On Using Nearly-Independent Feature Families for High Precision and Confidence

Omid Madani MADANI@GOOGLE.COM
Manfred Georg MGEORG@GOOGLE.COM
David A. Ross DROSS@GOOGLE.COM
Google Inc., Mountain View, CA 94043, USA

Editor: Steven C.H. Hoi and Wray Buntine

Abstract

Often we require classification at a very high precision level, such as 99%. We report that when very different sources of evidence such as text, audio, and video features are available, combining the outputs of base classifiers trained on each feature type separately, aka late fusion, can substantially increase the recall of the combination at high precisions, compared to the performance of a single classifier trained on all the feature types *i.e.*, early fusion, or compared to the individual base classifiers. We show how the probability of a joint false-positive mistake can be upper bounded by the product of individual probabilities of conditional false-positive mistakes, by identifying a simple key criterion that needs to hold. This provides an explanation for the high precision phenomenon, and motivates referring to such feature families as (nearly) independent. We assess the relevant factors for achieving high precision empirically, and explore combination techniques informed by the analysis. We compare a number of early and late fusion methods, and observe that classifier combination via late fusion can more than double the recall at high precision.

Keywords: Classifier Combination, Independent Features, High Precision, Late Fusion, Early Fusion, Ensembles, Multiple Views, Supervised Learning

1. Introduction

In many classification scenarios, in surveillance or in medical domains for example, one needs to achieve high performance at the extreme ends of the precision-recall curve.[1] For some tasks such as medical diagnosis and surveillance (for detecting rare but dangerous objects, actions, and events), a very high recall is required. In other applications, for instance for the safe application of a treatment or quality user experience, a high precision is the goal. In this paper, we focus on achieving high precision. In particular, the goal in our video classification application is maximizing recall at a very high precision threshold, such as 99%. This has applications to improved user experience and advertising. Self-training paradigms such as co-training (Blum and Mitchell, 1998) can also benefit from automatically acquired labeled data with very low false-positive rates. Achieving high precision raises a

[1] In binary classification, given a set of (test) instances, let T denote the set of truely positive instances, and let $\tilde{T}$ be the set that a classifier classifies as positive. The precision of the classifier is $\frac{|T \cap \tilde{T}|}{|\tilde{T}|}$, while recall is $\frac{|T \cap \tilde{T}|}{|T|}$. A precision-recall curve is obtained by changing the threshold at which the classifier classifies positive, from very conservative or low recall (small size $|\tilde{T}|$) to high recall.

number of challenges: features may be too weak or the labels may be too noisy to allow the classifiers to robustly reach such high precision levels. Furthermore, verifying whether the classifier has achieved high precision can require substantial amounts of labeled data.

On the other hand, many applications, *e.g.*, in multimedia, provide diverse sets of feature families and distinct ways of processing the different signals. Given access to a number of different feature families, a basic question is how to use them effectively. Consider two extremes: training one classifier on all the features, aka early fusion or fusion in the feature space, versus training separate classifiers on each family then combining their output, aka late fusion[2] or fusion in classifier/semantic space (Snoek et al., 2005). Training a single classifier on all the families has the advantage of simplicity. Furthermore, the learner can potentially capture interactions among the different features. However, there are complications: one feature family can be relatively dense and low dimensional, while another very high dimensional and sparse. Creating a single feature vector out of all may amount to mixing apples and oranges. This can require considerable experimentation for scaling individual feature values and whole feature families (and/or designing special kernels), and yet, learning algorithms that can effectively integrate all the features' predictiveness may not exist. Furthermore, for a significant portion of the instances, whole feature families can be missing, such as absent audio or speech signals in a video. Training separate classifiers then combining the outputs, may lose the potential of learning from feature interactions across different modalities, but it offers advantages: one can choose appropriate learning algorithms for each feature family separately, and then combine them for best results.

In this work, we find that training distinct base classifiers offers an important benefit with respect to high precision classification, in particular for maximizing recall at a high precision threshold. Feature families based on very different signals, for example, text, audio, and video features, can complement one another and yield independent sources of evidence. The pattern of false-positive errors that base classifiers make, each trained on a single feature family, may therefore be nearly independent. Using an independence assumption on false-positive mistakes of base classifiers and an additional positive correlation assumption, we derive a simple upper bound, basically the product of individual conditional false-positive probabilities, via Bayes' formula, on *joint* false-positive mistake (in case of two classifiers, the event of both classifiers making a mistake, given both classify positive).[3] Our subsequent analysis relaxes the assumptions and discovers a key intuitive property that needs to hold for the substantial drop in the probability of joint mistakes. Furthermore, such properties can be tested on heldout data, and thus the increased confidence in classification can be examined and potentially verified (requiring substantially less labeled data than brute-force validation). In our experiments on classification of videos, we find that recall can be more than doubled at high precision levels via late fusing of the nearly-independent base classifiers. We summarize our contributions as:

1. We report on the phenomenon of boosted precision at the beginning of the precision-recall curve, when combining independent feature families via late fusion.[4] We present

[2] Early fusion subsumes late fusion, if one imagines the learning search space large enough to include both learning of separate classifiers and combining. But early *vs.* late is a useful practical distinction.

