

Agnostic Active Learning

Jack Bowler

EECS 6890 Presentatoin

1 May 2014

Outline

- 1 Introduction
 - Authors
 - Preliminaries
 - contribution
- 2 A² Algorithm
- 3 Theorems
 - Correctness
 - Fall back Analysis
- 4 Improvements from A²
 - Exponential Improvement
 - Effect of Large Noise
 - Linear Separators under Uniform Distribution
- 5 Open Questions

Authors

- Maria-Florina Balcan
Carnegie Mellon University, Pittsburgh, PA
- Alina Beygelzimer
IBM T.J. Watson Research Center, Hawthorne, NY
- John Langford
Yahoo! research, New York, NY

Agnostic active learning In ICML '06: Proceedings of the 23rd international conference on Machine learning (2006), pp. 65-72, doi:10.1145/1143844.1143853 by Maria F. Balcan, Alina Beygelzimer, John Langford

Agnostic active learning Journal of Computer and System Sciences, Vol. 75, No. 1. (2009), pp. 78-89 by M. F. Balcan, A. Beygelzimer, J. Langford

Background

- The algorithm A² (for agnostic active) is the first active learning algorithm that finds an ϵ - optimized hypothesis in any hypothesis class when the underlying distribution as arbitrary forms of noise.
- The only assumption for A² is that it has access to a stream of unlabeled samples which are drawn *i.i.d.* from a fixed distribution.
- Under active learning the algorithm is allowed to draw random samples unlabeled examples and ask for labels of these examples.
- The only prior belief about the learning problem is that the target function, or a reasonable approximation to it, belongs to a given concept class.

Contributions

The main contribution of this paper is to prove the feasibility of agnostic active learning.

A² Algorithm

set $i \leftarrow 1$, $D_i \leftarrow D$, $H_i \leftarrow H$, $H_{i-1} \leftarrow H$, $S_{i-1} \leftarrow 0$, and $k \leftarrow 1$.

(1) **while** $\text{Disagree}_D(H_{i-1}) \left(\min_{h \in H_{i-1}} \text{UB}(S_{i-1}, h, \delta_k) - \min_{h \in H_{i-1}} \text{LB}(S_{i-1}, h, \delta_k) \right) > \epsilon$

set $S_i \leftarrow 0$, $H'_i \leftarrow H_i$, $k \leftarrow k + 1$

(2) **while** $\text{Disagree}_D(H'_i) \geq \frac{1}{2} \text{Disagree}_D(H_i)$

if $\text{Disagree}_D(H_i) (\min_{h \in H_i} \text{UB}(S_i, h, \delta_k) - \min_{h \in H_i} \text{LB}(S_i, h, \delta_k)) \leq \epsilon$

(*) **return** $h = \text{argmin}_{h \in H_i} \text{UB}(S_i, h, \delta_k)$

else $S'_i = \text{rejection sample } 2|S_i| + 1 \text{ samples } x \text{ from } D \text{ satisfying}$

$\exists h_1, h_2 \in H_1 : h_1 \neq h_2(x)$

$S_i \leftarrow S_i \cup \{(x, O(x)) : x \in S'_i\}$, $k \leftarrow k + 1$

(**) $H'_i = \{h \in H_i : \text{LB}(S_i, h, \delta_k) \leq \min'_{h \in H_i} \text{UB}(S_i, h', \delta_k)\}$, $k \leftarrow k + 1$

end if

end while

$H_{i+1} \leftarrow H'_i$, $D_{i+1} \leftarrow D_i$ restricted to $\{x : \exists h_1, h_2 \in H'_i : h_1(x) \neq h_2(x)\}$

$i \leftarrow i + 1$

end while

return $h = \text{argmin}_{h \in H_{i-1}} \text{UB}(S_{i-1}, h, \delta_k)$

A² Algorithm

set $i \leftarrow 1$, $D_i \leftarrow D$, $H_i \leftarrow H$, $H_{i-1} \leftarrow H$, $S_{i-1} \leftarrow 0$, and $k \leftarrow 1$.

(1) **while** $\text{Disagree}_D(H_{i-1}) \left(\min_{h \in H_{i-1}} \text{UB}(S_{i-1}, h, \delta_k) - \min_{h \in H_{i-1}} \text{LB}(S_{i-1}, h, \delta_k) \right) > \epsilon$
 set $S_i \leftarrow 0$, $H'_i \leftarrow H_i$, $k \leftarrow k + 1$
 (2) **while** $\text{Disagree}_D(H'_i) \geq \frac{1}{2} \text{Disagree}_D(H_i)$
 if $\text{Disagree}_D(H_i) (\min_{h \in H_i} \text{UB}(S_i, h, \delta_k) - \min_{h \in H_i} \text{LB}(S_i, h, \delta_k)) \leq \epsilon$
 (*) **return** $h = \text{argmin}_{h \in H_i} \text{UB}(S_i, h, \delta_k)$
 else $S'_i = \text{rejection sample } 2|S_i| + 1 \text{ samples } x \text{ from } D \text{ satisfying}$
 $\exists h_1, h_2 \in H_1 : h_1 \neq h_2(x)$
 $S_i \leftarrow S_i \cup \{(x, O(x)) : x \in S'_i\}$, $k \leftarrow k + 1$
 (**) $H'_i = \{h \in H_i : \text{LB}(S_i, h, \delta_k) \leq \min'_h \text{UB}(S_i, h', \delta_k)\}$, $k \leftarrow k + 1$
 end if
end while
 $H_{i+1} \leftarrow H'_i$, $D_{i+1} \leftarrow D_i \text{ restricted to } \{x : \exists h_1, h_2 \in H'_1 : h_1(x) \neq h_2(x)\}$
 $i \leftarrow i + 1$

end while

return $h = \text{argmin}_{h \in H} \text{UB}(S_{i-1}, h, \delta_k)$

A² Algorithm

set $i \leftarrow 1$, $D_i \leftarrow D$, $H_i \leftarrow H$, $H_{i-1} \leftarrow H$, $S_{i-1} \leftarrow 0$, and $k \leftarrow 1$.

X is an instance space and $Y = \{-1, 1\}$ is the set of possible labels. H is a hypothesis class, which is a set of function mapping from X to Y . We assume there is a distribution D over instances of X and that the instances are labeled by a possibly randomized oracle O .

A² Algorithm

set $i \leftarrow 1$, $D_i \leftarrow D$, $H_i \leftarrow H$, $H_{i-1} \leftarrow H$, $S_{i-1} \leftarrow 0$, and $k \leftarrow 1$.

- $i \leftarrow 1$: i iterates over the number of rounds, this is the number of times that region of uncertainty with respect to D has been reduced by half.
- $D_i \leftarrow D$: D_i is initialized to the entire distribution
- H_i and $H_{i-1} \leftarrow H$: H_i refers the the current version space, which is the set of Hypothesis that are consistent with all labels reveals so far. Both the current and previous class are initialized to the full hypothesis space.
- $S_{i-1} \leftarrow 0$: The previous sample space is initialized to 0.
- $k \leftarrow 1$: k keeps track of the number of bound calculations

A² Algorithm

set $i \leftarrow 1$, $D_i \leftarrow D$, $H_i \leftarrow H$, $H_{i-1} \leftarrow H$, $S_{i-1} \leftarrow 0$, and $k \leftarrow 1$.

(1) while $\text{Disagree}_D(H_{i-1}) \left(\min_{h \in H_{i-1}} \text{UB}(S_{i-1}, h, \delta_k) - \min_{h \in H_{i-1}} \text{LB}(S_{i-1}, h, \delta_k) \right) > \epsilon$

set $S_i \leftarrow 0$, $H'_i \leftarrow H_i$, $k \leftarrow k + 1$

(2) while $\text{Disagree}_D(H'_i) \geq \frac{1}{2} \text{Disagree}_D(H_i)$

if $\text{Disagree}_D(H_i) (\min_{h \in H_i} \text{UB}(S_i, h, \delta_k) - \min_{h \in H_i} \text{LB}(S_i, h, \delta_k)) \leq \epsilon$

(*) return $h = \text{argmin}_{h \in H_i} \text{UB}(S_i, h, \delta_k)$

else $S'_i = \text{rejection sample } 2|S_i| + 1 \text{ samples } x \text{ from } D \text{ satisfying}$

$\exists h_1, h_2 \in H_i : h_1 \neq h_2(x)$

$S_i \leftarrow S_i \cup \{(x, O(x)) : x \in S'_i\}$, $k \leftarrow k + 1$

(**) $H'_i = \{h \in H_i : \text{LB}(S_i, h, \delta_k) \leq \min'_{h \in H_i} \text{UB}(S_i, h', \delta_k)\}$, $k \leftarrow k + 1$

end if

end while

$H_{i+1} \leftarrow H'_i$, $D_{i+1} \leftarrow D_i \text{ restricted to } \{x : \exists h_1, h_2 \in H'_i : h_1(x) \neq h_2(x)\}$

$i \leftarrow i + 1$

end while

return $h = \text{argmin}_{h \in H_i} \text{UB}(S_i, h, \delta_k)$

A² Algorithm

$$(1) \text{ while } \text{Disagree}_D(H_{i-1}) \left(\min_{h \in H_{i-1}} \text{UB}(S_{i-1}, h, \delta_k) - \min_{h \in H_{i-1}} \text{LB}(S_{i-1}, h, \delta_k) \right) > \epsilon$$

$\text{Disagree}_D(H_i)$ is the probability that any there exists a pair of hypotheses in H_i that disagrees on a random example drawn from D .

$$\text{Disagree}_D(H_i) = \Pr_{x \sim D} [\exists h_1, h_2 \in H_i : h_1(x) \neq h_2(x)]$$

- $\text{Disagree}_D(H_i)$ is the volume of the current region of uncertainty with respect to D .
- Can be estimated to any desired precision with probability 1 using an unlabeled dataset with size limited to infinity.

