Unsupervised Improvement of Visual Detectors using Co-Training

Anat Levin, Paul Viola, Yoav Freund

ICCV 2003

UCSD CSE252c
presented by William Beaver
October 25, 2005
Task

- Traffic camera taking video
- You want to identify the cars
- Simple
  - pick a detector
  - gather training data
  - train
Training data needs

• Use a [Viola, Jones IJCV 2002] approach

• image size: 320x240 ⇒ ~50,000 locations to scan per frame

• a false positive rate of 1 in 10,000 ⇒ ~5 false positives per frame! ⇒ need a very low false positive rate

• ⇒ many examples needed

  • $O(10^9)$ negative examples and “several thousand” labeled positive examples

The real problem

the high cost of generating labeled examples
Unlabeled examples are typically easy to come by.

Unlabeled examples have been shown useful in text classification [Nigam, et al]

Assumption: unlabeled examples respect the class boundaries of the underlying distribution (typically Gaussian, two class mixtures)

density of unlabeled examples low near the class boundary
Margin concept

• [Viola, Jones] use Adaboost.
  • discriminative.
  • maximize margin
• Unambiguous examples are not informative if far from the margin
• Need
  • confident AND
  • examples with small margin (or negative)
• How do we get these examples?

*Thanks Tony Jebara for the ‘inspiration’
Adaboost \cite{Freund/Schapire 97}

Given: \((x_1, y_1), \ldots, (x_m, y_m)\) where \(x_i \in X, y_i \in Y = \{-1, +1\}\)

Initialize: \(D_1(i) = 1/m\)

For \(t = 1 : T\)

Train base learner using distribution \(D_t\)

Get base classifier (weak) \(h_t : X \rightarrow \mathbb{R}\)

choose \(\alpha_t \in \mathbb{R}\)

\[\alpha_t = \frac{1}{2} \ln \left( \frac{1 - \epsilon_t}{\epsilon_t} \right)\]

update \(D_{t+1}(i) = \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t}\)

\(H(x) = \text{sign}(\sum_{t=1}^{T} \alpha_t h_t(x))\)
Adaboost - margin

• Typically, expect Adaboost generalization error at most

\[ \hat{\Pr}[H(x) \neq y] + O(\sqrt{\frac{Td}{m}}) \]

\[ \text{margin}_{f(x,y)} = \frac{y \sum_t \alpha_t h_t(x)}{\sum_t |\alpha_t|} \in [-1, +1] \]

positive iff H correctly classifies correctly
margin magnitude is a measure of confidence

[Freund, Schapire 99]
Adaboost - margin

- larger margin on training $\Rightarrow$ superior upper bound on generalization error (test)

$$\Pr[f(\tilde{h}_i \leq 0) \leq \frac{1}{m} \sum_{i=1}^{m} [f(\tilde{h}_i) \leq \theta] + O\left(\frac{1}{\sqrt{m}} \left(\frac{d \log^2(m/d)}{\theta^2} + \log(1/\delta)\right)^{\frac{1}{2}}\right)$$

<table>
<thead>
<tr>
<th>Fraction of training samples with margin less than $\theta$</th>
<th>proportion of testing examples in error</th>
</tr>
</thead>
</table>

$$\hat{\Pr}[\text{margin}(x, y) \leq \theta] + \tilde{O}\left(\sqrt{\frac{d}{m\theta^2}}\right) \quad \forall \theta > 0$$

[Schapire, Freund,Bartlett, Lee 98]
Co-Training

• Co-Training [Blum, Mitchell COLT 98]
  • take two views of data.
  • there may exist set of examples with high margin using one feature and small (or negative) margin on another
  • (there is a formal framework if people are interested.)
Co-Training

- [Blum, Mitchell 98] proves co-training finds a very accurate rule from very small quantity of labeled data

- Co-Training assumptions:
  1. reasonable learning algorithm
  2. underlying dist which satisfies conditional independence
  3. initial weak classification rule
Co-Training

- Practically: train a pair of detectors
  - examples *confidently* labeled by one are used to correct a mislabeling by the other
  - margin magnitude give *confidence*
- train each classifier with *informative* examples
- Conditional independence assumption hard in practice
Approach

- Using this idea, focused on margin classifiers:
  1. train two classifiers with small dataset
  2. estimate margin thresholds above/below which all training data is correctly labeled
  3. use confidence to assign labels to unlabeled data
  4. add these examples and retrain
  5. repeat...
Details - data

• data from Washington State Department of Transit website
  • 15 sec video clips
  • 1 clip every 5mins.
  • 8 cameras.
  • three weeks.

