Cross-Domain Video Concept Detection Using Adaptive SVMs

AUTHORS: JUN YANG, RONG YAN, ALEXANDER G. HAUPTMANN
PRESENTATION: JESSE DAVIS
CS 3710 VISUAL RECOGNITION

Problem-Idea-Challenges

- Address accuracy mismatch in training/test data
  - Use A-SVMs and Classifier Selection Techniques
- Identify and Resolve classifier adaptation problems:
  - How to transform old classifiers into usable classifiers for new datasets
  - How to select best candidate classifier to be adapted

Figure 1. Mismatch between two different domains.
(**Left:** images of *computer monitor* from online catalog. **Right:** images of *computer monitor* captured from an office. *Images credit [5].*)

Relevance and Related Approaches

- Classifier Adaptation is important in several communities
  - Visual Recognition - Cross Domain Video Concept Detection
  - Data Mining - Drifting Concept Detection
  - Machine Learning - Transfer Learning and Incremental Learning
- A-SVM advances can promote ease of integration of works from other papers
  - e.g. Paper A can utilize SVMs from Paper B and Paper C with the help of Adaptive SVMs

This Paper's Approach

- Use A-SVMs to adapt one (or many) classifiers to the target dataset
  - Learn the delta function
  - Use delta function to "adapt" the SVM to target data
- Estimate performance of classifiers
  - Analyze their score distributions, etc.
  - Select "best" performers

Outline

- A-SVMs
  - SVMs
  - One-to-one vs. Many-to-one
  - Learning Algorithm
- Auxiliary Classifier Selection
  - Score Distribution and Score Aggregation
  - Predicting Performances
- Alternative Adaptation Methods
  - Aggregate vs. Ensemble
- Cross-Domain Video Concept Detection
  - Task -> Collection -> Adaptation

Adaptive Support Vector Machines

- Goal
  - Learn a classifier to correctly classify objects in primary dataset
- Idea
  - We have several existing SVM classifiers from various sources
  - We want to create an SVM that identifies classes on a new domain
  - Adapt the existing classifiers to our new target classifiers to utilize SVMs that have been trained on different sources for robustness/accuracy

Standard SVMs

- (1)
  - We want to train a standard SVM for $D_l^p = \{(x_i, y_i)\}_{i=1}^N$ where x_i is the i^{th} data vector (in the small, labeled subset of the primary dataset) and y_i is its binary label
  - Seeking decision boundary with small classification error for the trade off of a large marginalization

Scalar cost factor

Measure of the total classification error

Slack variable (degree of misclassification for our x)

Regularization term; inversely related to margin between training examples of two classes

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}\xi_i \qquad (1)$$

$$\text{s.t.} \quad \xi_i \geq 0, \quad y_i \mathbf{w}^T \phi(\mathbf{x}_i) \geq 1 - \xi_i, \quad \forall (\mathbf{x}_i, y_i) \in \mathcal{D}_l^p$$

One-to-one Adaptation

- (2)
  - We want to create a new "A-SVM" ($f(x)$) using $f^a(x)$ which was trained using the auxiliary data
  - We do this by adding the "delta function" mentioned early to the auxiliary classifier

Auxiliary classifier

Model's parameters (To be estimated from the labeled examples in D_l^p)

Data vector x mapped to feature vector Φ

$$f(\mathbf{x}) = f^a(\mathbf{x}) + \Delta f(\mathbf{x}) = f^a(\mathbf{x}) + \mathbf{w}^T \phi(\mathbf{x}) \qquad (2)$$

One-to-one Adaptation…

- (3)
  - Similarly to (1), the meaning for the classification error remains the same while $||w||^2$ here is the set of linear parameters of $\Delta f(x)$ as opposed to $f(x)$
  - The regularizer desires a minimal change (Δ) which in turn favors a decision function that is close to our auxiliary classifier
  - Large C = small influence; Small C = big influence; If good auxiliary => use small C

Different! Based on $\Delta f(x)$

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}\xi_i \qquad (3)$$

$$\text{s.t.} \quad \xi_i \geq 0$$

$$y_i f^a(\mathbf{x}_i) + y_i \mathbf{w}^T \phi(\mathbf{x}_i) \geq 1 - \xi_i, \quad \forall(\mathbf{x}_i, y_i) \in \mathcal{D}_l^p$$