[3] The bound has the same form as the Noisy-OR model (Henrion, 1987)

[4] In other words, the so-called *Duck Test* rings true! "If it looks like a duck, swims like a duck, and quacks like a duck, then it is probably a duck." See en.wikipedia.org/wiki/Duck_test

analyses that explain the observations and suggest ways for fusing classifiers as well as for examining dependencies among classifier outputs.

2. We conduct a number of experiments that demonstrate the high-precision phenomena, and compare several fusion techniques. Informed by our analysis, we illustrate some of the tradeoffs that exist among the different techniques.

2. Analyzing Fusion Based on False-Positive Independence

We focus on the binary classification setting in this paper, and on the two classifier case.[5] Each instance is a vector of feature values denoted by x, and has a true class denoted $y_x, y_x \in \{0, 1\}$. We are interested in high precision classification, and therefore analyze probability of (conditional) false-positive events. In deriving an upper bound on probability of joint false-positive mistake (equation 1 below), we make use of two assumptions regarding the way the classifiers' outputs are interdependent. We then discuss these assumptions, and subsequently present a relaxation that yields a more basic criterion. The assumptions:

1. Independence of false-positive mistakes:
   $P(f_2(x) = 1|y_x = 0, f_1(x) = 1) = P(f_2(x) = 1|y_x = 0)$
2. Positive (or non-negative) correlation: $P(f_2(x) = 1|f_1(x) = 1) \geq P(f_2(x) = 1)$,

Where $f_i(x) = 1$ denotes the event that classifier i classifies the instance as positive, and the event $(f_i(x) = 1|y_x = 0)$ denotes the conditional event that classifier i outputs positive given the true class is 0, and $(y_x = 0, f_i(x) = 1)$ is the conjunction of two events (the true class is negative, while $f_i(x) = 1$).

A simple intuitive upper bound on the probability of joint false-positive mistake can now be derived:

$$P(y_x = 0|f_2(x) = 1, f_1(x) = 1) = \frac{P(y_x = 0, f_2(x) = 1, f_1(x) = 1)}{P(f_2(x) = 1, f_1(x) = 1)} \tag{1}$$

$$\leq \frac{P(y_x = 0, f_2(x) = 1, f_1(x) = 1)}{P(f_2(x) = 1)P(f_1(x) = 1)} = \frac{P(f_2(x) = 1|y_x = 0, f_1(x) = 1)}{P(f_2(x) = 1)} \frac{P(y_x = 0, f_1(x) = 1)}{P(f_1(x) = 1)} \tag{2}$$

$$= \frac{P(f_2(x) = 1|y_x = 0)}{P(f_2(x) = 1)} \frac{P(y_x = 0, f_1(x) = 1)}{P(f_1(x) = 1)} = \frac{P(f_2(x) = 1, y_x = 0)}{P(y_x = 0)P(f_2(x) = 1)} P(y_x = 0|f_1(x) = 1) \tag{3}$$

$$= \quad (1 - P_2)(1 - P_1)P(y_x = 0)^{-1}, \tag{4}$$

where $P(y_x = 0)$ denotes the probability of the negative class (the negative prior), and P_i is short for $P(y_x = 1|f_i(x) = 1)$ (the "confidence" of classifier i that instance x is positive, or posterior probability of membership, or equivalently, precision of classifier i). Positive correlation was used in going from (1) to (2), and independence of false-positive events was used in (2) to (3). The bound in (4) has the form of a Noisy-OR model (Henrion, 1987).

Often, the positive class is tiny and $P(y_x = 0)^{-1} \approx 1$. Thus, the probability of failure can decrease geometrically, *e.g.*, from 10% error for each classifier, to 1% for the combination.

[5] Generalization of the bound to more than two binary classifiers is not difficult (using induction and generalizations of the assumptions).

This possibility of near geometric reduction in false-positive probability is at the core of the potential for substantial increase in precision, via late fusion in particular. In this paper, our focus is in further understanding and utilizing this phenomenon.

