ADA Boost Face Detection

1

Find Faces

2

Imagine a Stock Market Application

In the investment field, many experts get paid for their stock market forecasts.

Suppose a Rich Person (RP) who would like to hire a small team of Experts.

The RP would like to have a team of size, say, 50.

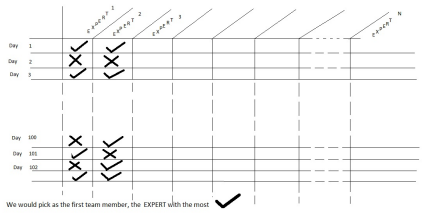
The RP has access to the performance records of the 10,000 experts for the past 5 years (which is 1,000 days, since each year has 200 stock-trading days).

Performance is judged by this: The market opens at 9:30am EST each trading day. At 8:30am each expert announces whether s/he thinks the market will go UP that day or not. Here, UP is defined as, perhaps, the Dow up by at least 20 points. At market close (typically, at 4pm), we know if each expert was right or not. Performance does not care about whether market went up or not, just cares about whether the expert called it correctly.

3

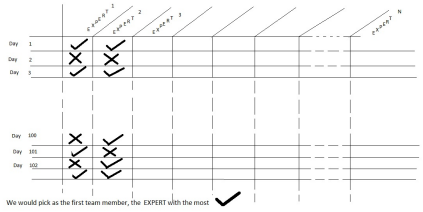
Imagine a Stock Market Application

So, our RP has access to a table that looks like this:



4

Picking the first member



It is a correct move to pick the first member as the one with the most correct forecasts

5

Picking the second member

- So, what should our strategy be to pick the second team member?
- Please spend a minute thinking about this and e-mail the instructor what you think.

We will pause here, while you finish this task.

6

Picking the second member

- Most students and others generally think that the correct thing to do is to pick the expert who has the next highest correct score.
- And that is what machine learning thinkers thought till Schapire (pronounced Sha-Pee-Ray) came along.
- Turns out the 2nd should be the expert who gets the most correct of the examples that the 1st expert got wrong!! i.e., complements the existing team

7

Picking the next member

- Pick from the available pool of experts, the one who most complements the existing (selected) team
- Note: each expert is assumed to be a Weak Expert, named so for being correct between 51% and somewhere in the 80's or low-90's percent of the time.
- BOOSTING means picking a team of weak experts, so the team performance is strong (close to 99.9%).

8

Observations about Picking the team

- Note: if an expert was worse than a Weak Expert, i.e., was correct less than 49% of the time, one would "force" it to be a weak expert by simply flipping its answers. So, if it said UP, the system would flip it to NOT-UP, and vice versa.
- The only time one cannot benefit from the flip is if the expert is exactly at 50%. This is very rare, and these are discarded from the pool of experts.

9

Recap:
Imagine a Stock Market Application

In the investment field, many experts get paid for their stock market forecasts.

Suppose a Rich Person (RP) who would like to hire a small team of Experts.

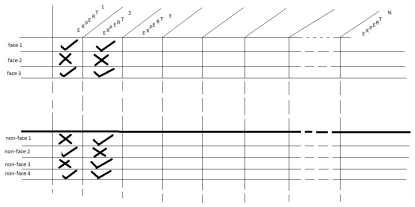
The RP would like to have a team of size, say, 50.
The RP has access to the performance records of the 10,000 experts for the past 5 years (which is 1,000 days, since each year has 200 stock-trading days).

Now, the actual problem can be stated fully:
Once the team (of 50) is selected, its job is to look at a new day (8:30am) and make a forecast (we will deal later with how the team functions as a team, i.e., how decisions will be made in a collaborative manner).

Realize that without that final task (to decide about a new day), the process would have been a waste of time. There is a Training Stage & a Testing Stage.

10

Now, do the vision problem



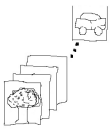
So, the problem can be phrased as: an expert is presented with a fixed-size picture, and needs to say if it is a face or not.

11

For Training Stage: Faces database

12

Training Stage: Non-Faces database

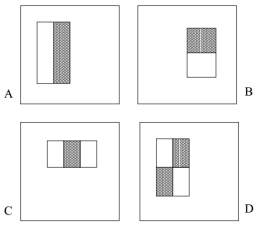


13

What is an expert?

14

Start with a pattern from below

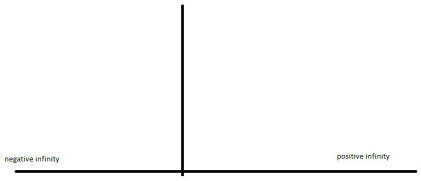


In the patterns above, the numbers in the region of the white bars are +1, the black bars have numbers of -1, and the surround background numbers have 0. The size of each pattern is the same as the size of a training image, typically something like 50x50 pixels.

15

Consider the number line

In all these plots of the number-line, the depiction of the y-axis is irrelevant, and is only included to indicate that at that position the numbers switch from negative to positive.

