Classification, Object Detection

Artificial Intelligence @ Allegheny College

Janyl Jumadinova

February 17, 2023

Object Recognition

Goal:
Find an object of a pre-defined class in a static image or video frame.

Object Recognition

Goal:

Find an object of a pre-defined class in a static image or video frame.

Approach:

- Extract certain image features, such as edges, color regions, textures, contours, etc.
- Use some heuristics to find configurations and/or combinations of those features specific to the object of interest.

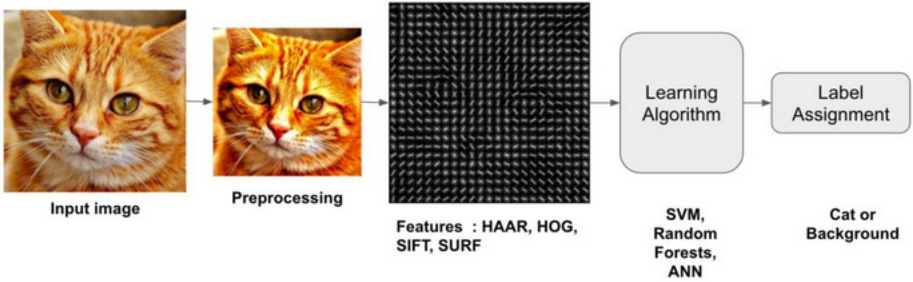
Statistical Model Training

- Training Set (Positive Samples/Negative Samples)
- Different features are extracted from the training samples and distinctive features that can be used to classify the object are selected.

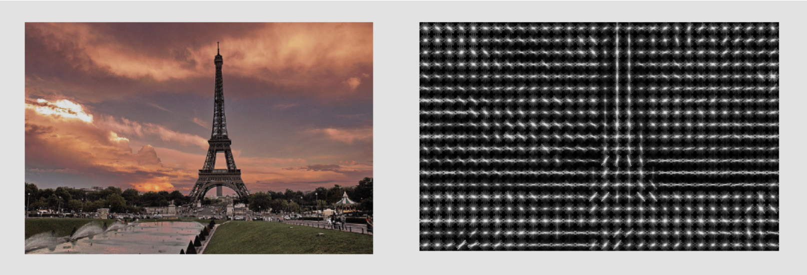
Statistical Model Training

- Training Set (Positive Samples/Negative Samples)
- Different features are extracted from the training samples and distinctive features that can be used to classify the object are selected.
- Each time the trained classifier does not detect an object (misses the object) or mistakenly detects the absent object (gives a false alarm), model is adjusted.

Process of Object Detection/Recognition

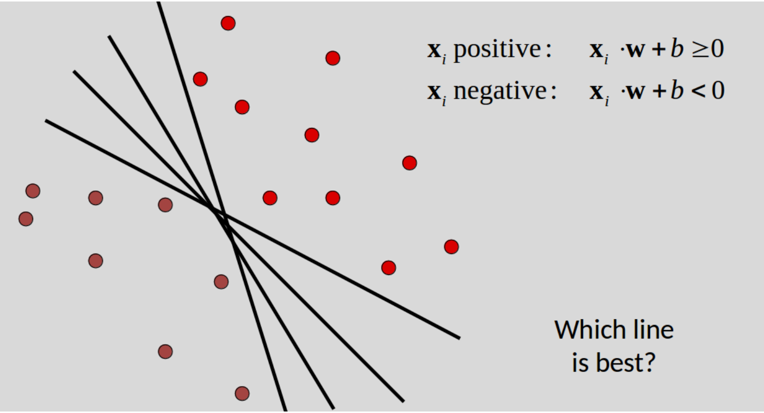


Histogram of Oriented Gradients (HoG)



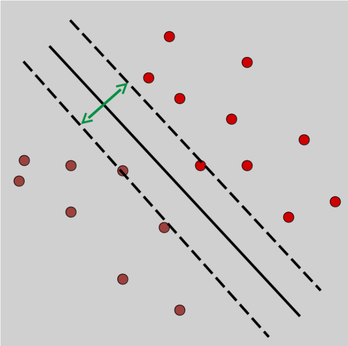
Linear classifiers

Find linear function to separate positive and negative examples



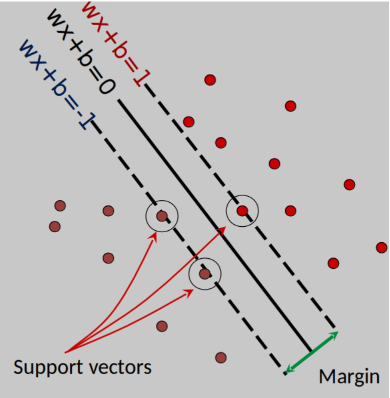
Support Vector Machines (SVMs)