A² Algorithm

$$(1) \text{ while } \text{Disagree}_D(H_{i-1}) \left(\min_{h \in H_{i-1}} \text{UB}(S_{i-1}, h, \delta_k) - \min_{h \in H_{i-1}} \text{LB}(S_{i-1}, h, \delta_k) \right) > \epsilon$$

A² relies on a subroutine which computes a lower bound $\text{LB}(S, h, \delta)$ and an upper bound $\text{UB}(S, h, \delta)$ on the true error rate $\text{err}_P(h)$ of h using sample S of examples drawn *i.i.d.* from P . Each of these bounds must hold for all h simultaneously with probability at least $1 - \delta$.

Definition 1. A subroutine for computing LB and UB is said to be *legal* if for all distributions P over $X \times Y$ for all $0 < \delta < 1/2$ and $m \in \mathbb{N}$.

$$\text{LB}(S, h, \delta) \leq \text{err}_P(h) \leq \text{UB}(S, h, \delta)$$

holds for all $h \in H$ simultaneously with probability $1 - \delta$ over the draw of S according to P^m

A² Algorithm

$$(1) \text{ while } \text{Disagree}_D(H_{i-1}) \left(\min_{h \in H_{i-1}} \text{UB}(S_{i-1}, h, \delta_k) - \min_{h \in H_{i-1}} \text{LB}(S_{i-1}, h, \delta_k) \right) > \epsilon$$

Definition 1. A subroutine for computing LB and UB is said to be *legal* if for all distributions P over $X \times Y$ for all $0 < \delta < 1/2$ and $m \in \mathbb{N}$.

$$\text{LB}(S, h, \delta) \leq \text{err}_P(h) \leq \text{UB}(S, h, \delta)$$

holds for all $h \in H$ simultaneously with probability $1 - \delta$ over the draw of S according to P^m

Classic example of such subroutines are the (distribution independent) VC bound, the Occam Razar bound or the newer data dependent generalization bounds, for example those based on Rademacher Complexities.

A² Algorithm

(1) **while** $\text{Disagree}_D(H_{i-1}) \left(\min_{h \in H_{i-1}} \text{UB}(S_{i-1}, h, \delta_k) - \min_{h \in H_{i-1}} \text{LB}(S_{i-1}, h, \delta_k) \right) > \epsilon$

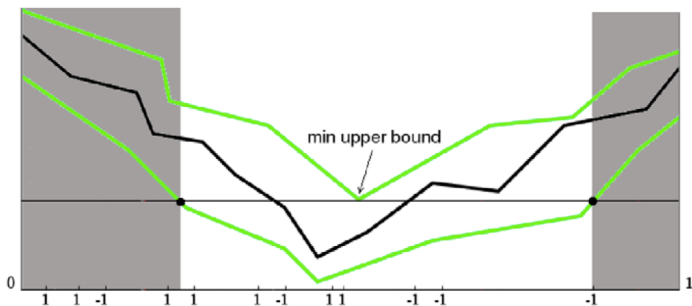


Fig. 3.1. A² in action: sampling, bounding, eliminating.

A² Algorithm

set $i \leftarrow 1$, $D_i \leftarrow D$, $H_i \leftarrow H$, $H_{i-1} \leftarrow H$, $S_{i-1} \leftarrow 0$, and $k \leftarrow 1$.

(1) **while** $\text{Disagree}_D(H_{i-1}) \left(\min_{h \in H_{i-1}} \text{UB}(S_{i-1}, h, \delta_k) - \min_{h \in H_{i-1}} \text{LB}(S_{i-1}, h, \delta_k) \right) > \epsilon$

set $S_i \leftarrow 0$, $H'_i \leftarrow H_i$, $k \leftarrow k + 1$

(2) **while** $\text{Disagree}_D(H'_i) \geq \frac{1}{2} \text{Disagree}_D(H_i)$

if $\text{Disagree}_D(H_i) (\min_{h \in H_i} \text{UB}(S_i, h, \delta_k) - \min_{h \in H_i} \text{LB}(S_i, h, \delta_k)) \leq \epsilon$

(*) **return** $h = \text{argmin}_{h \in H_i} \text{UB}(S_i, h, \delta_k)$

else $S'_i =$ rejection sample $2|S_i| + 1$ samples x from D satisfying

$\exists h_1, h_2 \in H_i : h_1 \neq h_2(x)$

$S_i \leftarrow S_i \cup \{(x, O(x)) : x \in S'_i\}$, $k \leftarrow k + 1$

(**) $H'_i = \{h \in H_i : \text{LB}(S_i, h, \delta_k) \leq \min'_{h \in H_i} \text{UB}(S_i, h', \delta_k)\}$, $k \leftarrow k + 1$

end if

end while

$H_{i+1} \leftarrow H'_i$, $D_{i+1} \leftarrow D_i$ restricted to $\{x : \exists h_1, h_2 \in H'_i : h_1(x) \neq h_2(x)\}$

$i \leftarrow i + 1$

end while

A² Algorithm

set $S_i \leftarrow 0$, $H'_i \leftarrow H_i$, $k \leftarrow k + 1$

- Initialize the sample S_i with 0
- Copy H_i into H'_i
- Advance the index k
- δ follows a schedule dependent on k , such that $\delta_k = \frac{\delta}{k(k+1)}$

A² Algorithm

set $i \leftarrow 1$, $D_i \leftarrow D$, $H_i \leftarrow H$, $H_{i-1} \leftarrow H$, $S_{i-1} \leftarrow 0$, and $k \leftarrow 1$.

(1) **while** $\text{Disagree}_D(H_{i-1}) \left(\min_{h \in H_{i-1}} \text{UB}(S_{i-1}, h, \delta_k) - \min_{h \in H_{i-1}} \text{LB}(S_{i-1}, h, \delta_k) \right) > \epsilon$
 set $S_i \leftarrow 0$, $H'_i \leftarrow H_i$, $k \leftarrow k + 1$

(2) **while** $\text{Disagree}_D(H'_i) \geq \frac{1}{2} \text{Disagree}_D(H_i)$

if $\text{Disagree}_D(H_i) (\min_{h \in H_i} \text{UB}(S_i, h, \delta_k) - \min_{h \in H_i} \text{LB}(S_i, h, \delta_k)) \leq \epsilon$

(*) **return** $h = \text{argmin}_{h \in H_i} \text{UB}(S_i, h, \delta_k)$

else $S'_i = \text{rejection sample } 2|S| + 1 \text{ samples } x \text{ from } D \text{ satisfying}$

$\exists h_1, h_2 \in H_1 : h_1 \neq h_2(x)$

$S_i \leftarrow S_i \cup \{(x, O(x)) : x \in S'_i\}$, $k \leftarrow k + 1$

(**) $H'_i = \{h \in H_i : \text{LB}(S_i, h, \delta_k) \leq \min'_{h \in H_i} \text{UB}(S_i, h', \delta_k)\}$, $k \leftarrow k + 1$

end if

end while

$H_{i+1} \leftarrow H'_i$, $D_{i+1} \leftarrow D_i$ restricted to $\{x : \exists h_1, h_2 \in H'_1 : h_1(x) \neq h_2(x)\}$

$i \leftarrow i + 1$

end while

return $h = \text{argmin}_{h \in H_i} \text{UB}(S_i, h, \delta_k)$

A² Algorithm

(2) **while** $\text{Disagree}_D(H'_i) \geq \frac{1}{2} \text{Disagree}_D(H_i)$

- Execute this loop until H'_i is found such that for a point randomly drawn from D the probability that H'_i contains two hypothesis that are in disagreement is less than $1/2$ that of the original H_i .
- This occurs when S_i has grown large enough to enough to eliminate a least half of the current region of uncertainty.
- Since the probability $\text{Disagree}_d(H_i)$ is reduced by $\frac{1}{2}$ on every iteration, the number of iterations cannot be more than $\log\left(\frac{1}{\epsilon}\right)$

A² Algorithm

set $i \leftarrow 1$, $D_i \leftarrow D$, $H_i \leftarrow H$, $H_{i-1} \leftarrow H$, $S_{i-1} \leftarrow 0$, and $k \leftarrow 1$.

