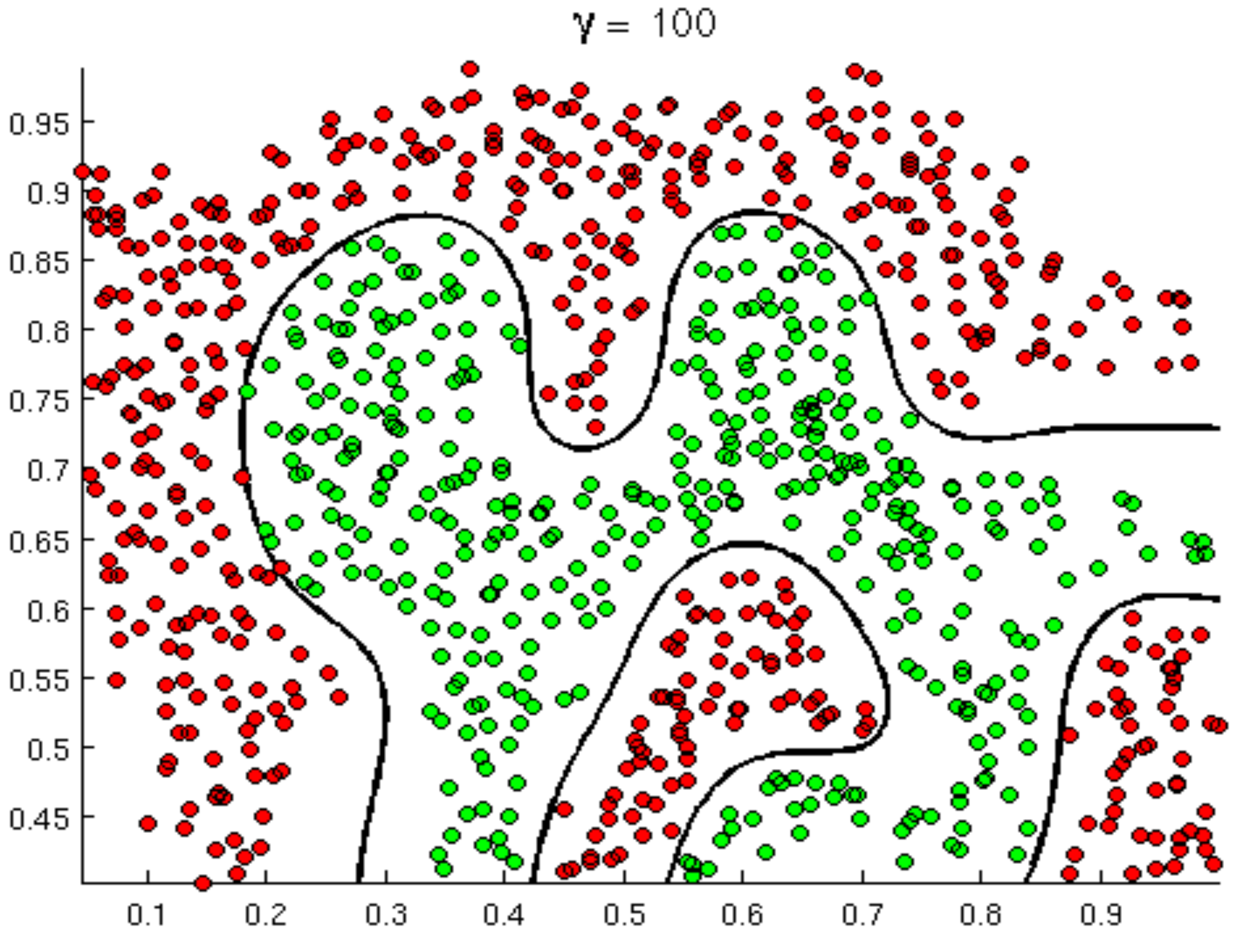
Logistic regression

- training example: (X, y)
- weight vector: W
- prediction: sigmoid(W^TX)
- cost: y * - log(sigmoid(W^TX))
  + (1 - y) * - log(1 - sigmoid(W^TX))
- regularization: ƛ



If $y = 1$, we want $\theta^T x \geq 1$ (not just ≥ 0)
If $y = 0$, we want $\theta^T x \leq -1$ (not just < 0)