2.1. Discussion of the Assumptions

There is an interesting contrast between the two assumptions above: one stresses *independence*, given the knowledge of the class, the other stresses *dependence*, given lack of such knowledge. The positive correlation assumption is the milder of the two and we expect it to hold more often in practice. However, it does not hold in cases when, for example, the two classifiers' probability outputs are mutually exclusive (*e.g.*, the classifiers output 1 on distinct clusters of positive instances). In our experiments, we report on the extent of the correlation. Very importantly, note that we obtain an extra benefit from positive correlation, if it holds: given that substantial correlation exists, the number of instances on which both classifiers output positive would be significantly higher than independence would predict.

Let us motivate assumption 1 on independence of false-positive mistakes when each classifier is trained on a feature family that is distant from the others. In the case of video classification, imagine one classifier is trained on visual features, while another is trained on textual features derived from the video's descriptive metadata (*e.g.*, title, description, etc). A plausible expectation is that the error-inducing ambiguities in one feature domain that lead to classifier errors do not co-occur with the ambiguities in the other domain. For example, "Prince of Persia" refers both to a movie and a video game, but it is easy to tell them apart by the visuals. There can of course be exceptions. Consider the task of learning to contrast two games in a video game series (such as "Uncharted 2" and "Uncharted 3"), and more genreally, but less problematic, video games in the same genre. Then the textual features may contain similar words, and the visuals could also be somewhat similar.

2.2. A Relaxation of the Assumptions

As we discussed above, base classifiers trained on different feature families may be only roughly independent in their false-positive behavior. Here, we present a relaxation of the assumptions that shows that the geometric reduction in false-positive probability has wider scope. The analysis also yields an intuitive understanding of when the upper bound holds.

When we replaced $P(f_2(x) = 1, f_1(x) = 1)$ by $P(f_2(x) = 1)P(f_1(x) = 1)$, we could instead introduce a factor, which we will refer to as positive correlation ratio r_p (the desired or "good" ratio):

$$r_p = \frac{P(f_2(x) = 1, f_1(x) = 1)}{P(f_2(x) = 1)P(f_1(x) = 1)},$$

Thus, the first step in simplifying the false-positive probability can be written as:

$$P(y_x = 0 | f_2(x) = 1, f_1(x) = 1) = \frac{P(y_x = 0, f_2(x) = 1, f_1(x) = 1)}{r_p P(f_2(x) = 1)P(f_1(x) = 1)}$$

The numerator can be rewritten in the same way, by introducing a factor which we will refer to as the false-positive correlation ratio, r_{fp} (the "bad" ratio):

$$r_{fp} = \frac{P(f_2(x) = 1, f_1(x) = 1, y_x = 0)}{P(f_2(x) = 1, y_x = 0)P(f_1(x) = 1, y_x = 0)},$$

Therefore:

$$P(y_x = 0|f_2(x) = 1, f_1(x) = 1) = \frac{r_{fp} P(f_2(x) = 1, y_x = 0) P(f_1(x) = 1, y_x = 0)}{r_p P(f_2(x) = 1) P(f_1(x) = 1)} = \frac{r_{fp}}{r_p}(1 - P_2)(1 - P_1).$$

Thus as long as the *bad-to-good* ratio $r = \frac{r_{fp}}{r_p}$ is around 1 or less, we can anticipate a great drop in the probability that both classifiers are making a mistake, in particular $(1 - P_2)(1 - P_1)$ is an upperbound when $r \leq 1$. The ratios r_p and r_{fp} can be rewritten in conditional form[6] as:

$$r_p = \frac{P(f_2(x) = 1|f_1(x) = 1)}{P(f_2(x) = 1)}, r_{fp} = \frac{P(f_2(x) = 1, y_x = 0|f_1(x) = 1, y_x = 0)}{P(f_2(x) = 1, y_x = 0)} \tag{5}$$

Both ratios involve a conditioned event in the numerator, and the unconditioned version in the denominator. Either measure can be greater or less than 1, but what matters is their ratio. For example, as long as the growth in the conditional overall positive outputs (r_p) is no less than the conditional false-positive increase r_{fp}, the product bounds the false-positive error of combination. We can estimate or learn the ratios on heldout data (see Sections 3.4 and 3.7). In our experiments we observe that indeed, often, $r_{fp} > 1$ (false-positive events are *NOT* necessarily independent, even for very different feature families), but also $r_p > r_{fp}$. The analysis makes it plausible that instances that are assigned good (relatively high) probabilities by *both* base classifiers are very likely positive, which explains why fusing by simply summing the base classifier scores can yield high precision at top rankings as well. We also compare this fusion-via-summation technique.