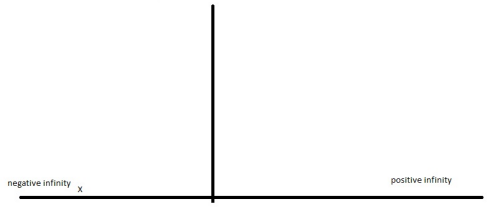


16

Convolution of pattern with first face

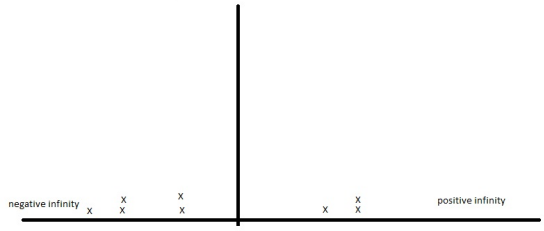
Our Convolution table has a combination of 0's, +1's, and -1's.
Hence, a one-step convolution between the table and our first face is some integer between negative infinity and positive infinity.

So, plot this resulting integer on the number line.

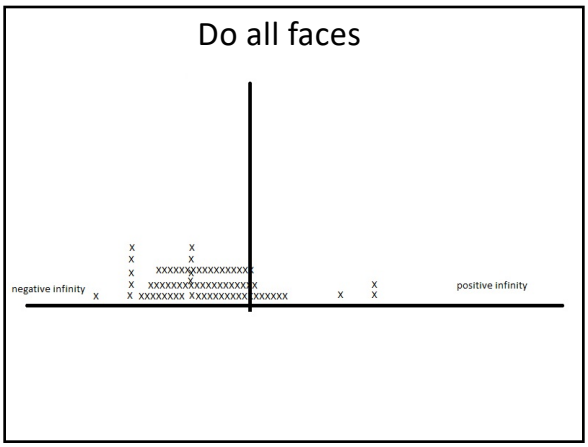


17

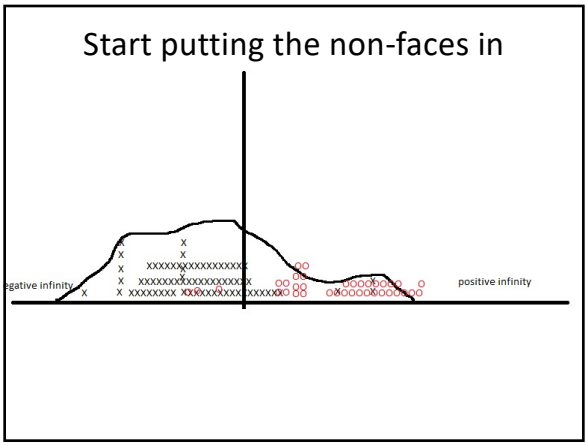
Bring in the second face, and more



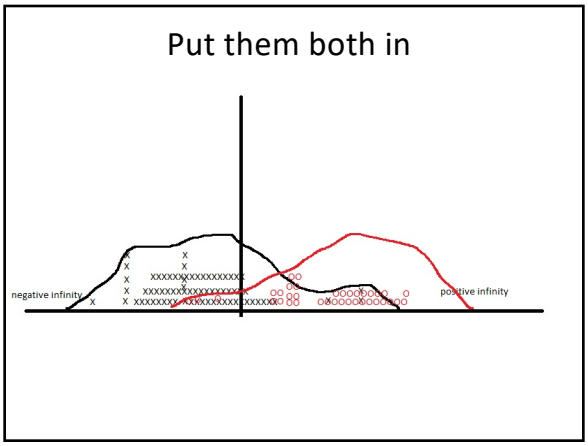
18


Do all faces
negative infinity
positive infinity

19


Start putting the non-faces in
positive infinity

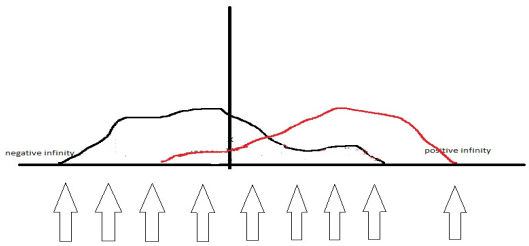
20


Put them both in
negative infinity
positive infinity

21

Start search for THRESHOLD

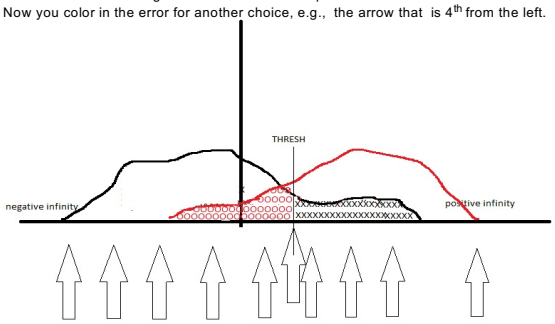
We consider a candidate Threshold (each arrow represents a possible Threshold).
For it, we then ask: Suppose we labeled its left-hand side as the realm of the negatives, and the right hand side as the realm of the positives; then what would be the error?
i.e., given that parity-sign label, how many example actual positives end up on its left, and how many actual negatives end up on its right, they constitute its error.



