Training Data Recycling for Multi-Level Learning

Jingchen Liu[1], Scott McCloskey[2], Yanxi Liu[1]
[1]*Pennsylvania State University,* [2]*Honeywell ACS Labs*
jingchen@cse.psu.edu, scott.mccloskey@honeywell.com, yanxi@cse.psu.edu

Abstract

Among ensemble learning methods, stacking with a meta-level classifier is frequently adopted to fuse the output of multiple base-level classifiers and generate a final score. Labeled data is usually split for base-training and meta-training, so that the meta-level learning is not impacted by over-fitting of base level classifiers on their training data. We propose a novel knowledge-transfer framework that reutilizes the base-training data for learning the meta-level classifier without such negative consequences. By recycling the knowledge obtained during the base-classifier-training stage, we make the most efficient use of all available information and achieve better fusion, thus a better overall performance. With extensive experiments on complicated video event detection, where training data is scarce, we demonstrate the improved performance of our framework over other alternatives.

1. Introduction

"Stacking" is a widely used ensemble method that first trains multiple base-level classifiers and then learns a meta-level classifier with an additional set of training data[12]. The training data for the base-level and meta-level classifiers are also referred to as held-in and held-out data, respectively. Usually each base-classifier generates a continuous (likelihood/confidence) score, which the meta-classifier then fuses to generate a final ranking. This framework has been successfully applied in various detection/ranking systems, *e.g.*, TRECVID[9, 7]) and the Netflix competition[11].

Fundamental to stacking methods is a need to divide the training data wisely, since labels used for base-level training cannot be used for meta-training without reducing performance. This is especially problematic when training data is limited, since subsets of the data may not sufficiently illustrate the underlying semantic concept. Our example is the TRECVID Multimedia Event Detection (MED) dataset[1], with which we detect 5 complicated events from several thousand clips comprising more than 350 hours. Each event is illustrated by about 100 video clips – severely insufficient in light of the broad intra- and inter-class semantic variation (Fig.1).

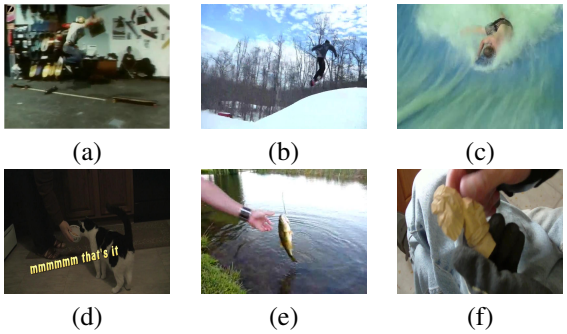

Figure 1. Frames illustrating wide semantic intra-class variation (a-c), with different types of 'board trick', and inter-class variation (d-f) between 'feeding an animal', 'landing a fish', and 'woodworking'.

The motivation of separating base and meta training data is that scores on training data exhibit over-fitting, and thus do not accurately reflect the performance of the classifier on unseen test data. This is illustrated in Fig.2, where the base-classifier's output likelihood score provides better separation of positive and negative labels on the training data (left) than on unseen testing data (right). A better training performance may indicate over-fitting and thus worse generalization. As we demonstrate in Sec. 3, naïvely using training scores to learn the meta classifier often reduces performance, due to this difference in score distribution.

Nevertheless, a more nuanced consideration of scores from training clips can provide information which will improve the fusion model. For example, the correlation among classifiers can be inferred in spite of

[1]http://www.nist.gov/itl/iad/mig/med11.cfm

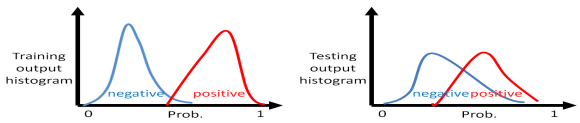

Figure 2. Score distributions: positive (red) and negative (blue) samples on training (left) and testing (right) data.