2.3. Events Definitions and Event Probabilities

We require probabilities for the conditional events of the sort $(y_x = 1|f_i(x) = 1)$, *i.e.*, posterior probability of class membership. Many popular classification algorithms, such as support vector machines, don't output probabilities. Good estimates of probability can be obtained by mapping classifier scores to probabilities using held-out (validation) data (*e.g.*, Niculescu-Mizil and Caruana (2005); Zadrozny and Elkan (2002)). Here, we generalize the events that we condition on to be the event that the classifier score falls within an interval (a bin). We compute an estimate of the probability that the true class is positive, given the score of the classifier falls in such intervals. One technique for extracting probabilities from raw classifier scores is via sigmoid fitting (Platt, 1999). We instead used the technique of binning the scores and reporting the proportion of positives in a bin (interval) as probability estimates, because sigmoid fitting did not converge for some classes, and importantly, we wanted to be conservative when estimating high probabilities. In various experiments, we did not observe a significant difference (*e.g.*, in quadratic loss) when using the two techniques.

[6] Both ratios are pointwise mutual information quantities between the two random events (Manning and Schutze, 1999).

3. Experiments

We report on game classification of videos. Game classification is the problem of classifying whether a video depicts mostly gameplay footage of a particular game.[7] Our particular objective in this application is to maximize recall at a very high precision, such as 99%. For evaluation and comparisons, we look both at ranking performance, useful in typical user-facing information-retrieval applications, as well as the problem of picking a threshold, using validation data, that with high probability ensures the desired precision. The latter type of evaluation is motivated by our application and more generally by decision theoretic scenarios where the system should make binary (committed) decisions or provide good probabilities. We begin by describing the experimental setting, then provide comparisons under the two evaluations, with discussions. Most of our experiments focus on visual and audio feature families. We report on the extent of dependencies among the two, and present results that include other feature families.

For experiments in this paper, we chose 30 game titles at random, from amongst the most popular recent games. We treat each game classification as a binary 1-vs-rest problem. For each game, we collected 3000 videos that had the game title in their video title. Manually examining a random subset of such videos showed that about 90% of the videos are truly positive (the rest are irrelevant or do not contain gameplay). For each game, videos from other game titles constitute the negative videos, but to further diversify the negative set, we also added 30,000 videos to serve as negatives from other game titles. The data, of 120,000 instances was split into 80% training, 10% validation, and 10% test.

3.1. Video Features and Classifiers

The video content features used include several different types, both audio (Audio Spectrogram, Volume, Mel Frequency, ..) and visual (Global visual features such as 8x8 hue-saturation, and PCA of patches at spatio-temporal interest points,..). For each type, features are extracted at every frame of the video, discretized using k-means vector quantization, and summarized using a histogram, one bin for each codeword (Toderici et al., 2010). Histograms for the various feature types are individually normalized to sum to 1, then concatenated to form a feature vector. The end result is roughly 13000 audio features and 3000 visual features. Each video vector is fairly dense (only about 50% are zero-valued). We also include experiments with two text related feature families, which we describe in (Section 3.6).

We used the passive-aggressive online algorithm as the learner (Crammer et al., 2006). This algorithm is in the perceptron linear classifier family. We used efficient online learning because the (video-content) feature vectors contain tens of thousands of dense features, and even for our relatively small problem subset, requiring all instances to fit in memory (as batch algorithms do) is prohibitive. For parameter selection (aggressiveness parameter and number of passes for passive-aggressive), we chose the parameters yielding best average Max F1,[8] on validation data for the classifier trained on *all features* appended together. This is our *early fusion* approach. We call this classifier *Append*. The parameters were 7 passes,

[7] These “gameplay” videos are user uploaded to YouTube, and can be tutorials on how to play a certain stage, or may demonstrate achievements, and so on.

[8] F1 is the harmonic mean of precision and recall. The maximum is taken over the curve for each problem.

and aggressiveness of 0.1, though the differences, *e.g.*, between aggressiveness of 1 and 0.01 were negligible at 0.774 and 0.778 resp. We also chose the best scaling parameter among $\{1, 2, 4, 8\}$ between the two feature families, using validation for best recall at 99% precision, and found scaling of 2 (on visual) to be best. We refer to this variant as Append$^+$. For classifiers trained on other features, we use the same learning algorithm and parameters as we did for Append. We note that one could use other parameters and different learning algorithm to improve the base classifiers.

We have experimented with 2 basic types of late fusion: fusion using the bound 4 of Section 2 (IND), where false-positive probability is simply the product of the false-positive probabilities of base classifiers, *i.e.*, Noisy-OR combination (and where we set $P(y_x = 0) = 0.97$, the negative prior), and fusion using the average of base classifier probability scores (AVG). In Section 3.5, we also report on learning a weighting on the output of each classifier (stacking), and we describe another stacking variant, IND Adaptive, as well a simpler hybrid technique, IND+AVG in Section 3.7.