Kernel SVM


γ = 100
0.95
0.9
0.85
0.8
0.75
0.7
0.65
0.6
0.55
0.5
0.45
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9

Linear SVM

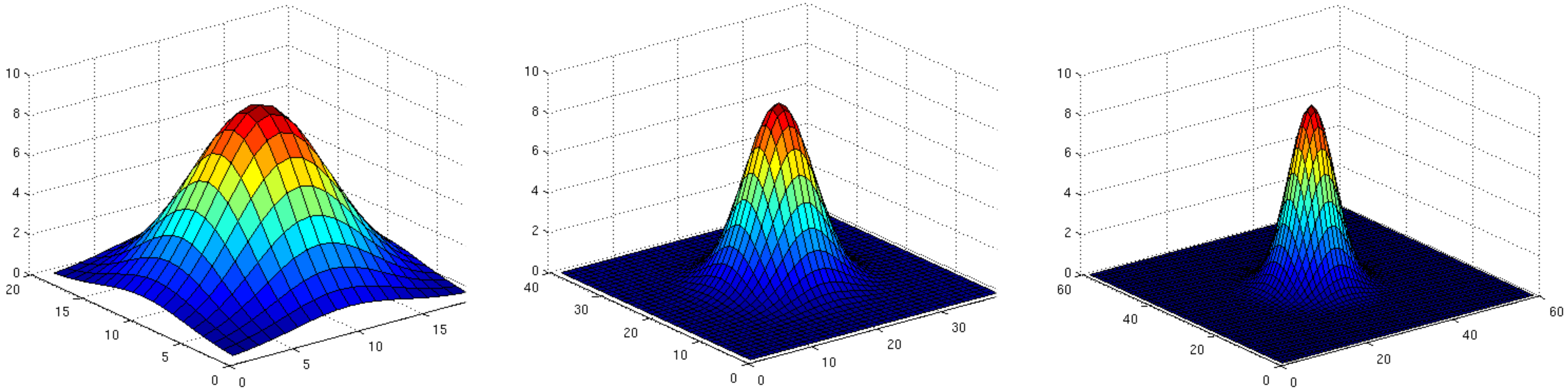
- training example: (X_{1xn}, y)

Kernel SVM

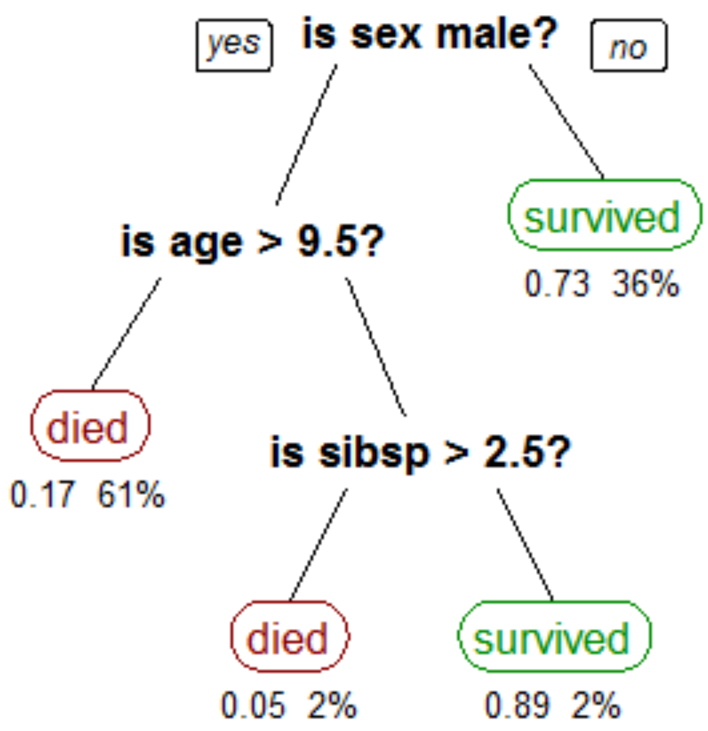
- training example: (D_{1xm}, y)
  D_{ij} = similarity of X_i with X_j

