CS489/698
Lecture 23: March 28, 2018

Bagging and Distributed Computing

[RN] Sec. 18.10, [M] Sec. 16.2.5,
[B] Chap. 14, [HTF] Chap 15-16,
[D] Chap. 11

Boosting vs Bagging

- Review

Independent classifiers/predictors

- How can we obtain independent classifiers/predictors for bagging?

- Bootstrap sampling
  - Sample (without replacement) subset of data
- Random projection
  - Sample (without replacement) subset of features

- Learn different classifiers/predictors based on each data subset and feature subset

Bagging

For k = 1 to K

$\boldsymbol{D}_k \leftarrow$ sample data subset

$\boldsymbol{F}_k \leftarrow$ sample feature subset

$h_k \leftarrow$ train classifier/predictor based on $\boldsymbol{D}_k$ and $\boldsymbol{F}_k$

Classification: $majority(h_1(\boldsymbol{x}), \dots, h_K(\boldsymbol{x}))$

Regression: $average(h_1(\boldsymbol{x}), \dots, h_K(\boldsymbol{x}))$

Random forest: bag of decision trees

Application: Xbox 360 Kinect

- Microsoft Cambridge
- Body part recognition: supervised learning

Depth camera

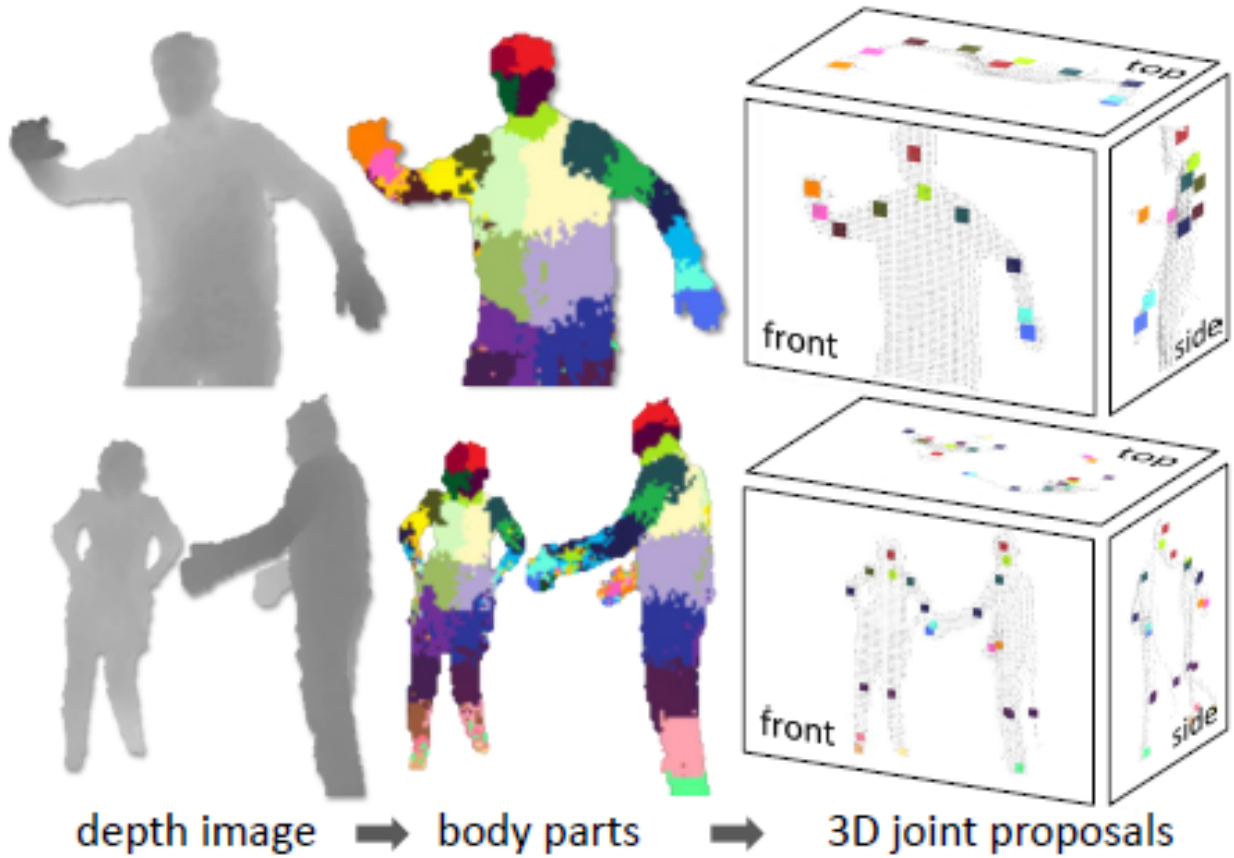
- Kinect

Infrared image

Gray scale depth map

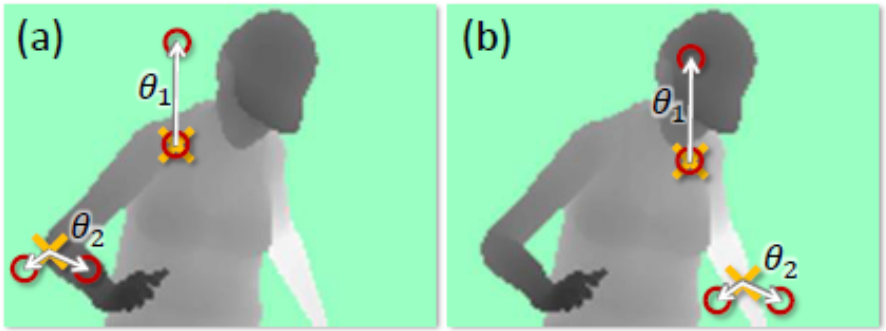
Kinect Body Part Recognition

- Problem: label each pixel with a body part

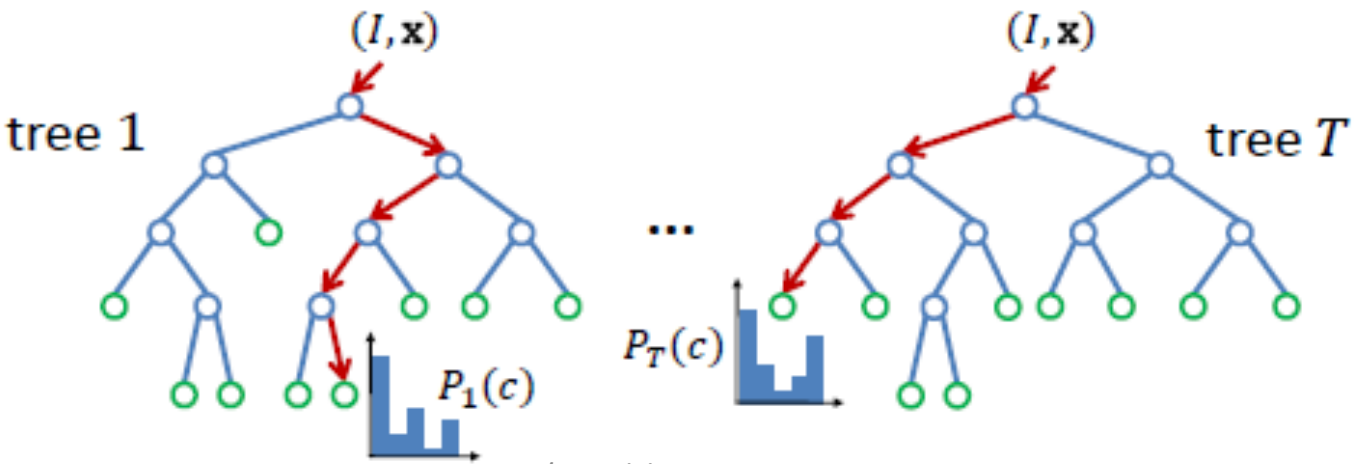


Kinect Body Part Recognition

- Features: depth differences between pairs of pixels



- Classification: forest of decision trees



Large Scale Machine Learning

- Big data
  - Large number of data instances
  - Large number of features

- Solution: distribute computation (parallel computation)
  - GPU (Graphics Processing Unit)
  - Many cores

GPU computation

- Many Machine Learning algorithms consist of vector, matrix and tensor operations
  - A tensor is a multidimensional array

- GPU (Graphics Processing Units) can perform arithmetic operations on all elements of a tensor in parallel

- Packages that facilitate ML programming on GPUs: Keras, PyTorch, TensorFlow, Theano, Caffe, DL4J

Multicore Computation

- Idea: Train a different classifier/predictor with a subset of the data on each core

- How can we combine the classifiers/predictors?
- Should we take the average of the parameters of the classifiers/predictors?

No, this might lead to a worse classifier/predictor. This is especially problematic for models with hidden variables/units such as neural networks and hidden Markov models

Bad case of parameter averaging

- Consider two threshold neural networks that encode the exclusive-or Boolean function

- Averaging the weights yields a new neural network that does not encode exclusive-or

Safely Combining Predictions

- A safe approach to ensemble learning is to **combine the predictions** (not the parameters)

- **Classification:** majority vote of the classes predicted by the classifiers

- **Regression:** average of the predictions computed by the regressors