COS 324, Precept #9: Face Detection: A Case Study for Boosting

December 4, 2017

1 Overview

Face recognition has become a sufficiently mature and reliable technology that it's running on your smartphone camera software. Nowadays, state-of-the-art face-recognition performance is achieved by *convolutional neural networks*, which we might discuss later in the course. But before deep learning took the throne, the dominant approach was a simple application of boosting, and is still used in situations where efficiency demands overshadow performance requirements (say, to focus on your face when you take a selfie).

We'll go through a simplified version of the Viola-Jones framework for face detection, a very important example of boosting in the real world.

2 Weak learners for faces

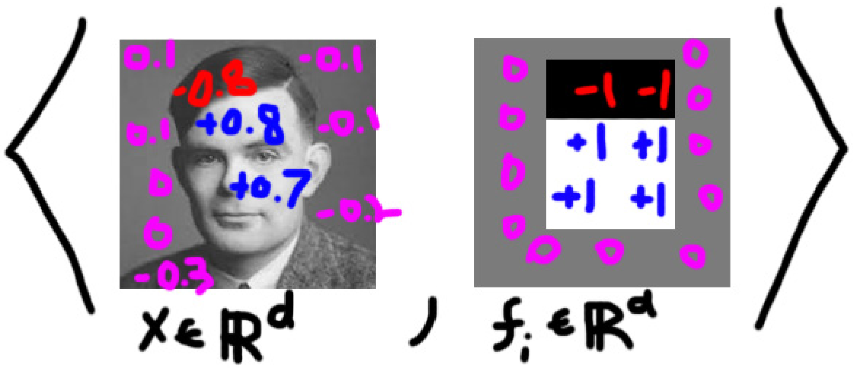
Concretely, we will focus on the following problem: given a normalized 100-by-100 pixel grayscale photograph $x \in \mathbb{R}^{10000}$, does it contain a human face? We might imagine having some dataset with positive examples (photos of faces) and negative examples (photos of buildings, trees, sky, etc.).

Informally, boosting is a meta-algorithm for training an ensemble of weak learners (simple hypotheses which are correct 51% of the time) to obtain a strong learner (a more expressive hypothesis which is correct 99% of the time).

What are some rules of thumb for detecting human faces? (see Google image search for "grayscale human face")

- A face tends to have two dark regions near the upper-center of the image. (eyes)
- A face tends to have a symmetric dark-light-dark region near the center. (nose)
- A face tends to have a dark or light region on top. (hair)

We can encode these rules of thumb as *linear classifiers*, sometimes called Haar features. To detect hair, we could consider measuring the correlation of the image x with a rough "image template" f_{hair} for hair:



Then, a classification rule could be $\text{sgn}(\langle x, f_{\text{hair}} \rangle)$. [1]

In general, we can define a family of linear classifiers by a collection of templates f_i, which are restricted to contain a positive and negative rectangle.

Unlike general linear classifiers, this is a finite hypothesis class! In an n-by-n image, there are $O(n^4)$ rectangles, so the number of such filters (sometimes called Haar filters) is (crudely) bounded by $O(n^8)$. So, by the agnostic version of the fundamental theorem of PAC learning, one can hope for good sample complexity.

There's no need to be scared about the exponent in $|\mathcal{H}|$. Statistically: it shows up as a constant in $\log |\mathcal{H}|$ in the PAC generalization bound. Computationally: our task in ERM is to find the f_i which achieves the highest training accuracy. This is a very structured problem in practice; one could hope to come up with faster algorithms than brute force, and much faster heuristic algorithms.

3 Boosting

To translate what we have so far into the language of boosting:

- The weak learner is an algorithm that is given a (weighted) dataset, and outputs the best "rectangular template" classifier.
- The assumption, in order to get our boosting guarantee, is that on any distribution of faces, the accuracy of the weak learner exceeds $\frac{1}{2} + \gamma$ (say, $\gamma = 0.01$).