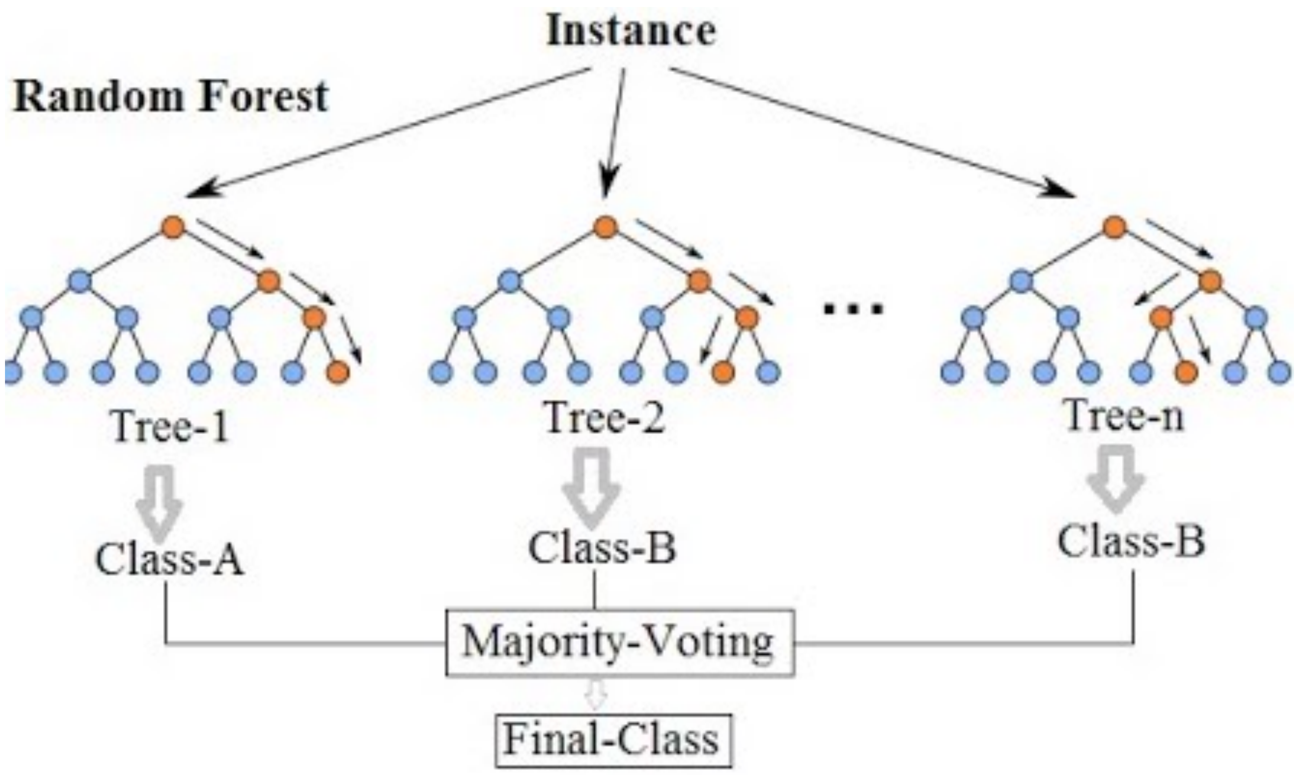
- Gaussian kernel:
  $D_{ij} = \exp(-|X_i - X_j|^2 / 2\sigma^2)$

increasing σ



Decision Trees


yes
is sex male?
no
is age > 9.5?
survived
0.73 36%
died
0.17 61%
is sibsp > 2.5?
died
0.05 2%
survived
0.89 2%


Instance
Random Forest
Tree-1
Tree-2
Tree-n
Class-A
Class-B
Class-B
Majority-Voting
Final-Class

Decision Tree Construction

- Entropy = unpredictability of outcome
- The decision tree is built from variables that reduce unpredictability i.e. the whole dataset entropy may be large, but the entropy of any one path through the decision tree is small
- For every variable, we partition the outcomes by possible value and measure whether such subsetting reduces the entropy. At each step we choose the variable with least entropy.
- e.g.
  entropy({examples where males survived, examples where males died})
  * weight of set
  + entropy({examples where females survived, examples where females died}})
  * weight of set

  vs. entropy(entire dataset)
- Stop when:
  - tree has a certain no. of levels
  - zero entropy on a path/no examples left

Ensembling

- Outcome of several trees feed into an assembler e.g. voting classifier
- Bootstrapping to create different training sets with replacement
  - Bagging – independent draws
  - Boosting – mispredicted examples get a higher probability of sampling