22

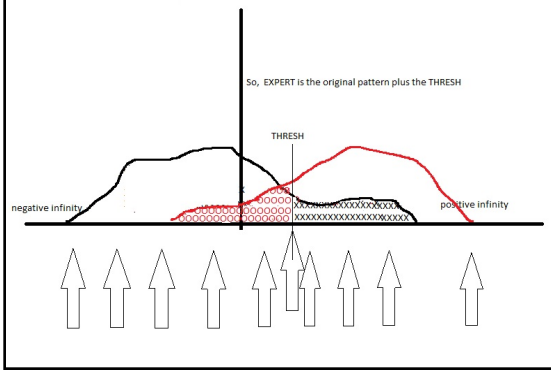
Find the best (i.e., lowest error)

Verify for yourself, that placements other than the one shown as best, would be worse.
We have shown coloring in the error for the best placement.
Now you color in the error for another choice, e.g., the arrow that is 4th from the left.



23

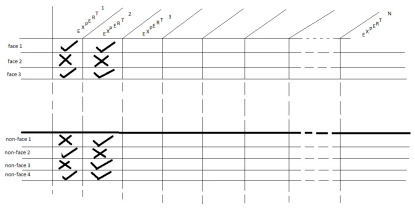
Find the best Threshhold



24

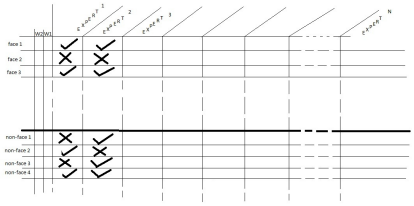
Remember …

So, now we are able to explain how the earlier table is obtained. i.e., once we have all our experts created (many hundreds of thousands of them, why so many???), we see how this table can be populated.



25

To facilitate selection, Add weights

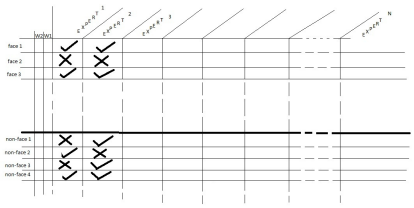


The idea here is to not have to keep track of what experts are complementing others and so on, and instead to have a numerical algorithm to achieve the same goals.

This is done using weights attached to each training example.
A training example is a member of our database. The weights will keep a history of how difficult an example is to be conquered (to be classified correctly).

26

Using weights attached to examples



The basic idea is that with each selection of an expert, the weights of examples that are not yet conquered, their weights will grow. At time to select, add every expert's error as the sum of the weights for the examples it has gotten wrong.

So, after many selections, say 45, if there are examples that are not yet conquered, their weights will be quite high so they will contribute high errors to the experts who get them wrong (which include the current team), but will contribute nothing to a fresh expert who has not been picked so far (because it gets fewer correct). So, now, it is its turn to be picked because it is conquering a really, really bad example (and hence it has a lower error).

27

Algorithm 1

The Algorithm boxes shown in these slides are from the Viola Jones paper.

- Given example images $(x_1, y_1), \ldots, (x_n, y_n)$ where $y_i = 0, 1$ for negative and positive examples respectively.
- Initialize weights $w_{1,i} = \frac{1}{2m}, \frac{1}{2l}$ for $y_i = 0, 1$ respectively, where m and l are the number of negatives and positives respectively.
- For $t = 1, \ldots, T$:

28

Algorithm 2

- For $t = 1, \ldots, T$:
  1. Normalize the weights,
  $$w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,j}}$$
  so that w_t is a probability distribution.
  2. For each feature, j, train a classifier h_j which is restricted to using a single feature. The error is evaluated with respect to w_t, $\epsilon_j = \sum_i w_i |h_j(x_i) - y_i|$.
  3. Choose the classifier, h_t, with the lowest error ϵ_t.
  4. Update the weights:
  $$w_{t+1,i} = w_{t,i}\beta_t^{1-e_i}$$
  where $e_i = 0$ if example x_i is classified correctly, $e_i = 1$ otherwise, and $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$.

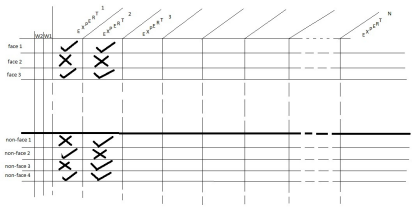
29

Algorithm 3

- The final strong classifier is:
$$h(x) = \begin{cases} 1 & \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0 & \text{otherwise} \end{cases}$$
where $\alpha_t = \log \frac{1}{\beta_t}$

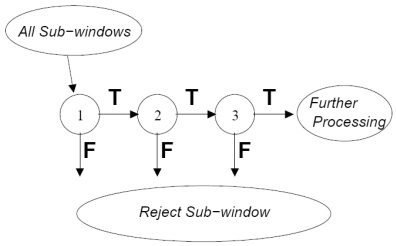
30

So, let's review how weights work



31

Exploit that faces are rare



32