One-to-one Adaptation…

- (9)
  - This is the equation for our adapted classifier; can be considered an enhanced version of our auxiliary classifier with support vectors from D_l^p

Lagrangian multiplier

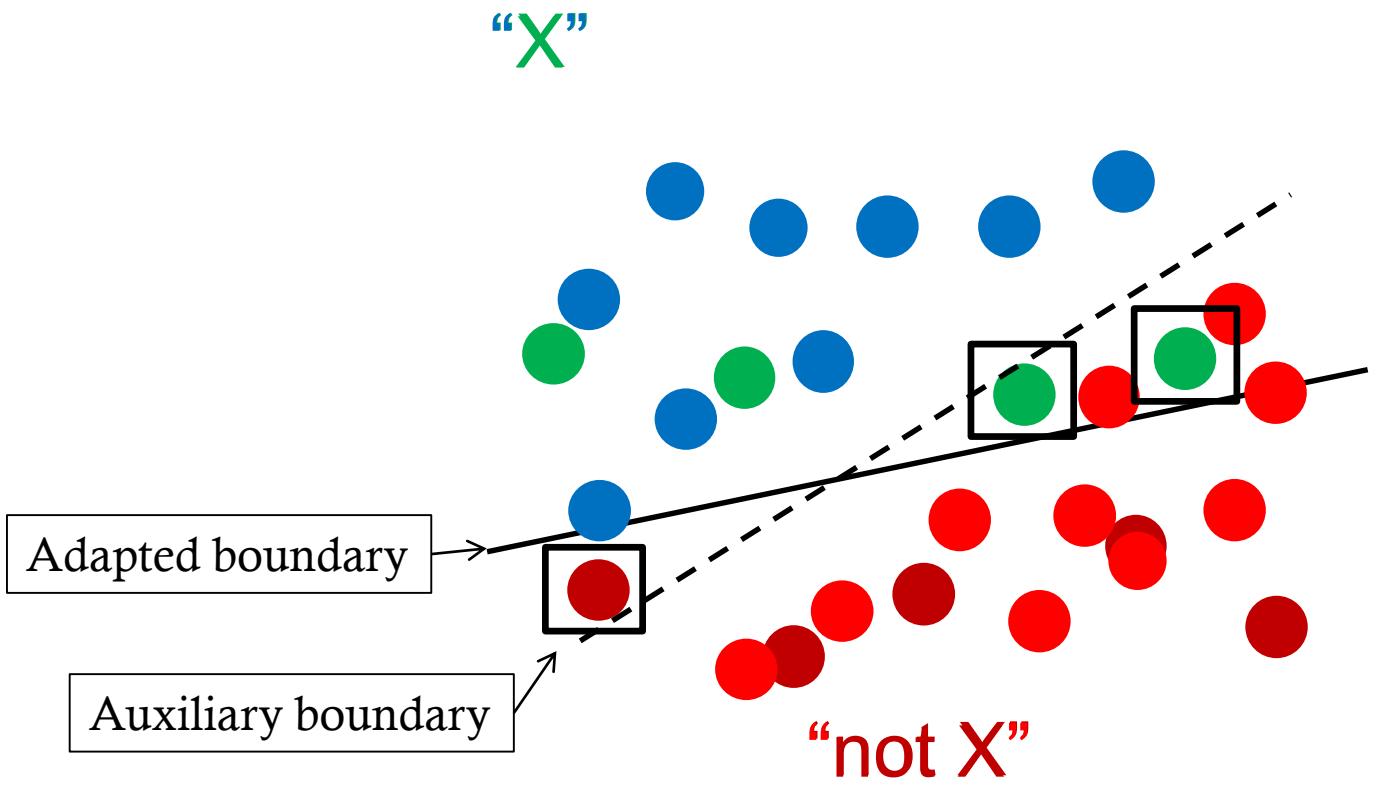
$$f(\mathbf{x}) = f^a(\mathbf{x}) + \sum_{i=1}^{N} \hat{\alpha}_i y_i K(\mathbf{x}, \mathbf{x}_i) \qquad (9)$$

The kernel function which determines the form of the decision boundary; calculated by using a feature map to project each data vector into a feature vector

