Object Detection

Haibin Ling

CIS 5543 – Computer Vision, Fall 2017

What is Object Detection?

• Locate an object in an input image

Viola & Jones, 2004

Dalal & Triggs, 2005

• Extensions
  – None or multiple objects
  – Object segmentation
  – Object detection in videos

Detection as Classification

• Binary classification
  \[ x \in \mathbb{R}^d \rightarrow f(x) \in \{+1,-1\} \]

Input feature vector Classifier function

□ Training/Learning – find the classifier function \( h(x) \)
\[ X = \{(x_i, y_i)\}_{i=1}^n \rightarrow f(x) \in \{+1,-1\} \]
Training samples Classifier

□ Example – face detection
\[ f(x) = +1 \quad \text{face} \]
\[ f(x) = -1 \quad \text{non-face} \]

Face Detection using AdaBoost

Viola & Jones, IJCV 2004

• Face detection
  – Fastest at the time, 15 frames per second

• Key ingredients
  – Feature selection – weak classifiers \( \leftrightarrow \) features
  – Fast feature evaluation – integral image
  – Fast classifier evaluation – classifier cascade

Principle of Boosting (Schapire’90)

• 三个臭皮匠，顶个诸葛亮
  Three cloggers with their wits combined, equal Zhuge Liang the master mind. (Or, two heads are better than one).

• In the language of learning: a strong classifier can be created by many weak classifiers.
\[ f(x) = f_1(x) + f_2(x) + f_3(x) + \cdots \]

□ Strong classifier: our target.
□ Weak classifier: better than chance, e.g., 51% classification rate.

Principle of Adaboost (Freund&Schapire’96)

• Different weak classifier have different capability
\[ f(x) = c_1 f_1(x) + c_2 f_2(x) + c_3 f_3(x) + \cdots \]

□ Failure is the mother of success
  □ Sequentially add weak classifiers
  □ Each new classifier, \( h_i \), focus on currently mis-classified samples.
Toy Example – taken from Antonio Torralba @MIT

Weak learners from the family of lines

Each data point has a class label:
\[ y_i = \begin{cases} +1 & \text{if } y_i = +1 \\ -1 & \text{if } y_i = -1 \end{cases} \]

and a weight:
\[ w_i = 1 \]

\[ h \iff \text{error} = 0.5 \text{ it is at chance} \]

This one seems to be the best

This is a 'weak classifier': it performs slightly better than chance.

Toy example

We set a new problem for which the previous weak classifier performs at chance again

Each data point has a class label:
\[ y_i = \begin{cases} +1 & \text{if } y_i = +1 \\ -1 & \text{if } y_i = -1 \end{cases} \]

We update the weights:
\[ w_i \leftarrow w_i \cdot \exp(-y_i f_t) \]

We set a new problem for which the previous weak classifier performs at chance again

Each data point has a class label:
\[ y_i = \begin{cases} +1 & \text{if } y_i = +1 \\ -1 & \text{if } y_i = -1 \end{cases} \]

We update the weights:
\[ w_i \leftarrow w_i \cdot \exp(-y_i f_t) \]

We set a new problem for which the previous weak classifier performs at chance again

Each data point has a class label:
\[ y_i = \begin{cases} +1 & \text{if } y_i = +1 \\ -1 & \text{if } y_i = -1 \end{cases} \]

We update the weights:
\[ w_i \leftarrow w_i \cdot \exp(-y_i f_t) \]
**Toy example**

The strong (non-linear) classifier is built as the combination of all the weak (linear) classifiers.

**Rectangular features**
- Two-rectangle features (A,B)
- Three-rectangle feature (C)
- Four-rectangle feature (D)

**Properties**
- Over complete
- Computationally efficient
- Feature computation → summing rectangle intensities

**Fast Feature Computation**

Integral Image
Integration along rows
\[ s(x,y) = s(x-1,y) + i(x,y) \]
Integration along columns
\[ i(x,y) = i(x-1,y) + s(x,y) \]

Using the integral image representation one can compute the value of any rectangular sum in constant time. The sum inside rectangle D we can compute as:
\[ i(4) + i(1) - i(2) - i(3) \]

**What are the features used?**
- Haar features?
  - There are over 45k different Haar features.
  - Computationally prohibitive.
  - Feature selection needed.
- Joint feature selection and classifier learning
  - Combine a subset of discriminative features to create an effective classifier
  - An effective classifier → Adaboost
  - Selected Haar features → Weak classifiers

**Speed-up through Cascading**

- Simple, boosted classifiers can reject many negative sub-windows while detecting all positive instances.
- Series of such simple classifiers can achieve good detection performance while eliminating the need for further processing of negative sub-windows

Training: subsequent classifiers are trained only on examples which pass through all the previous classifiers.
AdaBoost.M1