(1) **while** $\text{Disagree}_D(H_{i-1}) \left(\min_{h \in H_{i-1}} \text{UB}(S_{i-1}, h, \delta_k) - \min_{h \in H_{i-1}} \text{LB}(S_{i-1}, h, \delta_k) \right) > \epsilon$

set $S_i \leftarrow 0$, $H'_i \leftarrow H_i$, $k \leftarrow k + 1$

(2) **while** $\text{Disagree}_D(H'_i) \geq \frac{1}{2} \text{Disagree}_D(H_i)$

if $\text{Disagree}_D(H_i) (\min_{h \in H_i} \text{UB}(S_i, h, \delta_k) - \min_{h \in H_i} \text{LB}(S_i, h, \delta_k)) \leq \epsilon$

(*) **return** $h = \text{argmin}_{h \in H_i} \text{UB}(S_i, h, \delta_k)$

else $S'_i = \text{rejection sample } 2|S| + 1 \text{ samples } x \text{ from } D \text{ satisfying}$

$\exists h_1, h_2 \in H_1 : h_1 \neq h_2(x)$

$S_i \leftarrow S_i \cup \{(x, O(x)) : x \in S'_i\}$, $k \leftarrow k + 1$

(**) $H'_i = \{h \in H_i : \text{LB}(S_i, h, \delta_k) \leq \min'_h \text{UB}(S_i, h', \delta_k)\}$, $k \leftarrow k + 1$

end if

end while

$H_{i+1} \leftarrow H'_i$, $D_{i+1} \leftarrow D_i \text{ restricted to } \{x : \exists h_1, h_2 \in H'_1 : h_1(x) \neq h_2(x)\}$

$i \leftarrow i + 1$

end while

return $h = \text{argmin}_{h \in H_i} \text{UB}(S_i, h, \delta_k)$

A² Algorithm

```
if DisagreeD (Hi) (minh ∈ HiUB (Si, h, δk) − minh ∈ HiLB (Si, h, δk)) ≤ ε  
(*) return h = argminh ∈ HiUB (Si, h, δk)
```

- If H_i represents a hypothesis class that is within the error bound for sample S_i , return the first hypothesis under the upper bound.

A² Algorithm

set $i \leftarrow 1$, $D_i \leftarrow D$, $H_i \leftarrow H$, $H_{i-1} \leftarrow H$, $S_{i-1} \leftarrow 0$, and $k \leftarrow 1$.

(1) **while** $\text{Disagree}_D(H_{i-1}) \left(\min_{h \in H_{i-1}} \text{UB}(S_{i-1}, h, \delta_k) - \min_{h \in H_{i-1}} \text{LB}(S_{i-1}, h, \delta_k) \right) > \epsilon$

set $S_i \leftarrow 0$, $H'_i \leftarrow H_i$, $k \leftarrow k + 1$

(2) **while** $\text{Disagree}_D(H'_i) \geq \frac{1}{2} \text{Disagree}_D(H_i)$

if $\text{Disagree}_D(H_i) (\min_{h \in H_i} \text{UB}(S_i, h, \delta_k) - \min_{h \in H_i} \text{LB}(S_i, h, \delta_k)) \leq \epsilon$

(*) **return** $h = \text{argmin}_{h \in H_i} \text{UB}(S_i, h, \delta_k)$

else $S'_i = \text{rejection sample } 2|S_i| + 1 \text{ samples } x \text{ from } D \text{ satisfying}$

$\exists h_1, h_2 \in H_1 : h_1 \neq h_2(x)$

$S_i \leftarrow S_i \cup \{(x, O(x)) : x \in S'_i\}$, $k \leftarrow k + 1$

(**) $H'_i = \{h \in H_i : \text{LB}(S_i, h, \delta_k) \leq \min'_{h \in H_i} \text{UB}(S_i, h', \delta_k)\}$, $k \leftarrow k + 1$

end if

end while

$H_{i+1} \leftarrow H'_i$, $D_{i+1} \leftarrow D_i$ restricted to $\{x : \exists h_1, h_2 \in H'_1 : h_1(x) \neq h_2(x)\}$

$i \leftarrow i + 1$

end while

return $h = \text{argmin}_{h \in H} \text{UB}(S_{i-1}, h, \delta_k)$

A² Algorithm

else $S'_i =$ rejection sample $2|S_i| + 1$ samples x from D satisfying
 $\exists h_1, h_2 \in H_1 : h_1 \neq h_2(x)$

- Draw additional samples from the distribution for which two hypotheses still under consideration do not agree.

A² Algorithm

set $i \leftarrow 1$, $D_i \leftarrow D$, $H_i \leftarrow H$, $H_{i-1} \leftarrow H$, $S_{i-1} \leftarrow 0$, and $k \leftarrow 1$.

(1) **while** $\text{Disagree}_D(H_{i-1}) \left(\min_{h \in H_{i-1}} \text{UB}(S_{i-1}, h, \delta_k) - \min_{h \in H_{i-1}} \text{LB}(S_{i-1}, h, \delta_k) \right) > \epsilon$

set $S_i \leftarrow 0$, $H'_i \leftarrow H_i$, $k \leftarrow k + 1$

(2) **while** $\text{Disagree}_D(H'_i) \geq \frac{1}{2} \text{Disagree}_D(H_i)$

if $\text{Disagree}_D(H_i) (\min_{h \in H_i} \text{UB}(S_i, h, \delta_k) - \min_{h \in H_i} \text{LB}(S_i, h, \delta_k)) \leq \epsilon$

(*) **return** $h = \text{argmin}_{h \in H_i} \text{UB}(S_i, h, \delta_k)$

else $S'_i = \text{rejection sample } 2|S| + 1 \text{ samples } x \text{ from } D \text{ satisfying}$

$\exists h_1, h_2 \in H_1 : h_1 \neq h_2(x)$

$S_i \leftarrow S_i \cup \{(x, O(x)) : x \in S'_i\}$, $k \leftarrow k + 1$

(**) $H'_i = \{h \in H_i : \text{LB}(S_i, h, \delta_k) \leq \min'_{h \in H_i} \text{UB}(S_i, h', \delta_k)\}$, $k \leftarrow k + 1$

end if

end while

$H_{i+1} \leftarrow H'_i$, $D_{i+1} \leftarrow D_i \text{ restricted to } \{x : \exists h_1, h_2 \in H'_1 : h_1(x) \neq h_2(x)\}$

$i \leftarrow i + 1$

end while

return $h = \text{argmin}_{h \in H_i} \text{UB}(S_{i-1}, h, \delta_k)$

A² Algorithm

$$S_i \leftarrow S_i \cup \{(x, O(x)) : x \in S'_i\}, k \leftarrow k + 1$$

- New labels are requested from the Oracle for some of the data points where the surviving hypotheses disagree.
- Labels are not given to data points where all hypotheses agree. This means that an optimal surviving hypothesis on H_i remains an optimal hypothesis on H_{i+1} .

A² Algorithm

set $i \leftarrow 1$, $D_i \leftarrow D$, $H_i \leftarrow H$, $H_{i-1} \leftarrow H$, $S_{i-1} \leftarrow 0$, and $k \leftarrow 1$.