Note: The same RBF kernel function is used in all methods in the experiment

e.g. $K(x_i, x_j) = e^{-\rho\left||x_i - x_j|\right|^2}$ with $\rho = 0.1$

Learning Adapted Attributes



Many-to-one Adaptation

- (10)
  - Idea is to incorporate several auxiliary classifiers to produce a new classifier using the methods mentioned in the one-to-one adaptation
- (11)
  - Same idea as (3) except $f^a(x)$ becomes: $\sum_{k=1}^{M} t_k f_k^a(x)$

t_k: (0,1) the weight of each auxiliary classifier $f_k^a(x)$

$$f(\mathbf{x}) = \sum_{k=1}^{M} t_k f_k^a(\mathbf{x}) + \Delta f(\mathbf{x}) = \sum_{k=1}^{M} t_k f_k^a(\mathbf{x}) + \mathbf{w}^T \phi(\mathbf{x}) \quad (10)$$

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N} \xi_i \quad (11)$$

$$\text{s.t.} \quad \xi_i \geq 0, \quad y_i \sum_{k=1}^{M} t_k f_k^a(\mathbf{x}_i) + y_i \mathbf{w}^T \phi(\mathbf{x}_i) \geq 1 - \xi_i$$

Many-to-one Adaptation…

- (13)
  - Again, similar to the equation from the one-to-one adaptation except we do the same replacement that we did in 11 ($f^a(x)$ becomes $\sum_{k=1}^{M} t_k f_k^a(x)$)
  - We now have the equation for our adapted classifier using many-to-one

$$f(\mathbf{x}) = \sum_{k=1}^{M} t_k f_k^a(\mathbf{x}) + \sum_{i=1}^{N} \hat{\alpha}_i y_i K(\mathbf{x}, \mathbf{x}_i) \qquad (13)$$

Outline

- A-SVMs
  - SVMs
  - One-to-one vs. Many-to-one
  - Learning Algorithm
- Auxiliary Classifier Selection
  - Score Distribution and Score Aggregation
  - Predicting Performances
- Alternative Adaptation Methods
  - Aggregate vs. Ensemble
- Cross-Domain Video Concept Detection
  - Task -> Collection -> Adaptation

Auxiliary Classifier Selection

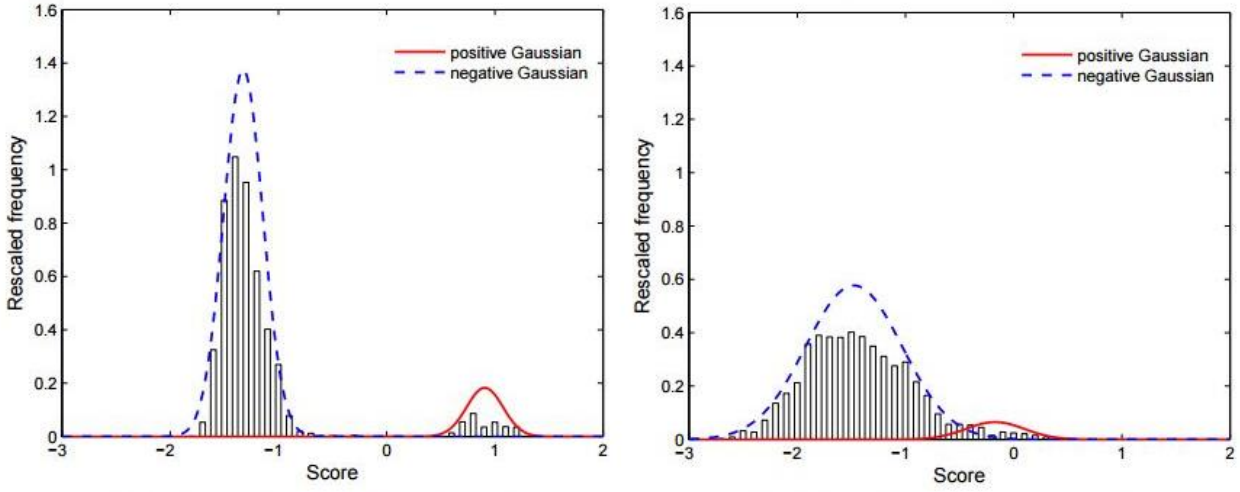
- Goal
  - Select the best classifier such that the one created does better than the one it is derived from with respect to the primary dataset
- Problems
  - Difficult to compute the “best” classifier
  - i.e. How do we gauge the performance without running on the primary dataset? (costly!)
- Solution
  - Utilize meta-data features to gauge performance (can be done without data labels!)