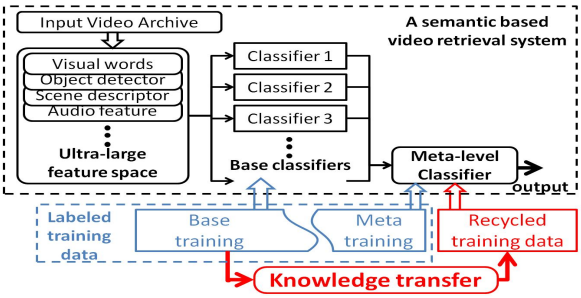

Figure 3. A semantic-based video retrieval system with base-training data recycling.

over-fitting. While cross-validation can provide similar benefits in some cases, generating balanced partitions of the training data is complicated when the numbers of positive examples are very low. We are thus motivated to re-use base-level data for meta-training, which is especially appealing in cases when the amount of labeled data is limited. This approach allows us to use more data for training better base-classifiers with less concern about under-training of the meta classifier. Fig.3 illustrates our training data recycling model, where the entire training data is split for base-training and meta-training, and base-training data is 'recycled' for meta-training.

The idea of re-using the base-training data can be interpreted as a knowledge-transfer process[10], where the scores output by the base-classifiers (a vector X_S of probabilities) on training clips, together with the binary event label y_S, constitutes the source domain $D_S = \{X_S, y_S\}$. The base classifier output scores X_T on unseen (meta-training) data with corresponding labels y_T define the target domain $D_T = \{X_T, y_T\}$. Clearly the score distributions are different $P(X_S) \neq P(X_T)$ (Fig.2), yet D_S contains valuable information to guide the meta-classification problem defined in D_T : func$(X_T) \rightarrow y_T$. Among transfer learning approaches, a good fit is transfer-adaboost (TrAdaBoost)[1], which is a generalization of AdaBoost[2] that leverages source domain data with a different distribution given limited sampling of the target domain.

2. Our Framework

The goal of our base-training-data recycling framework is to use both the meta-level training data $D_T = \{X_T, y_T\}$ and to transfer the knowledge from base-level training data $D_S = \{X_S, y_S\}$. Let $X_S = (x_S^{(1)}, \ldots, x_S^{(M)})$ and $X_T = (x_T^{(1)}, \ldots, x_T^{(M)})$, where M is the number of base classifiers. We first do a histogram equalization to re-balance the training score distribution according to the testing score distribution on each base classifier as in Fig.2, so that the source domain after marginal equalization $\hat{D}_S = \{\hat{X}_S, y_S\}$ has the same marginal score-distribution on each base classifier: $P(\hat{x}_S^{(m)}) = P(x_T^{(m)})$, for $m = 1, \ldots, M$. Note the positive and negative data are adjusted separately.

After histogram equalization, the joint score distribution of $\hat{X}_S$ and X_T are still different despite their identical marginal distribution. We therefore adopt the TrAdaBoost algorithm to learn a meta-level fusion classifier given both $\hat{D}_S$ and D_T. We extract an M-by-1 score vector x_i from each data sample $i \in \{1, \ldots, N_S\}$ indexing the balanced source domain (base-training) data from $\hat{D}_S$, and $i = in\{n_S + 1, \ldots, n_S + n_T\}$ indexing the target domain (unseen meta-training) data; the detailed algorithm is given in Algorithm 1.

With respect to training data recycling, the crucial feature of TrAdaBoost is that the cost c_i for data i in the target domain D_T increases when the fusion residue is big so that the following iterations will focus on 'tough' data. On the other hand, c_i for data i in the source domain $\hat{D}_S$ decreases if the residue is big, indicating data i in $\hat{D}_S$ doesn't quite fit into D_T.