We can run the AdaBoost meta-algorithm for training, as covered in lecture:

- Start with the dataset S, which has m samples $(x_i \in \mathbb{R}^{10000}, y_i \in \{\pm 1\})$

[1] In reality, we might want to incorporate a bias term.

- Maintain weights $p_t \in \mathbb{R}^m$, initially uniform ($p_1 = \mathbf{1}/m$).
- For $t = 1, \ldots, T$:
  - Weak learner outputs a hypothesis f_t whose predictions $\operatorname{sgn}(\langle f_t, x\rangle)$ agrees best with the distribution p_t over S:

    $$f_t := \operatorname*{argmin}_{f \in \mathcal{H}} p_t^\top (\mathbf{1}_{\operatorname{sgn}(\langle f_t, x_i\rangle) = y_i})$$

  - Let $\varepsilon_t := p_t^\top (\mathbf{1}_{\operatorname{sgn}(\langle f_t, x_i\rangle) = y_i}) - \frac{1}{2}$. (Recall weak learner assumption: $\varepsilon_t \geq \gamma$.)
  - Multiplicative weight update: for only correctly classified points x_i,

    $$p_{t+1}(i) := p_t(i) \cdot (1 - \varepsilon_t)$$

  - Normalize the distribution: $p_{t+1} \leftarrow \frac{p_{t+1}}{\|p_{t+1}\|_1}$.
- When given a new image x, return the weighted majority vote of weak learner's T outputs:

  $$\mathsf{Prediction}(x) = \operatorname{sgn}\left(\sum_{t=1}^{T} \varepsilon_t \cdot \operatorname{sgn}(\langle f_t, x\rangle)\right).$$

Some notes on what happens:

- By the end of $T = O(\log m/\gamma^2)$ rounds, assuming the weak learner was successful every time, the resulting ("majority-of-rules-of-thumb") hypothesis T classifies the training data perfectly.
- Note that this hypothesis class is more expressive than $\mathcal{H}$. As mentioned in lecture, despite being more expressive, this doesn't overfit. A technical formulation is outside the scope for the course.
  - An incomplete intuition: the weighted majority of simple hypotheses isn't that much "larger" than the simple hypotheses themselves. For those interested: the slickest way to reason about this involves bounding the *Rademacher complexity* (a more fine-grained analogue of $\log |\mathcal{H}|$) of a convex combination of learners.
- Intuitive interpretation: by decreasing the weight on correctly classified points, AdaBoost places emphasis on harder-to-classify points. Over T rounds, AdaBoost asks the weak learner to classify harder and harder datasets.
- Where the regret bound comes in: the average of distributions played by AdaBoost (which is another distribution) is close to the hardest possible distribution for the weak learner.

4 Adversarial examples

An exciting direction in machine learning is *defending against adversarial examples.* Suppose you've built a face verification system into Apple ID, and it works quite well on your training data. You start a high-tech biometric security startup, and users start to use your system to gain access to buildings, or log into protected systems.

Do we expect this boosting method to be robust to arbitrary (malicious) test examples? Certainly not- imagine a face with strategically taped-on LEDs. Or a cat's face. Or a Piet Mondrian painting.

- Fooling the Viola-Jones boosting method: `https://ahprojects.com/projects/hyperface/`
- Article with fun pictures (and a video): `https://io9.gizmodo.com/how-fashion-can-be-used-to-`
- Recent news, fooling the iPhone X's FaceID (which uses a different method): `https://www.theverge.com/2017/11/13/16642690/bkav-iphone-x-faceid-mask`
- Happens to humans too: `https://en.wikipedia.org/wiki/Pareidolia`

This is extremely worrisome– imagine an autonomous driving system, which classifies visual input from an on-board camera. At a busy downtown intersection, I could print a billboard which causes the system to confuse "pedestrian ahead" with "merge into traffic". This is a major area of active research.