Selection by Score Distribution

- Classifier produces score based on likelihood of positive/negative instance
  - e.g. scores of positive instances should be separated from scores of negatives instances
- Problem
  - Difficult to examine the score separation because instance labels from the primary data are often unknown

Selection by Score Distribution…

- Solution
  - Assume scores of (+) and (-) data follow distributions
  - Recover the distributions using Expectation Maximization
  - Use two Gaussian distributions to fit the scores of both instances
  - EM algorithm iteratively improves the model parameters until it finds two Gaussian distributions that best fit the scores



(a) Classifier A (AP=0.972) (b) Classifier B (AP=0.124)

Figure 3: The score distribution of two classifiers on the same dataset. The histograms show the actual score distributions, and the Gaussian curves are fit by EM. Classifier A performs better than B.

Selection by Score Aggregation

- Idea
  - The average of multiple classifiers will tell us more than any individual one
    1) Aggregate output of these multiple classifiers
    2) Predict the labels of the primary data
    3) Use pseudo labels to evaluate individual classifiers
- Implementation
  - Compute the posterior distribution (18)
  - Evaluate individual classifiers by measuring **agreement between output and estimate posterior probability**
    - Convert posteriors into pseudo labels and then compute a performance metric (i.e. Average Precision) based on these labels

Prediction of Classifier Performance

- We now have:
  - Meta level features based on score distribution
  - Meta level features based on score aggregation
- To predict a classifiers performance we:
  - Build a regression model
  - Trained using SVR
  - Input: Our computed meta level features
  - Output: Classifier's performance on primary data
  - We select our classifier based on (highest) AP due to its common use in video concept detection

Outline

- A-SVMs
  - SVMs
  - One-to-one vs. Many-to-one
  - Learning Algorithm
- Auxiliary Classifier Selection
  - Score Distribution and Score Aggregation
  - Predicting Performances
- Alternative Adaptation Methods
  - Aggregate vs. Ensemble
- Cross-Domain Video Concept Detection
  - Task -> Collection -> Adaptation

Alternative Adaptation Methods

- Aggregate Approach
  - Trains a single SVM using all labeled examples in all auxiliary datasets AND the primary dataset (19)
  - Computationally expensive
  - Involves using the Auxiliary data (vs. just the classifiers)

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}\xi_i + \sum_{k=1}^{M} C_k^a \sum_{i=1}^{N}\xi_i^k \qquad (19)$$

$$\text{s.t. } \xi_i \geq 0, \quad \xi_i^k \geq 0$$

$$y_i \mathbf{w}^T \phi(\mathbf{x}_i) \geq 1 - \xi_i, \quad \forall (\mathbf{x}_i, y_i) \in \mathcal{D}_l^p$$

$$y_i^k \mathbf{w}^T \phi(\mathbf{x}_i^k) \geq 1 - \xi_i^k, \quad \forall (\mathbf{x}_i^k, y_i^k) \in \mathcal{D}_k^a, \forall k$$

Alternative Adaptation Methods…

- Ensemble Approach
  - Combines output of classifiers trained separately on their respective datasets
  - Final score is calculated using (20) which is similar to (10)
  - Important difference:
    - A-SVMs use the delta function which can provide additional information with few labeled examples
    - In the ensemble approach, the primary classifier is trained independently from the auxiliary classifiers

$$f^{ens}(\mathbf{x}) = Cf^{p}(\mathbf{x}) + \sum_{k=1}^{M} C_k^a f_k^a(\mathbf{x}) \qquad (20)$$

$$f(\mathbf{x}) = \sum_{k=1}^{M} t_k f_k^a(\mathbf{x}) + \Delta f(\mathbf{x}) = \sum_{k=1}^{M} t_k f_k^a(\mathbf{x}) + \mathbf{w}^T \phi(\mathbf{x}) \qquad (10)$$