(1) **while** $\text{Disagree}_D(H_{i-1}) \left(\min_{h \in H_{i-1}} \text{UB}(S_{i-1}, h, \delta_k) - \min_{h \in H_{i-1}} \text{LB}(S_{i-1}, h, \delta_k) \right) > \epsilon$

set $S_i \leftarrow 0$, $H'_i \leftarrow H_i$, $k \leftarrow k + 1$

(2) **while** $\text{Disagree}_D(H'_i) \geq \frac{1}{2} \text{Disagree}_D(H_i)$

if $\text{Disagree}_D(H_i) (\min_{h \in H_i} \text{UB}(S_i, h, \delta_k) - \min_{h \in H_i} \text{LB}(S_i, h, \delta_k)) \leq \epsilon$

(*) **return** $h = \text{argmin}_{h \in H_i} \text{UB}(S_i, h, \delta_k)$

else $S'_i = \text{rejection sample } 2|S| + 1 \text{ samples } x \text{ from } D \text{ satisfying}$

$\exists h_1, h_2 \in H_1 : h_1 \neq h_2(x)$

$S_i \leftarrow S_i \cup \{(x, O(x)) : x \in S'_i\}$, $k \leftarrow k + 1$

(**) $H'_i = \{h \in H_i : \text{LB}(S_i, h, \delta_k) \leq \min_{h' \in H_i} \text{UB}(S_i, h', \delta_k)\}$, $k \leftarrow k + 1$

end if

end while

$H_{i+1} \leftarrow H'_i$, $D_{i+1} \leftarrow D_i \text{ restricted to } \{x : \exists h_1, h_2 \in H'_i : h_1(x) \neq h_2(x)\}$

$i \leftarrow i + 1$

end while

return $h = \text{argmin}_{h \in H_{i-1}} \text{UB}(S_{i-1}, h, \delta_k)$

A² Algorithm

$$(**) H'_i = \{h \in H_i : \text{LB}(S_i, h, \delta_k) \leq \min'_{h' \in H_i} \text{UB}(S_i, h', \delta_k)\}, k \leftarrow k + 1$$

- Update the set of hypotheses included in H_i based on the new sample set, S'_i and new δ_k

A² Algorithm

set $i \leftarrow 1$, $D_i \leftarrow D$, $H_i \leftarrow H$, $H_{i-1} \leftarrow H$, $S_{i-1} \leftarrow 0$, and $k \leftarrow 1$.

(1) **while** $\text{Disagree}_D(H_{i-1}) \left(\min_{h \in H_{i-1}} \text{UB}(S_{i-1}, h, \delta_k) - \min_{h \in H_{i-1}} \text{LB}(S_{i-1}, h, \delta_k) \right) > \epsilon$

set $S_i \leftarrow 0$, $H'_i \leftarrow H_i$, $k \leftarrow k + 1$

(2) **while** $\text{Disagree}_D(H'_i) \geq \frac{1}{2} \text{Disagree}_D(H_i)$

if $\text{Disagree}_D(H_i) (\min_{h \in H_i} \text{UB}(S_i, h, \delta_k) - \min_{h \in H_i} \text{LB}(S_i, h, \delta_k)) \leq \epsilon$

(*) **return** $h = \text{argmin}_{h \in H_i} \text{UB}(S_i, h, \delta_k)$

else $S'_i = \text{rejection sample } 2|S| + 1 \text{ samples } x \text{ from } D \text{ satisfying}$

$\exists h_1, h_2 \in H_1 : h_1 \neq h_2(x)$

$S_i \leftarrow S_i \cup \{(x, O(x)) : x \in S'_i\}$, $k \leftarrow k + 1$

(**) $H'_i = \{h \in H_i : \text{LB}(S_i, h, \delta_k) \leq \min'_{h \in H_i} \text{UB}(S_i, h', \delta_k)\}$, $k \leftarrow k + 1$

end if

end while

$H_{i+1} \leftarrow H'_i$, $D_{i+1} \leftarrow D_i$ restricted to $\{x : \exists h_1, h_2 \in H'_1 : h_1(x) \neq h_2(x)\}$

$i \leftarrow i + 1$

end while

return $h = \text{argmin}_{h \in H_{i-1}} \text{UB}(S_{i-1}, h, \delta_k)$

A² Algorithm

$$H_{i+1} \leftarrow H'_i, D_{i+1} \leftarrow D_i \text{ restricted to } \{x : \exists h_1, h_2 \in H'_1 : h_1(x) \neq h_2(x)\}$$

$$i \leftarrow i + 1$$

- Set H_i to the new hypothesis set from the previous iteration.
- Restrict the Distribution D to the area under which there are still hypotheses which do not agree.
- Increment i to track the current number of rounds.

3.1 Correctness

Theorem 3.1 *For all H , for all (D, O) , for all valid subroutines for computing UB and LB , for all $0 < \epsilon < 1/2$ and $0 < \delta < 1/2$, with probability $1 - \delta$, A^2 returns an ϵ -optimal hypothesis or does not terminate.*

- This theorem makes two claims:
 - all bound evaluations are valid simultaneously with probability at least $1 - \delta$
 - the procedure produces an ϵ -optimal hypothesis upon termination
- The first proof is based on the fact that the k th bound evaluation fails by, at most, probability $\frac{\delta}{k(k+1)}$, which means the union is bounded by the sum, $\sum_{k=1}^{\text{infy}} \frac{\delta}{k(k+1)}$
- The second part of the theorem is based on the observation that in order for the algorithm to terminate it must meet the condition
$$\text{Disagree}_D(H_i) \left(\min_{h \in H_i} UB(S_i, h, \delta_k) - \min_{h \in H_i} LB(S_i, h, \delta_k) \right) \leq \epsilon$$

3.2 Fall back

Theorem 3.2 *For all H , for all (D, O) , for all UB and LB satisfying the assumption $m(2\epsilon, \delta, H) \leq \frac{m(\epsilon, \delta, H)}{2}$, for all $0 < \epsilon < 1/2$ and $0 < \delta < 1/2$, the algorithm **A²** makes at most $2m(\epsilon, \delta', H)$ calls to the oracle **O**, where $\delta' = \frac{\delta}{N(\epsilon, \delta, H)(N(\epsilon, \delta, H)+1)}$ and $N(\epsilon, \delta, H)$ satisfies*

$$N(\epsilon, \delta', H) \geq \ln \frac{1}{\epsilon^2} \ln m\left(\epsilon, \frac{\delta}{N(\epsilon, \delta, H)(N(\epsilon, \delta, H)+1)}, H\right)$$

. Here $m(\epsilon, \delta, H)$ is the sample complexity of UB and LB

- $N(\epsilon, \delta, H)$ is the number of bound evaluations throughout the life of the algorithm.
- The number of rounds is bounded by $\log_2 \frac{1}{\epsilon}$
- This implies that the maximum number of rounds throughout the life of the algorithm is $\log_2 \frac{1}{\epsilon} \log_2 m(\epsilon, \delta', H)$
- Adding up the number of calls to the oracle, **O**, in all rounds gives at most

Corollary 3.3

Corollary 3.3 *For all hypothesis classes H of VC-dimension V_H , for all distributions (D, O) over $X \times Y$, for all $0 < \epsilon < 1/2$ and $0 < \delta < 1/2$, the algorithm A^2 requires at most $\tilde{O}\left(\frac{1}{\epsilon^2} \left(V_H \ln \frac{1}{\epsilon} + \ln \frac{1}{\delta}\right)\right)$ labeled examples from the oracle O .*

- $m(\epsilon, \delta, H)$ and Theorem 3.2 implies an upper bound on $N = N(\epsilon, \delta, H)$

4.1 Exponential Improvements

Theorem 4.1 *Let H be the set of thresholds on an interval. For all distributions (D, O) where D is a continuous probability distribution function, for any $\epsilon < \frac{1}{2}$ and $\frac{\epsilon}{16} \geq \eta$, the algorithm A^2 makes*

$$O\left(\ln\left(\frac{1}{\epsilon} \ln\left(\frac{\ln \frac{1}{\epsilon\delta}}{\delta}\right)\right)\right)$$

calls to the oracle O on examples drawn i.i.d. from D , with probability $1 - \delta$.

- An exponential improvement in sample complexity is shown when the noise rate is small

4.2 Effect of Large Noise

Theorem 4.2 *Let H be the set of thresholds on an interval. Suppose that $\epsilon < \frac{1}{2}$ and $\eta > 16\epsilon$. For all D , with probability $1 - \delta$, the algorithm A^2 requires at most $\tilde{O}\left(\frac{\eta^2 \ln \frac{1}{\delta}}{\epsilon^2}\right)$ labeled samples*

- A slower improvement is shown when noise rate, η is large. In the extreme case when η is $\frac{1}{2}$, there is no improvement.

4.4 Linear Separators under Uniform Distribution

Theorem 4.4 This reflects the exponential saving given by active learning in the number of labeled examples when the data is drawn uniformly from the unit sphere in \mathbb{R}^d and labels are consistent with a linear separator going through the origin. A² provides exponential savings even in the presence of arbitrary forms of noise.

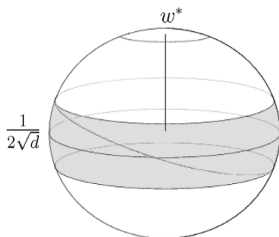


Fig. 4.1. The region of uncertainty after the first iteration (schematic).

- A² provides the first justification of why one can hope to achieve strong results (similar to Perceptron-based active learner) in the harder agnostic learning case, when the noise rate is sufficiently small with respect to the desired error.