As to the fusion learner, let the overall data and their fusion residue be organized in $x = [(x_1, \ldots, x_{n_S+n_T})^T, \overline{1}]$ and $e = (e_1, \ldots, e_{n_S+n_T})^T$, respectively, where $\overline{1}$ is a $(n_S + n_T)$-by-1 auxiliary one vector. With the costs organized in a diagonal matrix $\Lambda(i,i) = c_i$, we apply linear, regularized least-square fusion and solve for a weighted MMSE solution that minimizes mean-squared fusion residue:

$$W^* = \arg\min_w \{e^T \cdot \Lambda \cdot e + \lambda \|w\|^2\}, \qquad (1)$$

where λ controls regularization (we use $\lambda = 0.01$), and $e = x \cdot w - y$. The MMSE solution is thus given by

$$W^* = (x^T \Lambda x + \lambda I)^{-1} x^T \Lambda y. \qquad (2)$$

Also note that in the testing stage, we combine fusion classifiers from *all* iterations, which differs from traditional binary-classification-based TriAdaBoost. This is because our pre-balanced the marginal distribution of X_S, empirically, already performs well.

Algorithm 1 TrAdaBoost for training data recycling

Input: $x_i \in \Re^M$, $y_i \in \{0,1\}$, $i = 1, \ldots, n_S + n_T$

Initialize: cost vector $c_i = 1, i = 1, \ldots, n_S + n_T$

For $t = 1, \ldots, T$

1. normalize the cost vector $c_i = c_i / (\sum_i c_i)$
2. fusion learner $f^{(t)}(x_i) \rightarrow [0, 1]$
3. fusion residue $e_i = |f^{(t)}(x_i) - y_i|$
4. target domain error $\epsilon = \frac{\sum_{i=n_S+1}^{n_S+n_T} c_i e_i}{\sum_{i=n_S+1}^{n_S+n_T} c_i}$
5. set $\beta_t = \epsilon/(1-\epsilon)$, $\beta = 1/(1+\sqrt{2 \ln n_S / T})$
6. update the cost

$c_i \rightarrow c_i \cdot \beta^{e_i}, \quad i = 1, \ldots, n_S$

$c_i \rightarrow c_i \cdot \beta_t^{-e_i}, \quad i = n_S + 1, \ldots, n_S + n_T$

Output: $f^{(t)}$ and $\alpha_t = -\log \beta_t$, for $t = 1, \ldots, T$

Testing stage: fused score $s(x_i) = \sum_{t=1}^{T} f^{(t)}(x_i) \alpha_t$

3 Experiments

We experiment on video event detection of 5 challenging video categories from the *TRECVID2011* dataset: *attempting a board trick*; *feeding an animal*; *landing a fish*; *wedding ceremony* and *woodworking*. We conduct stacked learning with $M = 4$ base classifiers, each of which estimates event probability based on a different multimedia feature:

- Motion is captured by a bag of words feature on 3D histograms of oriented gradients [4], classified by an SVM with Histogram Intersection Kernel (HIK).
- The relationship between events and objects is captured using the Object Bank feature [5], computed using the reference code, and the maximum response of each detector across the clip's frames is classified with an SVM using HIK.
- The relationship between events and their environments is captured using the Gist feature [8], which is computed on a random 20 frame subset of the video, and the 20 outputs of a per-frame linear SVM are averaged to give a base classifier score.
- Low-level audio information is captured using Mel-Frequency Cepstral Coefficients (MFCCs), computed using the HTK Speech Recognition Toolkit[2], and an SVM with HIK is trained using a bag of words quantization of the MFCC features.

The training dataset contains 2062 videos, with around 100 positive labels per event category. We split 80% of the data for training the 4 base classifiers (fixed) and subsets of the remaining 20% are used for learning the meta-level classifier. The testing dataset contains 4292 videos with on average 101 positive labels per event category. Both the training and testing sets are imbalanced, with negative labels heavily outnumbering positive labels.