- Discriminative classifier based on optimal separating line (for 2D case)
- Maximize the margin between the positive and negative training examples

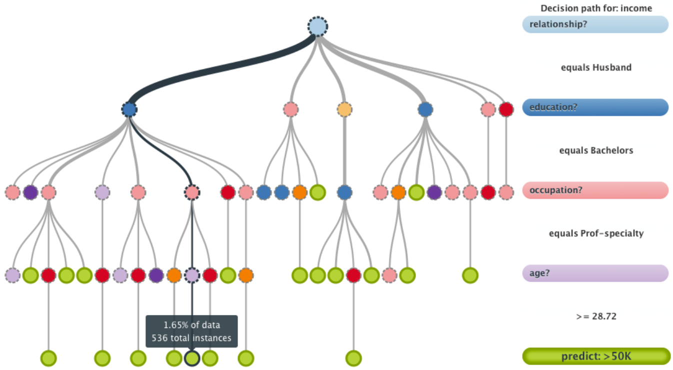


Support Vector Machines (SVMs)

- Want line that maximizes the margin.

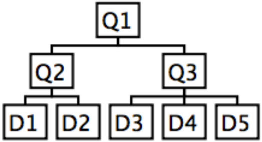


Decision Tree



Decision Tree

- Root node
  - Entry point to a collection of data
- Inner nodes (among which the root node)
  - A question is asked about data
  - One child node per possible answer
- Leaf nodes
  - Correspond to the decision to take (or conclusion to make) if reached



Decision Tree

- Represented by a series of binary splits.
- Each internal node represents a value query on one of the variables — e.g. "Is $X_3 > 0.4$". If the answer is "Yes", go right, else go left.

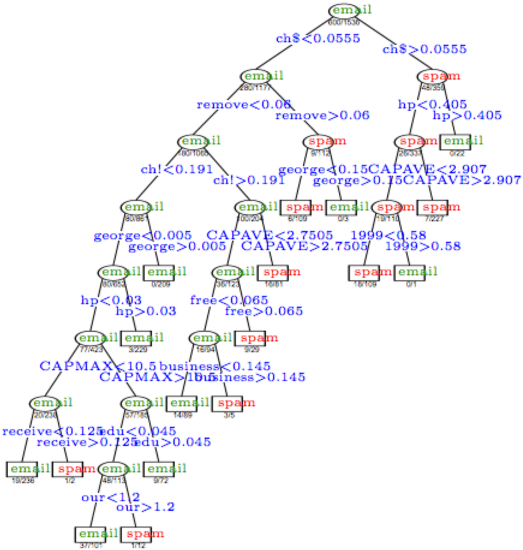
Decision Tree

- Represented by a series of binary splits.
- Each internal node represents a value query on one of the variables — e.g. "Is $X_3 > 0.4$". If the answer is "Yes", go right, else go left.
- The terminal nodes are the decision nodes.
- New observations are classified by passing their X down to a terminal node of the tree, and then using majority vote.

Decision Tree

- ✔ Can handle huge datasets
- ✔ Can handle mixed predictors—quantitative and qualitative
- ✔ Easily ignore redundant variables
- ✔ Handle missing data elegantly
- ✔ Small trees are easy to interpret
- ✘ large trees are hard to interpret
- ✘ Often prediction performance is poor

Decision Tree



Model Averaging

Classification trees can be simple, but often produce noisy and weak classifiers.

- **Bagging - averaging of trees**: Fit many large trees to bootstrap-resampled versions of the training data, and classify by majority vote.
- **Random Forests - smarter averaging of trees**: Fancier version of bagging.
- **Boosting - smartest averaging of trees**: Fit many large or small trees to reweighted versions of the training data. Classify by weighted majority vote.

Model Averaging

Classification trees can be simple, but often produce noisy and weak classifiers.