Outline

- A-SVMs
  - SVMs
  - One-to-one vs. Many-to-one
  - Learning Algorithm
- Auxiliary Classifier Selection
  - Score Distribution and Score Aggregation
  - Predicting Performances
- Alternative Adaptation Methods
  - Aggregate vs. Ensemble
- Cross-Domain Video Concept Detection (Experiments)
  - Collection -> Adaptation

Collection/Organization

- TREC Video Retrieval Evaluation 2005 (TRECVID)
  - 86 hours of footage; 74,523 video shots
  - All shots annotated (with binary) using 39 semantic concepts (e.g. outdoor scene, indoor scene, news genre, etc.)
  - 13 news programs, 6 channels (thus a suitable candidate for Cross-Domain concept detection)
  - 1 of the 39 concepts is chosen as a target concept and 1 of the 13 programs is chosen as a target program (with only 384 settings that qualified under their terms of relevancy)

Strategies - Experiments

- Adaptation strategies are necessary to build concept classifiers for the target program when few labeled examples are present
- Setup
  1) Rank all the classifiers trained on other programs by their usefulness with respect to the target program
  2) Select top ranked classifiers (programs) as auxiliary classifiers
  3) Train the classifier for the target program based on some adaptation method
- Note: Methods are specifically tweaked s.t. they are still comparable (i.e. same RBF kernel function, fixed variables when necessary, etc.)

Strategies - Experiments…

1) Selection Criterion
   - Oracle, Random, Prior, Sample, Meta
2) Number of Auxiliary Classifiers
   - Vary the number of selected classifiers from 1-5 to observe the impact it has on classification performance (as shown in figure 6)
3) Adaptation Methods
   - Prim, Aux, Adapt, Aggr, Ensemble

Results (Adaptation Methods)

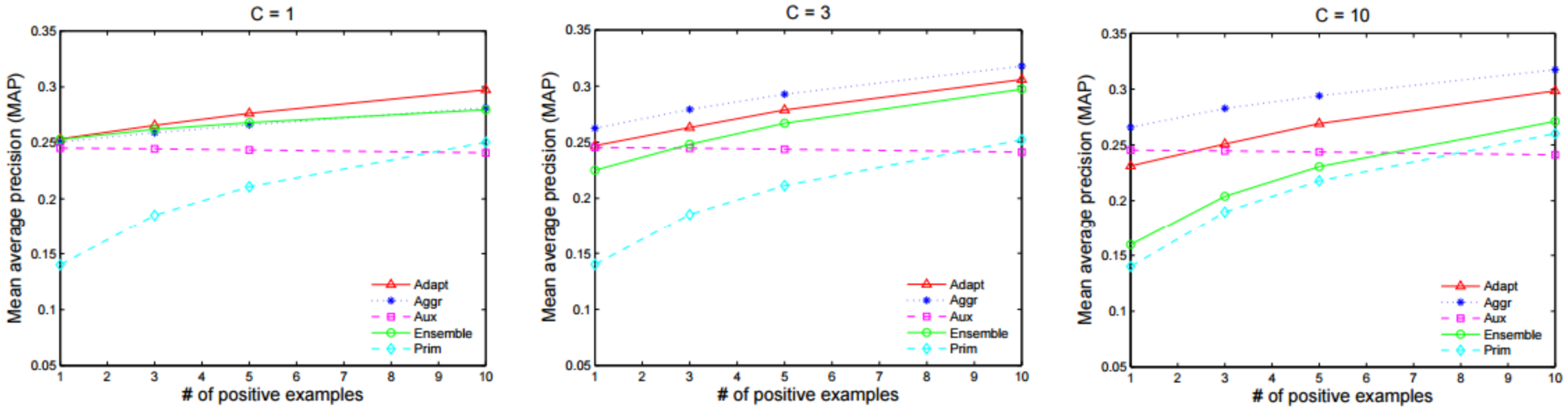


Figure 4: The performance (MAP) of 5 methods on the target program with $C = 1, 3, 10$ and `Prior` selection criterion of auxiliary classifier.