Discrete AdaBoost (Freund & Schapire 1996b)
1. Start with weights \( w_0 = 1/N, i = 1, \ldots, N \).
2. Repeat for \( m = 1, 2, \ldots, M \):
   (a) Fit the classifier \( f_m(x) \in \{-1, 1\} \) using weights \( w_i \) on the training data.
   (b) Compute \( \epsilon_m = E_{\epsilon_m} = E_{\epsilon_m}(1_{y \neq f_m(x)}) \).
   (c) Set \( w_i \leftarrow w_i \exp(\epsilon_m) \) if \( y \neq f_m(x) \), \( i = 1, \ldots, N \), and renormalize so that \( \sum w_i = 1 \).
3. Output the classifier \( \text{sign}(\sum_{m=1}^M \epsilon_m f_m(x)) \)

Additive Model

• Additive Model

\[
f(x) = \sum_{m=1}^M \beta_m b(x; \gamma_m)
\]

Fitting an additive model by minimize a loss function \( L(y, f(x)) \)

\[
(\beta_m, \gamma_m) = \arg \min_{\beta, \gamma} \sum_{i=1}^N L(y, f_m(x) + \beta b(x; \gamma))
\]

Forward Stagewise Additive Fitting

1. Input: \( X = \{(x_i, y_i)\}_{i=1}^N, y_i \in \{-1, 1\} \)
2. Initialization \( f^{(0)}(x) = 0 \)
3. For \( m = 1 \) to \( M \):
   (a) Compute \( (\beta_m, \gamma_m) = \arg \min_{\beta, \gamma} \sum_{i=1}^N L(y, f^{(m-1)}(x) + \beta b(x; \gamma)) \)
   (b) Set \( f^{(m)}(x) = f^{(m-1)}(x) + \beta_m b(x; \gamma_m) \)
4. Output \( f(x) = f^{(M)}(x) \)

This is AdaBoost!

For classification, use weak classifiers \( g(x) \in \{-1, 1\} \)

\[
(\beta_m, g_m) = \arg \min_{\beta, g} \sum_{i=1}^N w^{(m)} \exp\left(-\beta y f(x) + g(x)\right)
\]

- Solve it, we have
  \[
g_m = \arg \min_{g} \sum_{i=1}^N w^{(m)} I(y, g_m(x))
  \]
  - Fitting the training data with current weights \( w^{(m)} \)
  - Coefficient for weak classifier \( g_m(x) \)
  \[
  \beta_m = \frac{1}{2} \log \frac{1 - \epsilon_m}{\epsilon_m}
  \]
  - Weighted training error

Fitting with Exponential Loss

- Exponential loss function
  \[
  L(y, f(x)) = \exp(-y f(x))
  \]

- For each component
  \[
  (\beta_m, \gamma_m) = \arg \min_{\beta, \gamma} \sum_{i=1}^N L(y, f^{(m-1)}(x) + \beta b(x; \gamma))
  \]
  \[
  = \arg \min_{\beta, \gamma} \sum_{i=1}^N \exp\left[-y (f^{(m-1)}(x) + \beta b(x; \gamma))\right]
  \]
  \[
  = \arg \min_{\beta, \gamma} \sum_{i=1}^N w^{(m)} \exp\left[-\beta y b(x; \gamma)\right]
  \]
  \[
  w^{(m)} = \exp(-y f^{(m-1)}(x))
  \]

Object Detection using Histogram of Oriented Gradient (HOG)

Dalal & Triggs, CVPR 2005

- Focus on creating robust encoding of images
- Linear SVM as classifier on normalized image windows, is reliable & fast
- Moving window based detector with non-maximum suppression over scale space
Overall Architecture

Learning Phase
- Create normalised training data set
- Encode images into feature vectors
- Learn binary classifier

Object/Non-object decision

Detection Phase
- Scan image at all scales and locations
- Run classifier to obtain object/non-object decisions
- Fuse multiple detections in 3-D position & scale space
- Run classifier to obtain object/non-object decisions
- Scan image at all scales and locations
- Run classifier to obtain object/non-object decisions

Object detections with bounding boxes

Multi-Scale Detection

After dense multi-scale scan of detection window
- Clip Detection Score
- Map each detection to 3D [x,y, scale] space
- \[ f(x) = \sum_{i=1}^{n} \exp \left( \frac{-||x_i - x||^2}{2\sigma^2} \right) \]
- Apply robust mode detection, like mean shift.

Final detections

Descriptor Processing Chain

Object/Non-object
- Linear SVM
- Collect HOGs over detection window
- Contrast normalize over overlapping spatial cells
- Weighted vote in spatial & orientation cells
- Compute gradients
- Gamma compression
- Image Window

Descriptor cues: Motorbikes

Average gradients
- Weighted pos wts
- Weighted neg wts
- Input window
- Detection Examples
- Dominant pos orientations
- Dominant neg orientations

HOG for Human Detection

General Object Detection

Object Detection with Discriminatively Trained Part Based Models, P. Felzenszwalb, R. Girshick, D. McAllester, D. Ramanan, PAMI, 2010
Deep Learning for Object Detection

Yolo 2

http://pulsdms.com/yolo