4.4 Linear Separators under Uniform Distribution

Theorem 4.4 Let $X = \{x \in \mathbb{R}^d : \|x\| = 1\}$ (a unit sphere), D is uniform over X , let H be the class of linear separators through the origin and LB and UB be the VC bound. Then for any $0 < \epsilon < \frac{1}{2}$, $0 < \eta < \frac{\epsilon}{16\sqrt{d}}$, and $\delta > 0$, with probability $1 - \delta$, A² requires

$$O\left(d \left(d \ln d + \ln \frac{1}{\delta'}\right) \ln \frac{1}{\epsilon}\right)$$

calls to the labeling oracle, where $\delta' = \frac{\delta}{N(\epsilon, \delta, H)(N(\epsilon, \delta, H) + 1)}$ and

$$N(\epsilon, \delta, H) = O\left(\ln \frac{1}{\epsilon} \left(d^2 \ln d + d \ln \frac{d \ln \frac{1}{\epsilon}}{\delta}\right)\right)$$

Open Questions

- A^2 is, like other selective sampling algorithms is non-aggressive in it's choice of query points. Using an aggressive querying strategy has been shown to produce better bounds under certain conditions. Designing an aggressive agnostic active learning algorithm remains an open problem.
- Generally the conditions sufficient and necessary for active learning in the agnostic case have not been derived. Another open question would be to derive and analyze the optimal agnostic active learning strategy.
- Most of the current active learning is focused on the binary classification problem. Future work could involve implementing other loss functions.



LARGE-SCALE LIVE ACTIVE LEARNING: TRAINING OBJECT DETECTORS WITH CRAWLED DATA AND CROWDS

Sudheendra Vijayanarasimhan and Kristen Grauman

Mo Zhou, Electrical Engineering Dept., Columbia University

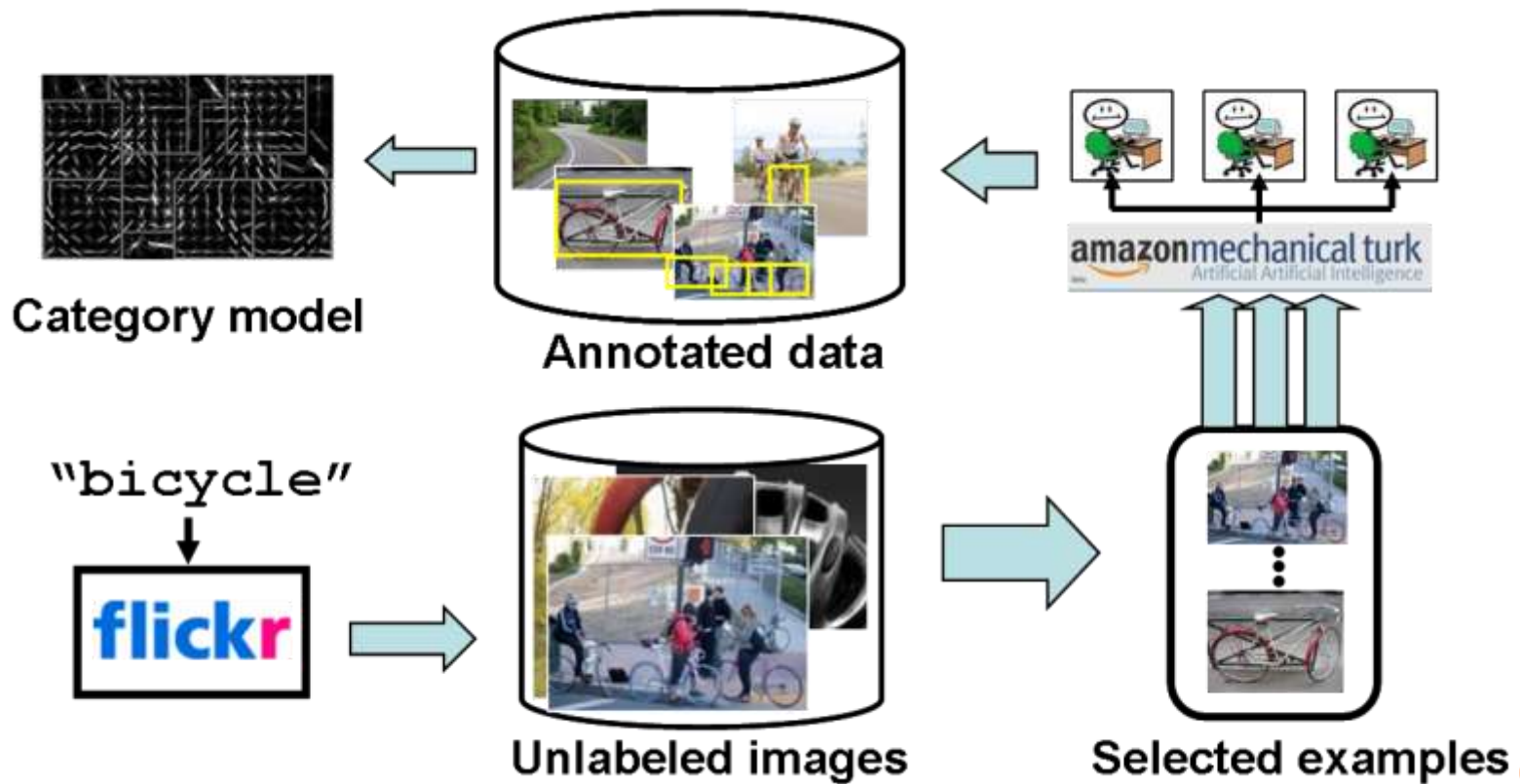
BACKGROUND

- Supervised learning, semi-supervised learning, unsupervised learning - **active learning**
- In object recognition, active learning can be used to build up training sets efficiently
- Only 'sandbox' data are currently tested
- Most crowd-sourced collection processes need human fine-tuning repeatedly



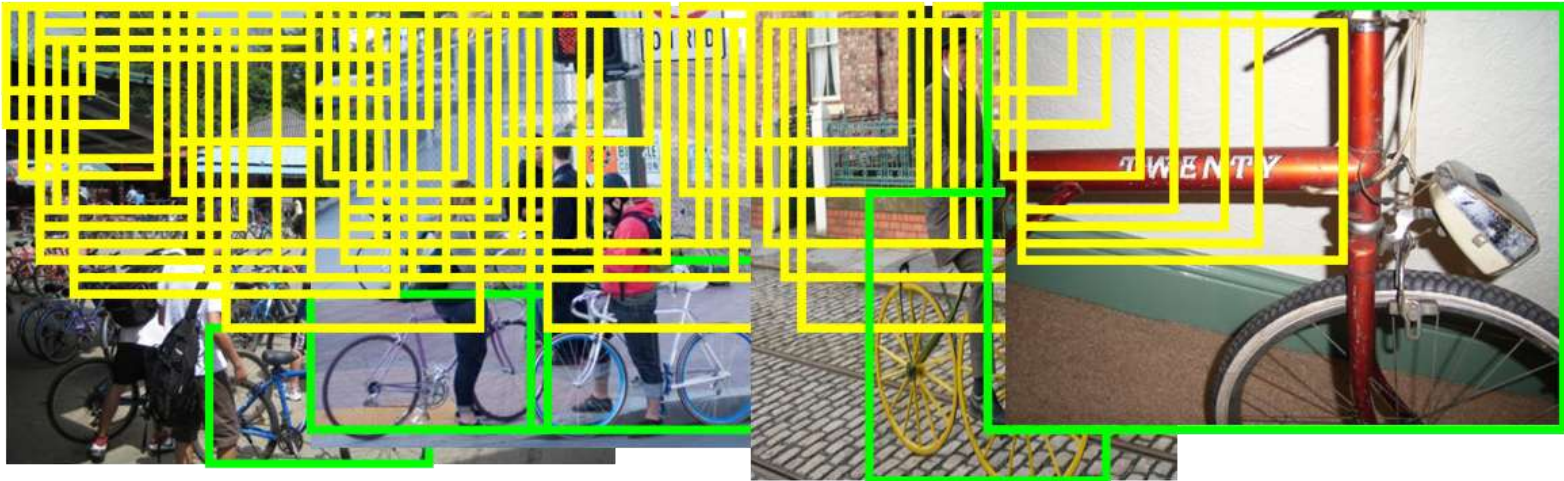
GOAL

- Test on uncompromised datasets and make the whole process automatic



CHALLENGE

- Large number of candidate windows in each image and most are useless
- 1000 windows/image results to millions examples