- The Aggregate Method performs best (C > 1) as we increase the weight of C (conversely reducing the weight of the adapted method)

Results (Adaptation Methods)…

# of positive / all examples	Total training time (minutes)		
	Prim/Ensemble	Adapt	Aggr
1 / 16.5	10.3	13.5	191.0
3 / 49.4	10.5	13.8	209.9
5 / 82.4	12.1	14.9	229.0
10 / 164.7	17.4	20.1	271.9

Table 2: The total training time of each method summed from all the settings.

- While we saw that Aggregate performs the best as we increase the examples, so does the training time (in addition to it being the most costly training to begin with)

Results (Auxiliary Classifier Selection)

Metric	MAP of the top-ranked classifier				Ratio of the optimal classifier ranked in top 3			
# of positive examples	1	3	5	10	1	3	5	10
Oracle	0.247	0.247	0.247	0.247	100%	100%	100%	100%
Prior	0.211	0.211	0.211	0.211	68.2%	68.2%	68.2%	68.2%
Meta	0.188	0.201	0.208	0.216	56.6%	60.7 %	68.0%	76.9 %
Sample	0.153	0.186	0.201	0.218	44.6%	57.8%	64.9%	76.2%
Random	0.132	0.132	0.132	0.132	30.1%	30.1%	30.1%	30.1%

Table 3: Comparison of 5 auxiliary classifier selection criteria.

- Metrics are in (in general) descending order of MAP
- MAP only changes (increases) w/r/t # of pos. examples for Meta and Sample

Results (Auxiliary Classifier Selection)…

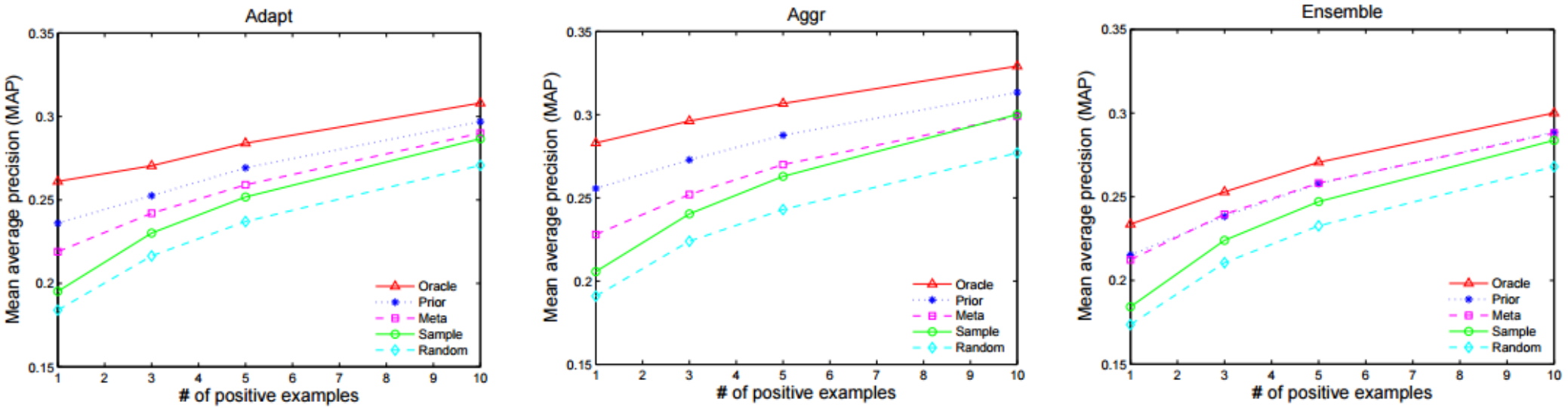


Figure 5: Performance of `Adapt`, `Aggr`, and `Ensemble` using 5 auxiliary classifier selection criteria. Only top-ranked classifier is used as the auxiliary classifier; $C = 3$.