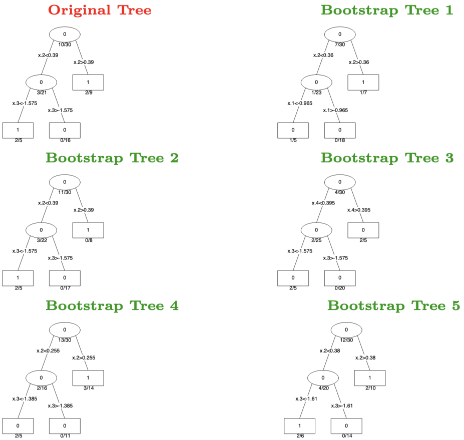
- **Bagging - averaging of trees**: Fit many large trees to bootstrap-resampled versions of the training data, and classify by majority vote.
- **Random Forests - smarter averaging of trees**: Fancier version of bagging.
- **Boosting - smartest averaging of trees**: Fit many large or small trees to reweighted versions of the training data. Classify by weighted majority vote.

In general,

$$\text{Boosting} \succ \text{Random Forests} \succ \text{Bagging} \succ \text{Single Tree}.$$

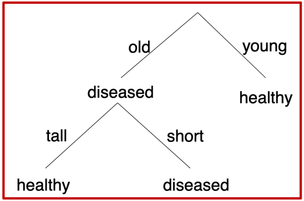
Bootstrap: basic idea

- Randomly draw datasets with replacement from the training data
- Each sample the same size as the original training set

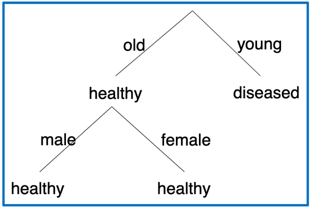


Random Forest

Tree 1



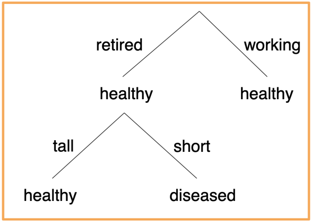
Tree 2



New sample:
old, retired, male, short
Tree predictions:
diseased, healthy, diseased

Majority rule:
diseased

Tree 3



Weak Classifier

- Computed feature value is used as input to a very simple decision tree classifier with 2 terminal nodes

$$\begin{cases} 1 & x_i \geq t_i \\ -1 & x_i \leq t_i \end{cases}$$

Boosted Classifier

- Complex and robust classifier is built out of multiple weak classifiers using a procedure called boosting.
- The boosted classifier is built iteratively as a weighted sum of weak classifiers.

Boosted Classifier

- Complex and robust classifier is built out of multiple weak classifiers using a procedure called boosting.
- The boosted classifier is built iteratively as a weighted sum of weak classifiers.
- On each iteration, a new weak classifier f_i is trained and added to the sum.

Boosted Classifier

- Complex and robust classifier is built out of multiple weak classifiers using a procedure called boosting.
- The boosted classifier is built iteratively as a weighted sum of weak classifiers.
- On each iteration, a new weak classifier f_i is trained and added to the sum.
- The smaller the error f_i gives on the training set, the larger is the coefficient/weight that is assigned to it.

scikit-learn

scikit-learn already contains many classifiers

https://scikit-learn.org/stable/modules/classes.html#module-sklearn.linear_model

https://scikit-learn.org/stable/modules/classes.html#module-sklearn.ensemble

Classification Summary

- **Support Vector Machines (SVMs)**:
  - works for linearly separable and linearly inseparable data; works well in a highly dimensional space (text classification)
  - inefficient to train; probably not applicable to most industry scale applications
- **Random Forest**:
  - handle high dimensional spaces well, as well as the large number of training data; has been shown to outperform others

Classification Summary

No Free Lunch Theorem:

Wolpert (1996) showed that in a noise-free scenario where the loss function is the misclassification rate, if one is interested in off-training-set error, then there are no a priori distinctions between learning algorithms. On average, they are all equivalent.

Classification Summary

No Free Lunch Theorem:

Wolpert (1996) showed that in a noise-free scenario where the loss function is the misclassification rate, if one is interested in off-training-set error, then there are no a priori distinctions between learning algorithms. On average, they are all equivalent.

Occam's Razor principle:

Use the least complicated algorithm that can address your needs and only go for something more complicated if strictly necessary.

"Do we Need Hundreds of Classifiers to Solve Real World Classification Problems?"

`http://jmlr.org/papers/volume15/delgado14a/delgado14a.pdf`