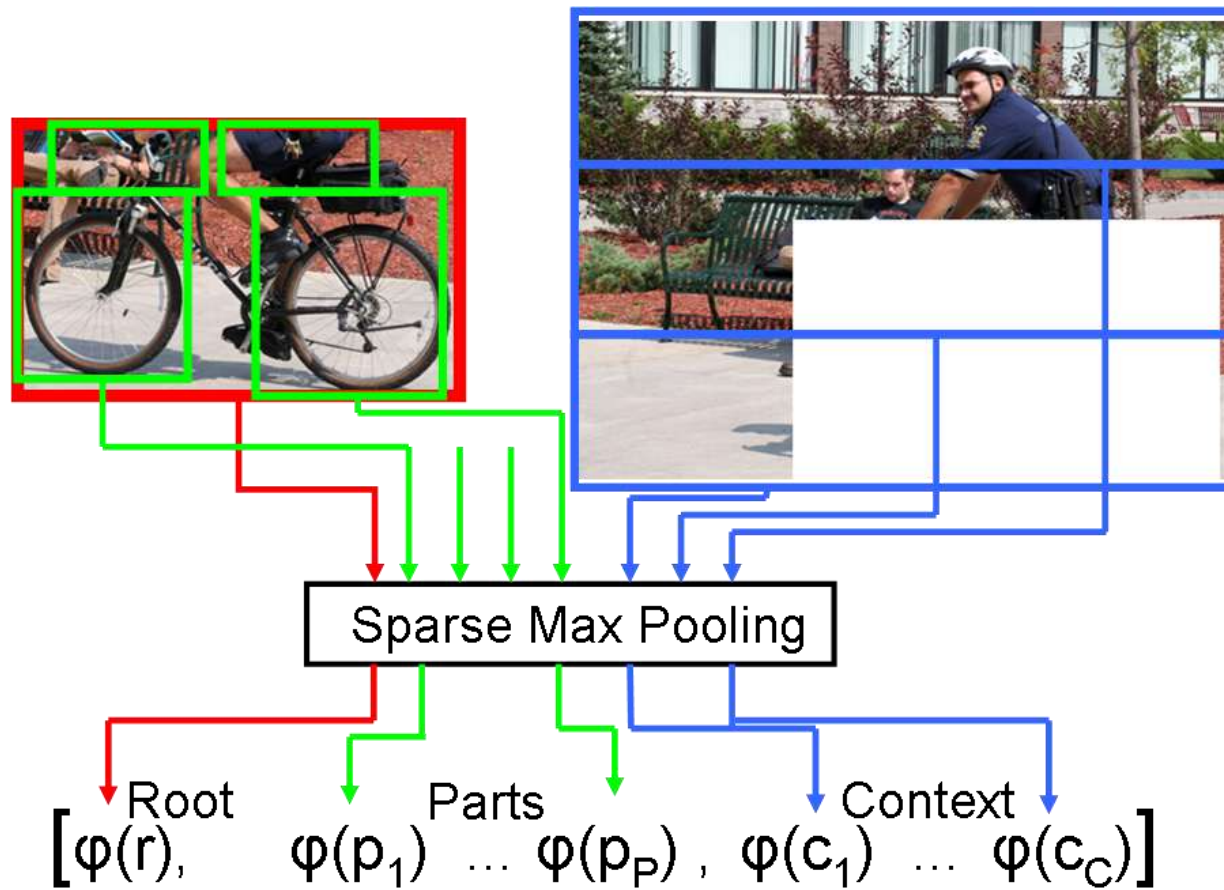


APPROACH: LINEAR CLASSIFICATION

- Part-based detector amenable to linear SVM
- Given root window r , multiple part windows p_1, p_2, \dots, p_m that overlap the root, and context windows c_1, c_2, \dots, c_n surrounding an object
- Root window: global summary of object
- Part window: local feature summary
- Context window: incorporate contextual cues, i.e. sky, ground, road, etc.
- Concatenate max-pooled responses from a sparse coding of the features within each window

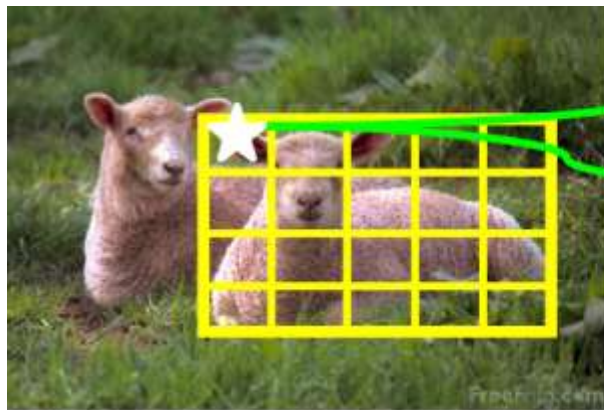


APPROACH: LINEAR CLASSIFICATION

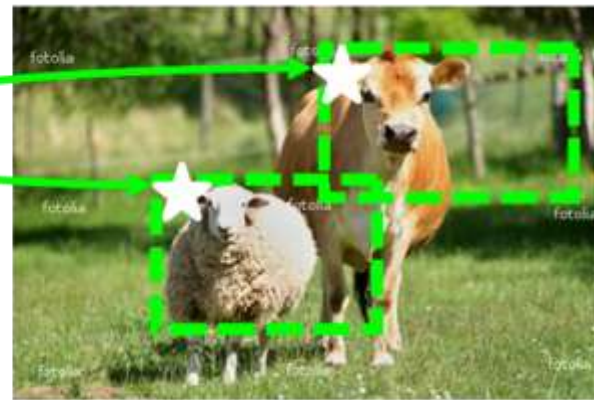


APPROACH: WINDOW GENERATION

- Use a grid-based, Hough-like projections to generate windows from unlabeled images
- Divide a training window into $N \times M$ grid and record visual word, grid location, bounding box
- Rank according to frequency of occurrence and use the top 3000 boxes from each image



Training image

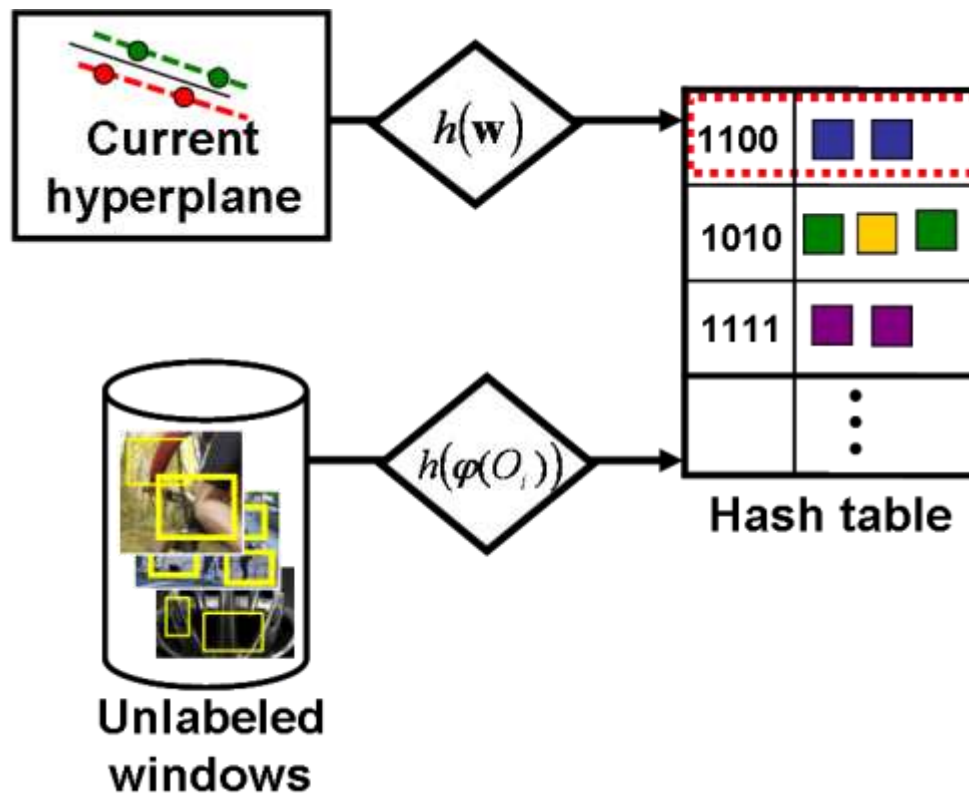


Novel test image




APPROACH: ACTIVE SELECTION

- Initialize system with a trained linear SVM
- Put all generated unlabeled windows into a hash-table using hyperplane hash function
- Hash detector directly to the bin during selection



APPROACH: ANNOTATION COLLECTION

- Post selected images on Mechanical Turk
- Provide multiple options to avoid incorrect boxes
- Post same image to 5-10 annotators for consensus



(q) ☐ box₂ bicycle (normal)