- Oracle performs the best (but as stated is unrealistic), and Prior does the second best
- Note that most of the methods converge as our number of (+) examples increase

Results (Auxiliary Classifier Selection)…

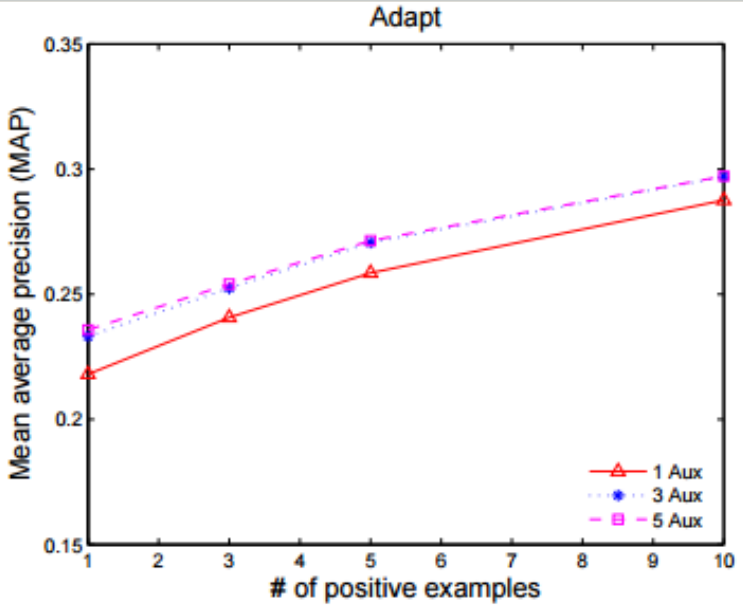


Figure 6: Performance of Adapt with different number of auxiliary classifiers selected by Meta ($C = 3$)

- It appears with respect to the given parameters that increasing the number of auxiliary classifiers past 3 does not increase performance by much (if at all)

Discussion

- Advantages
  - Significantly reduced training time (paper's approach vs. aggregate approach)
  - Competitive accuracy w/r/t the aggregate approach (surpasses ensemble approach)
- Disadvantages
  - Auxiliary classifier selection is critical, if a method fails to select a good one accuracy would presumably plummet
  - Meta data dependent on source (must be reliable)
- Ideas/Future Work
  - Explore different options for auxiliary classifier selection
  - Make C a variable? Base off of…
- Comments

Tabula Rasa: Model Transfer for Object Category Detection

AUTHORS: YUSUF AYTAR, ANDREW ZISSERMAN

Problem and Approach

- Problem
  - Training detectors for a new category is costly
  - Need sufficient data to train positive and negative annotated images
  - Must be done for each desired new category
- Approach/Idea
  - Take a similar pre-existing detector (e.g. using motorcycles to create a detector for bicycles) and use it as a base for learning another class
  - Use transfer learning methods to regularize the training of the new classifier

Example

(a) (b) (c)

Figure 1. **The benefit of transfer learning**. The learnt HOG detector template for a motorbike (a) is used as the source for learning a bicycle template together with the samples shown in (b). The resulting learnt bicycle HOG detector template (c) clearly has the shape of a bicycle. Note, here and in the rest of the paper we only visualize the positive components of the HOG vector.

Model SVM

- We have two categories
  - Target Category – the category we wish to detect (the new category; similar to primary classifier)
  - Source Category – the category which we already have a trained model for (similar to auxiliary classifier)
- Goal is to have an object detector for target category **using knowledge from source category** and available samples of target category
- Three methods of knowledge transfer
  - A-SVM, Project Model Transfer SVM, Deformable Adaptive SVM

Experiments

- Two types
  - Inter-class transfer – transfer from one class to another
    - One-shot learning, Multi-shot learning (MSL), MSL w/ multiple components
  - Specialization – transfer from **superior** class to **subordinate** class (i.e. from a generic class with lots of information to a specific class with detailed/single case information)
- Performed on PASCAL VOC 2007 dataset (Also a small subset dubbed the PASCAL-500)