The overall performance of the ranking system is evaluated using *average precision* (AP), defined as

$$AP = \frac{1}{N_p} \sum_{i \in \{y^+\}} Pr(i), \tag{3}$$

where N_p is the number of positive labels, $Pr(i)$ is the precision statistics based on top-ranked data with a cut-off at the ith positive data. The AP statistics is equivalent to the Area-Under-ROC-Curve (AUC) statistic or normalized Wilcoxon-Mann-Whitney (WMW) ranking statistics[3, 13]. We also evaluate the performance of the system on its best operating point based on $F1$ statistics, defined as

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}. \tag{4}$$

The goals of our experiments are to understand how our method performs compared to other stacked learning approaches, and to understand how these performances depend on the ratio r between the number of meta-training (held out) and base-training (held in) clips. We bootstrap base-training data (sampling with replacement) as D_S, and sample a subset of the output score from meta-training data as D_T, thereby varying ratio r from $\frac{1}{4}$ down to $\frac{1}{20}$. We repeat this 100 times, and evaluate the average AP and $F1$ performance on all 5 events. We compare our approach with 3 others:

- **Baseline 1:** meta-training using only meta-training data.
- **Baseline 2:** naively concatenating base-training data with meta-training data.
- **Average fusion** of base-classifier likelihood scores with no meta-training.

For all but average fusion, we use a regularized least-square fusion classifier (Eqn. 2), which has shown to yield to better performances at meta-level than non-linear SVM according to [6].

Quantitative comparisons are plotted in Fig.4. Average fusion (black line) gives the worst performance, indicating the necessity of supervised learning of meta-level classifier. Baseline 2 (blue) performs second worst in general, confirming the point that base-training data

[2] http://htk.eng.cam.ac.uk/

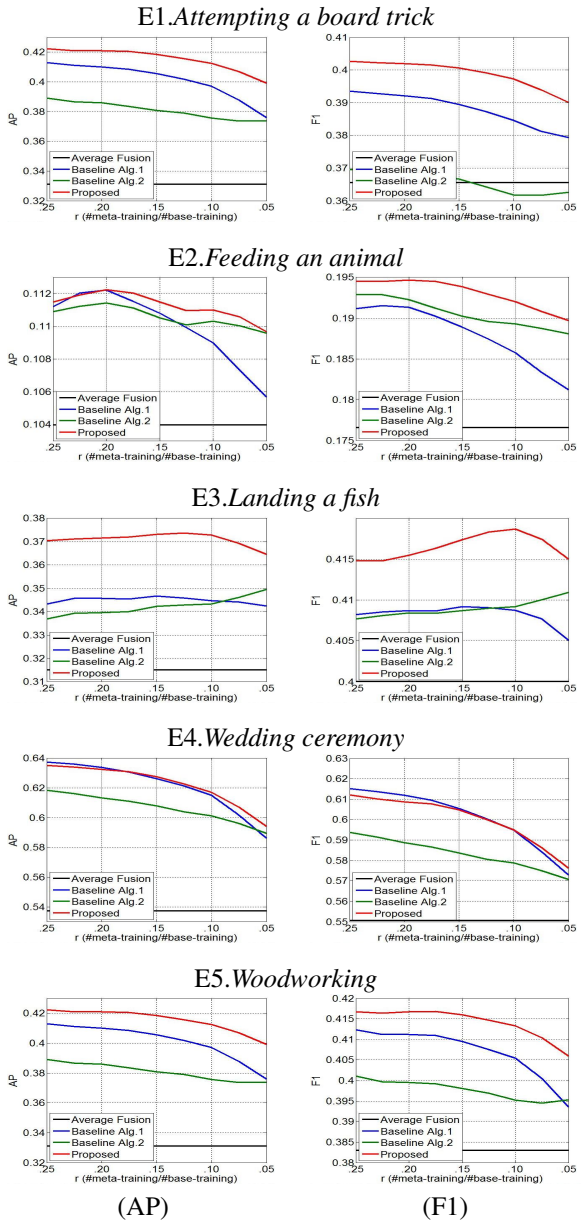

Figure 4. Quantitative comparison on 5 events. AP/F1 statistics are plotted on the left/right.

cannot be directly applied to train the meta-level classifier. By either metric, our approach (red) clearly outperforms the other methods on four events, and matches the performance of baseline 1 (green) on *wedding ceremony*. It also can be seen in general that the advantage of training-data recycling (red) over traditional methods (blue) increases as meta-level training data decreases.