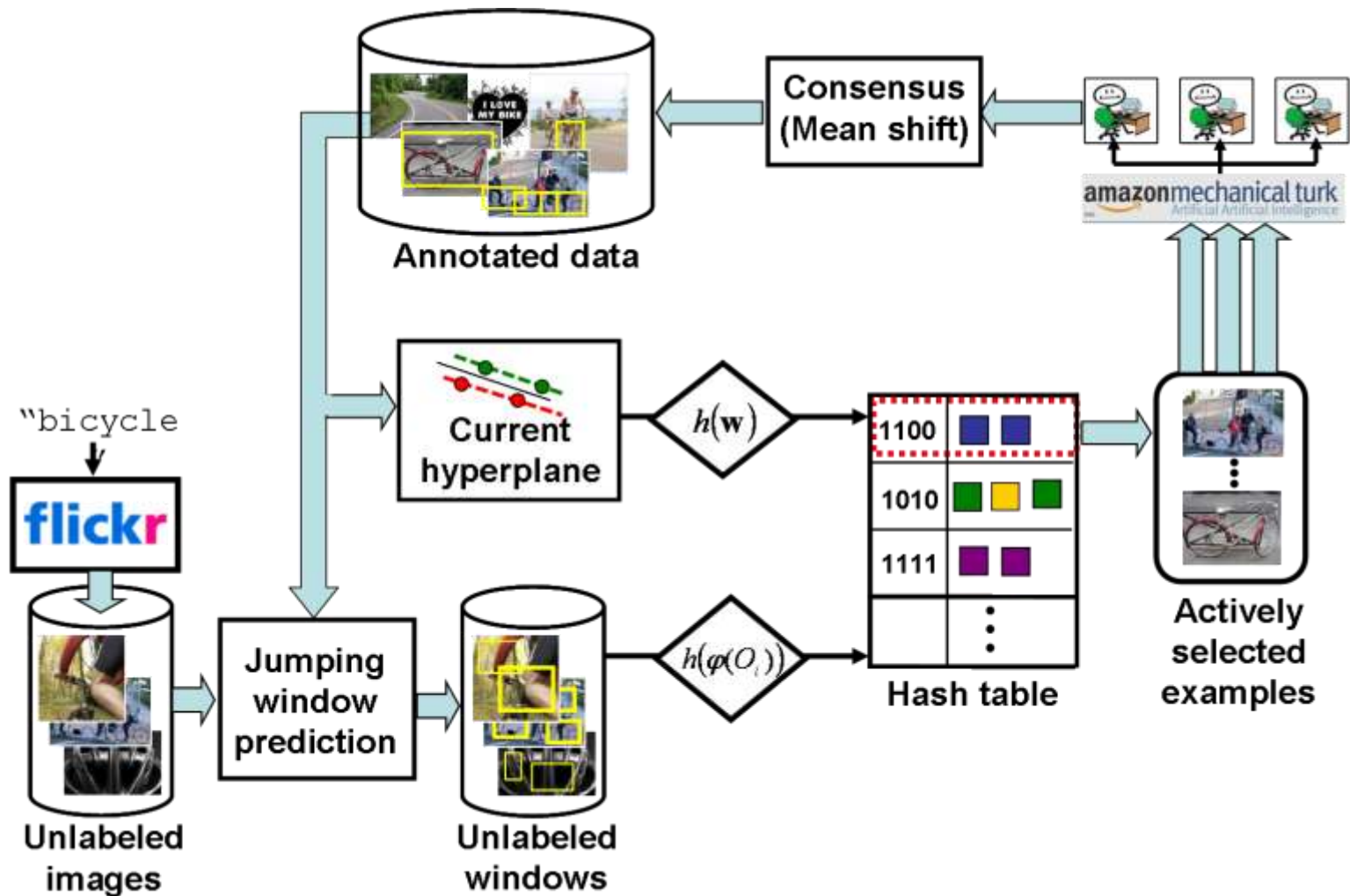
(w) ☐ box₂ bicycle (truncated)

(e) ☐ box₂ bicycle (unusual)

(r) ☐ There are no bicycles

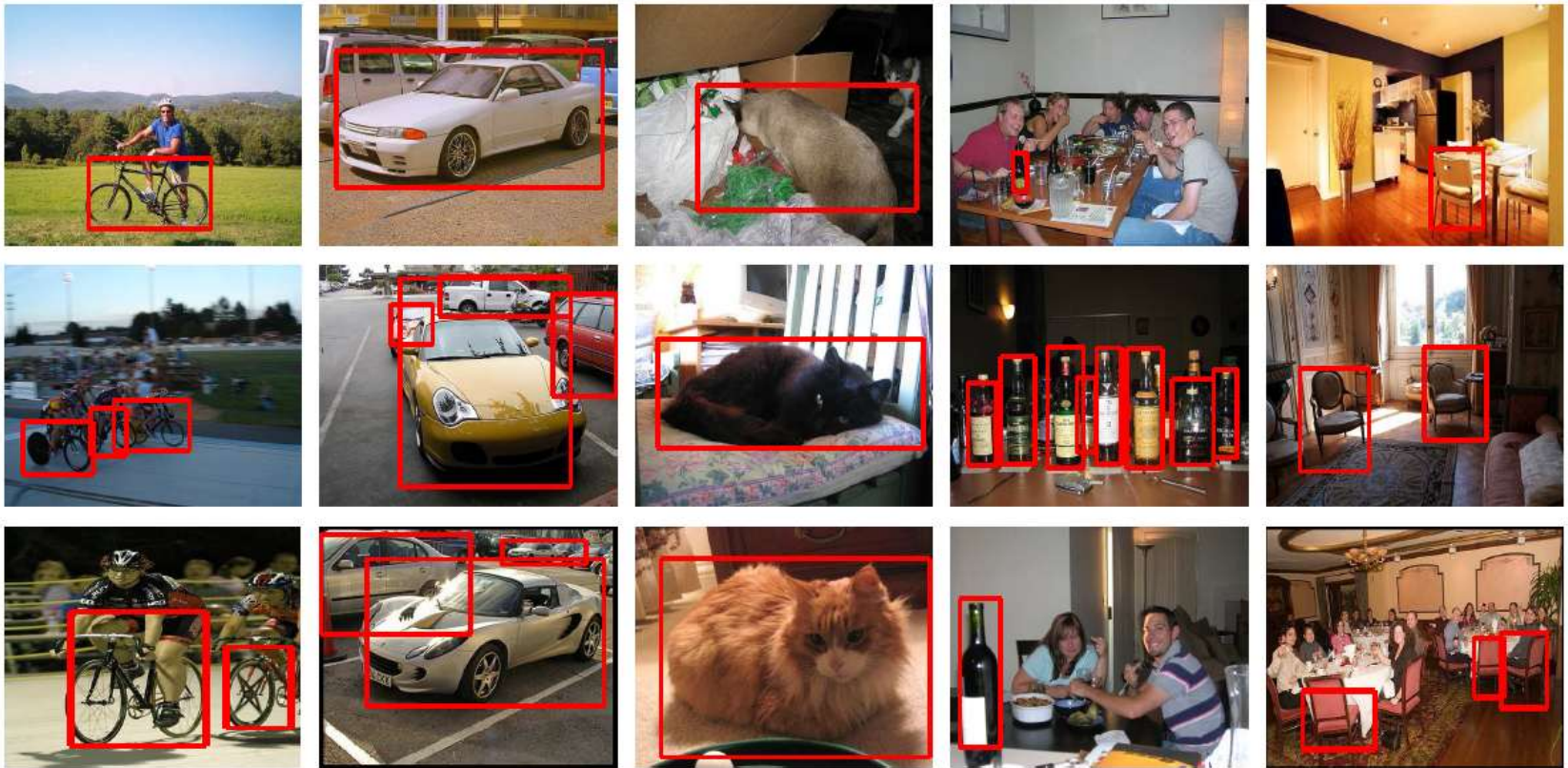
(t) ☐ There are more than 3 bicycles.