Experiments…

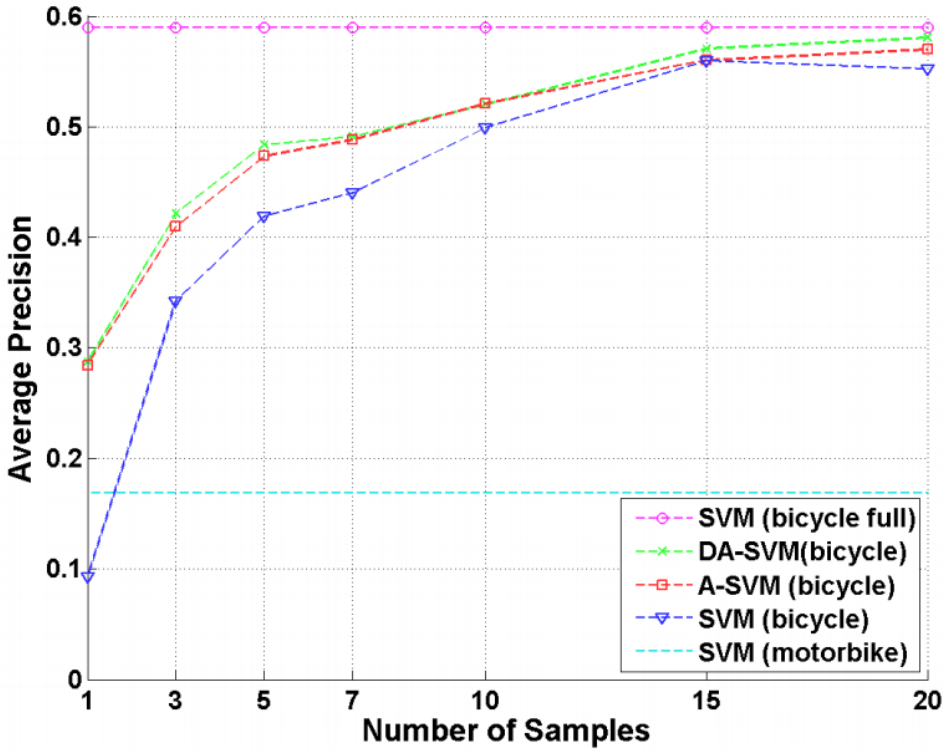
Ranks	Base. SVM	A-SVM	DA-SVM	PMT-SVM
01-15	40.5 ± 07.2	**53.9 ± 04.2**	53.7 ± 04.3	53.5 ± 05.7
16-30	33.0 ± 13.5	52.5 ± 08.3	51.9 ± 08.8	**54.7 ± 05.7**
31-45	26.4 ± 13.3	47.1 ± 07.3	47.1 ± 07.6	**48.5 ± 08.7**
46-60	14.0 ± 09.3	42.4 ± 03.7	**42.5 ± 04.2**	27.8 ± 11.3

Source: motorbike(**44.7%**), **Target:** bicycle(**70.1%**), **Test-set:** PASCAL-500,
Test-procedure: pascal-side-only

In all the tables, test configuration information is given similar to the line above, the values for the source and target are the AP scores of the source(i.e. motorbike) and full target(i.e. bicycle trained with all available samples) detectors on the target task.

Table 2. **Average Precision (AP) comparison of baseline and transfer SVMs on the one shot learning task.** Models are learned using one sample of *bicycle* class and the *motorbike* classifier as the source. The top row displays the average AP results using one of the top 15 (high ranked) samples ranked by the source classifier. Next rows display the next 15 in the ranking. Note the tremendous boost obtained by the transfer method compared to the base SVM (without transfer).

Experiments…



Source: motorbike(**16.9%**), **Target:** bicycle(**59.0%**),
Test-set: PASCAL-COMPLETE, **Test-procedure:** pascal-side-only

Figure 6. AP comparison between baseline SVM and model transfer methods on *bicycle* detection task.

Discussion

- Positives?
  - Better accuracy performance overall
  - Faster learning
  - Base accuracy ≠ 0
- Negatives?
  - Use of only side facing images in training data?
  - Most beneficial when there's a lack of data (increase in performance over typical SVMs degrades with sample increase)
- Extensions?

Resources/References

- http://www.cs.cmu.edu/~juny/Prof/papers/acmmm07jyang.pdf
- http://www.robots.ox.ac.uk/~yusuf/publications/2011/Aytar11/aytar11.pdf
- http://www.cs.cmu.edu/~juny/AdaptSVM/index.html
- http://people.cs.pitt.edu/~kovashka/cs3710_sp15/research.pdf
- http://www-scf.usc.edu/~boqinggo/domainadaptation.html
- http://www.csie.ntu.edu.tw/~cjlin/papers/nusvmtutorial.pdf
- http://www.cs.rit.edu/~rlaz/prec20092/slides/ClassifierSelection.pdf
- http://scikit-learn.org/stable/modules/svm.html
- http://en.wikipedia.org/wiki/Support_vector_machine