Figure 1: Example images used to test and train the car detection system. On the left are the original images. On the right are the results of the detection system.
Classifiers

- Two Classifiers ("views" of the data)
  - gray image = what you’d expect
  - background difference = image - average background from clip.
Details - data

- data - training
- 50 cars identified in total (that’s all!) images from 3 of the cameras.
- box drawn around car (minimize background).
- each cropped and scaled to 20 x 28 (aspect ratio 1.4). if smaller, discard
- 22,000 additional unlabeled images available.
Details - data

• data - validation

• 6 separate images.

• every car labeled.

• practical: chosen to have as many positive examples as possible. want to be sure that true positives are not labeled as negative
Initial training

- Each classifier trained identically and independently

- Input to classifier construction:
  target detection rate (100%),
  target false positive rate (0.2%),
  training data (+:50 cars; -:?: images ),
  validation dataset

- Features: selected via LogAdaBoost [Collins et al]
  - Feature = simple linear function of rectangular sums and a threshold.

- Each round: select feature with lowest weighted error

- Selected feature assigned a weight based on performance
Initial training - details

• 320 x 240 images → >50,000 locations to check. For efficiency, use a cascade of classifiers

• early classifiers constrained to use few features but have high detection rate.

• later classifiers have more features and lower detection rate; only trained on true and false positives of earlier stages

• classifier construction runs until rate met.

• Use the validation images to adjust threshold. Run classifier and tune until reach required detection rate.
Co-Training

• initial classifiers: two, 5 stage cascades. each stage has 4 features. (20 features total)

• First, retrain final stage of cascade to increase number of features (to 30). This increases confidence range value.

• scan 22k unlabeled examples. (1.1 billion locations!)

• sample from these scans

• Use classification scores - signed measure of confidence - to label
Sampling issues

• how to sample negatives: there are many to choose from.

• uniform sampling? Ignores margin! need to know which are important

• importance sampling: randomly selecting using a distribution proportional to the examples weight \((w_i)\)

scan data once,

if \(N\alpha w_i < 1\)

select with \(Pr = (N\alpha w_i)\) and assign weight = 1.

else // \(N\alpha w_i \geq 1\)

\(Pr = 1\) and weight = \(N\alpha w_i\)
Sampling issues

- how to sample positives
- challenge is alignment: only select peaks (local max) of scoring function.
- weight assigned to each peak is sum of weights above the selection threshold and nearby the peak (not well defined)
- **Margin plot**: determine thresholds by performance on the validation set

- $\theta_p$ (pos threshold) is max score achieved by negative examples (above is likely to be positive)

- $\theta_n$ (neg threshold) is min score achieved by positive examples (below likely to be negative)
co-training

• new example sampled and labeled via the thresholds
• Then add, with label, to training set
• now, boost and add three new features to classifier.
• resample unlabeled data again.
• 12 rounds gives 66 features (30 orig + (3 x 12))
• cascade helping two problems:
  • asymmetry: there are few positive examples in an image while there are thousands of negative examples. the cascade increases the ratio
  • efficiency: do not need every example. only those which pass the early cascade stages
Testing

- set of hand labeled examples. not used anywhere in training.
- 90 images
- 980 positive examples
- test those that make it through the additional cascade on the co-trained layer
The image shows two ROC curves. The left curve represents the grey classifier, and the right curve represents the backsub classifier. The green line indicates the original classifier, while the black line represents the co-trained classifier.
Figure 5: Detection results. (a)- The gray level classifier before co-training. (b)- The gray level after co-training. (c)- The backsub before co-training. (d)- The backsub after co-training.

Robust real-time object detection.

Algorithm/Process Summary

- Initial setup
  - Train two detectors
    - Detector One: operates on grey level images
    - Detector Two: operates on difference images
    - Each detector has 5 classifier stages each containing 4 features.
    - Detection rate on training set is 100%; false positive is 0.2%.
  - Final stage is retrained to contain 30 features, which produces more reliable classification scores). Note: the scores assigned by this final stage are used to select additional examples for co-training.
  - Two thresholds are computed $\theta_p$ and $\theta_n$.
    - $\theta_p$ = max score achieved by a negative patch from the validation set
    - $\theta_n$ = min score achieved by a positive patch from the validation set

- Co-training process (run for 12 rounds)
  - 22,000 unlabeled images are scanned.
  - In each range of co-training patches are sampled from the 22,000 unlabeled images
    - Image patches labeled positive by the 5 stage cascade are examined
    - Positive patches are extracted if score is greater than $\theta_p$
      - Pick local maxima in the score function (to improve alignment)
    - Negative patches have score less than $\theta_n$
      - Since there are so many negative examples, examples are selected at random based on AdaBoost weight
  - The final stage of the classifier is augmented with 3 additional features using this new training data.