APPROACH: GLOBAL VIEW



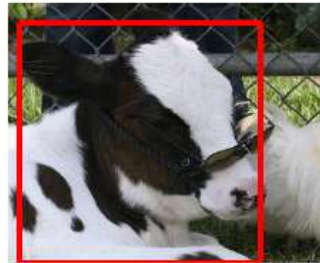
RESULTS: EXAMPLES

- True positives:

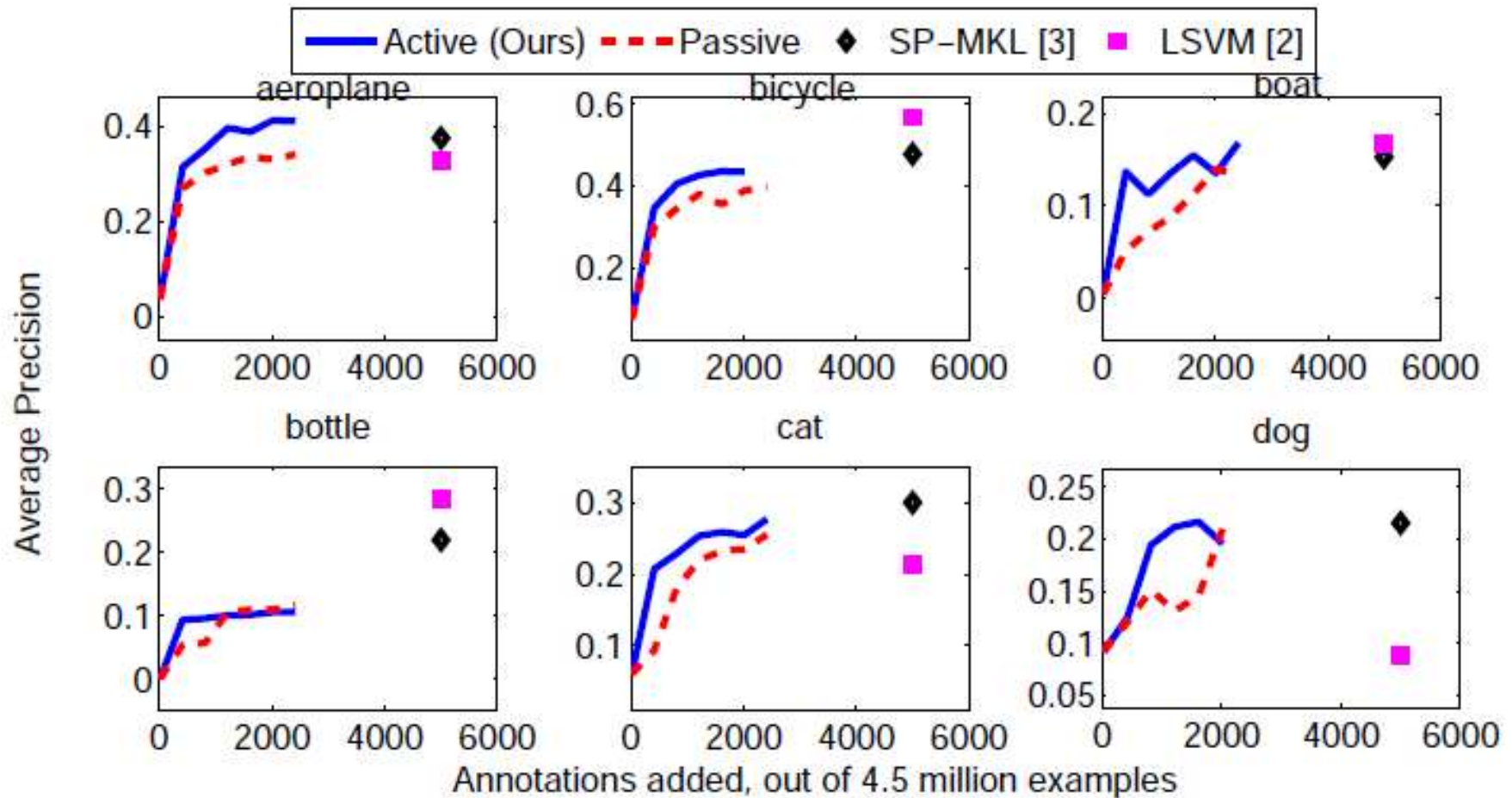


RESULTS: EXAMPLES

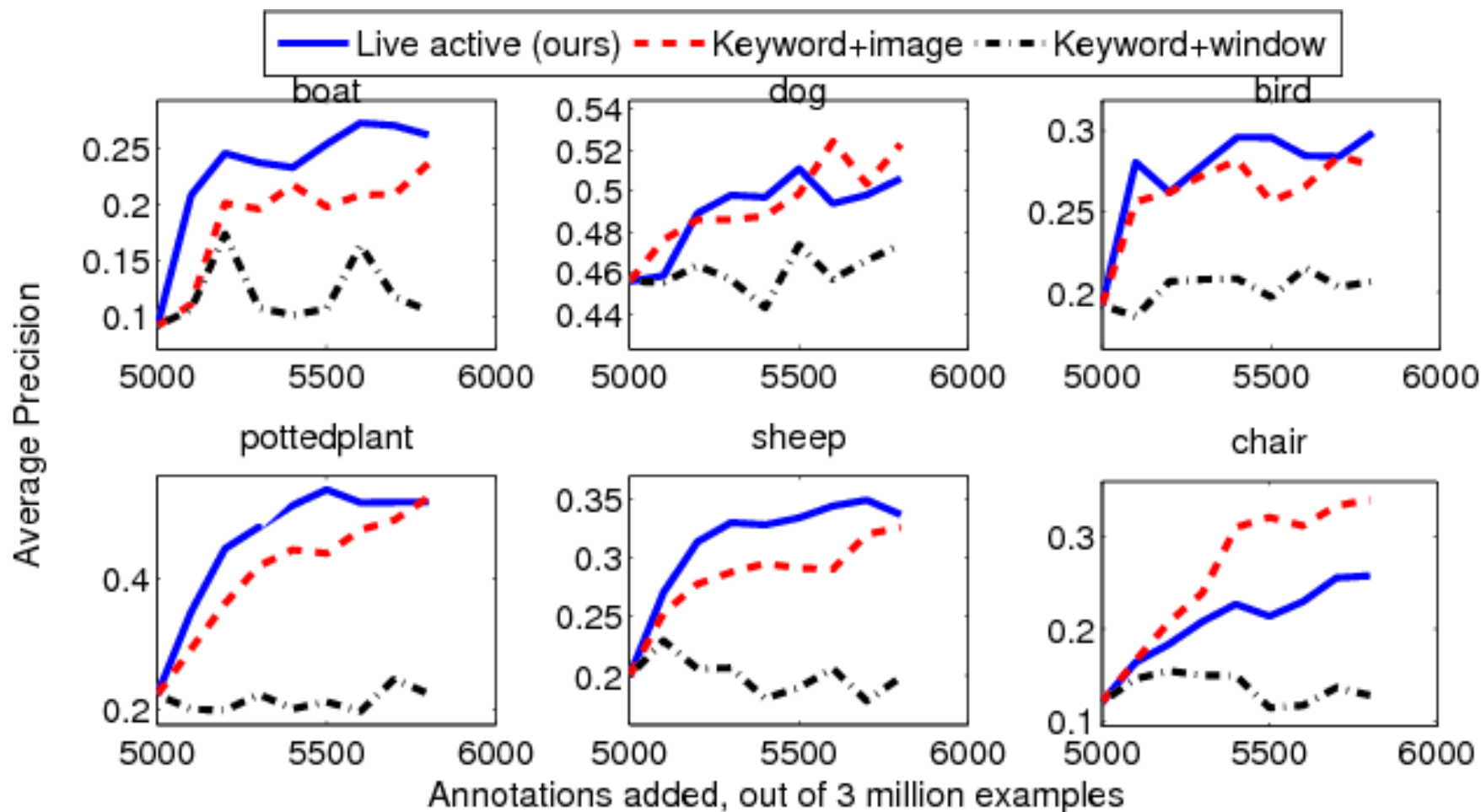
- False positives:



RESULTS: ACCURACY ON PASCAL



RESULTS: ACCURACY ON FLICKR



RESULTS: COMPUTATION TIME

- Time mostly spent on pooling sparse codes
- Efficiency in selecting useful images and retraining the classifier makes live learning practical

	Active Selection	Training	Detection/image
[1] + active	10 mins	5 mins	150 secs
[1] + passive	0 mins	5 mins	150 secs
LSVM [2]	3 hrs	4 hrs	2 secs
SP + MKL [3]	93 hrs	> 2 d	67 secs



CONCLUSION

- A novel efficient part-based linear detector that delivers excellent performance
- A jumping window and hashing scheme suitable for the proposed detector that retrieves relevant instances among millions of candidates
- The first active learning results for which both image data and annotations are automatically obtained, with minimal involvement from vision experts
- Overall: An effective end-to-end system for online learning of object detectors



REFERENCES

- [1] Large-Scale Live Active Learning: Training Object Detectors with Crawled Data and Crowds, S. Vijayanarasimhan, K. Grauman, CVPR 2011
- [2] Object Detection with Discriminatively Trained Part Based Models, P. Felzenszwalb, R. Girshick, D. McAllester, D. Ramanan, TPAMI, 99(1), 2009
- [3] Multiple Kernels for Object Detection, A. Vedaldi, V. Gulshan, M. Varma, A. Zisserman, ICCV 2009
- [4] Large-Scale Live Active Learning: Training Object Detectors with Crawled Data and Crowds, <http://vision.cs.utexas.edu/projects/livelearning/>