We also show, in Fig. 5, the *AP*/*F1* performance changes with different number of triAdaboost iterations, taking E1 with $r = 0.25$ as an example. Iteration-1 indicates the performance after we rebalancing the data based on histogram prior than applying triAdaboost. The performance gain is obvious after the 1st iteration triAdaboost being applied and it keeps increasing in general.

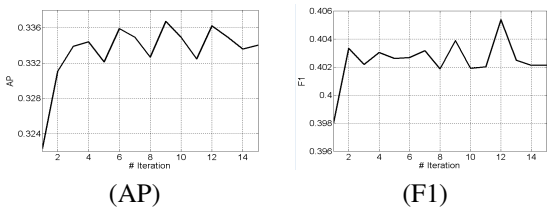

Figure 5. Performance change (E1) with the number of triAdaboost iterations.

4 Conclusion

We propose a novel framework in stacked learning to re-used base-level training data for meta-level learning. We address this problem as a knowledge transfer and first apply a histogram re-balancing to the marginal distribution of source-domain features (base-classifier score output on held-in data) according to target-domain features (score output on held-out data). We then adapt the TriAdaBoost algorithm, with a weighted least-square fusion learner, for training the meta-level score fusion. Experiments of our framework on detecting 5 challenging video events demonstrate obvious performance gains relative to other approaches.

Acknowledgements

The authors would like to thank the following for providing base classifier scores used in our experiments: Byungki Byun, Ilseo Kim, Ben Miller, Greg Mori, Sangmin Oh, Amitha Perera, and Arash Vahdat.

This work was supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior National Business Center contract number D11PC20069. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DOI/NBC, or the U.S. Government.

References

[1] W. Dai, Q. Yang, G.-R. Xue, and Y. Yu. Boosting for transfer learning. In *ICML*, pages 193–200, 2007.

[2] Y. Freund and R. Schapire. A decision theoretic generalization of online learning and an application to boosting. *JCSS*, 55(1):119–139, 1997.

[3] A. Herschtal and B. Raskutti. Optimizing area under the roc curve using gradient descent. In *ICML*, 2004.

[4] A. Klaser, M. Marszalek, and C. Schmid. A spatio-temporal descriptor based on 3d-gradients. In *BMVC*, 2008.

[5] L. Li, H. Su, E. Xing, and F. Li. Object bank: A high-level image representation for scene classification & semantic feature sparsification. In *NIPS*, 2010.

[6] J. Liu, S. McCloskey, and Y. Liu. Local export forest of score fusion for video event classification. In *ECCV*, 2012.

[7] C. Ma and C. Lee. An efficient gradient computation approach to discriminative fusion optimization in semantic concept detection. In *ICPR*, 2008.

[8] A. Oliva and A. Torralba. Modeling the shape of the scene: a holistic representation of the spatial envelope. *IJCV*, 42(3):145–175, 2001.

[9] P. Over, G. Awad, J. Fiscus, B. Antonishek, A. F. Smeaton, W. Kraaij, and G. Quenot. TRECVID 2010 – An Overview of the Goals, Tasks, Data, Evaluation Mechanisms and Metrics. In *TRECVID*, 2011.

[10] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Trans. on Knowledge and Data Engineering*, 22:1345–1359, 2010.

[11] J. Sill, G. Takacs, L. Mackey, and D. Lin. Feature-weighted linear stacking. In *arXiv:0911.0460*, 2009.

[12] D. H. Wolpert. Stacked generalization. *Neural Networks*, 5:241–259, 1992.

[13] L. Yan, R. Dodier, M. Mozer, and R. Wolniewicz. Optimizing classifier performance via an approximation to the wilcoxon-mann-whitney statistics. In *ICML*, 2003.