3.2. Ranking Evaluations

Table 1 reports recalls at different (high) precision thresholds,[9] and max F1, for audio and visual classifiers as well as early (Append, Append$^+$) and late fusion techniques (IND, AVG, and extensions). Figure 3.2 shows the precision-recall curves for a few classifiers on one of the problems. We observe that late fusion substantially improves performance at the high precision thresholds or regions of the curve. This is remarkable in that we optimized the parameters (experimenting with several parameters and picking the best), for the early fusion (Append) techniques. It is possible that more advanced techniques, such as multi-kernel learning, may significantly improve the performance of the early fusion approach, but a core message of this work is that late fusion is a simple efficient approach to utilizing nearly-independent features for high precision (see also the comparisons of Gehler and Nowozin (2009)). Importantly, note that max F1 is about the same for many of the techniques. This underscores the distinction that we want to make that the main performance benefit of late over early fusion, for nearly-independent features, appears to be mainly early in the precision-recall curve.

We will be using rec@99 for recall at 99% precision. When we pair the rec@99 values for each problem, at the 99% precision threshold, AVG beats all other methods above it in the table, and IND beats Append and the base classifiers (at 99% confidence level). As we lower the precision threshold or if we compare max F1 scores, the improvements from late fusion decrease.

hsp

The improvement in recall at high precision from late fusion should grow when the baseline classifiers have comparable performance, and all do fairly well, but not necessarily exceptionally well! Figure 2 illustrates this (negative) correlation with the absolute difference in F1 score between the base classifiers: the smaller the difference, in general the stronger the boost from late fusion.[10]

[9] In these results, we rank the test instances by classifier score and compute precision/recall.

[10] Interestingly Append$^+$ appears to have an advantage when the performances of one feature family dominates the other (high x values). We leave further exploration of this observation to future work.

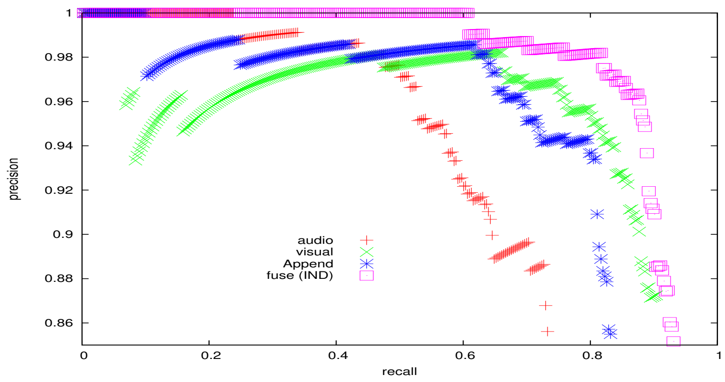

Figure 1: Precision vs. recall curves, for classifier trained on visual only, audio only, the union of the two features, Append, and using fusion, on one of the 30 game classes ("Kingdom Hearts"). Fusion substantially increases recall at high precisions.

Table 1: Average recall, over 30 classes, for a few precision thresholds on the test set.

Prec. →	99%	95%	90%	Max F1
Audio	0.046	0.093	0.13	0.51
Visual	0.13	0.50	0.63	0.81
Append	0.14	0.41	0.59	0.78
Append$^+$	0.26	0.39	0.57	0.82
IND	0.33	0.55	0.66	0.82
AVG	0.45	0.62	0.70	0.82
IND+AVG	0.45	0.62	0.72	0.83
IND Adaptive	0.47	0.65	0.72	0.83

3.3. Threshold Picked using Validation Data

We now focus on the setting where a threshold needs to be picked using the validation data, *i.e.*, the classifier has to decide on the class of each instance in isolation during testing. See Table 2. Thus in contrast to table 1, in which the best threshold was picked on test instances, here, we assess how the probabilities learned on validation "generalize".

In our binning, to map raw score to probabilities, we require that a bin have at least 100 points, and 99% of such points to be positive, for its probability estimate ≥ 0.99. Therefore in many cases, the validation data may not yield a threshold for a high precision, when there is insufficient evidence that the classifier can classify at 99% precision. For a given binary problem, let E_τ denote the set of test instances that obtained a probability no less than the desired threshold τ. E_τ is empty when there is no such threshold or when no test instances meet it. The first number in the triples shown is the number of problems (out of 30) for which $|E_\tau| > 0$ (the set is not empty). For problems with $|E_\tau| > 0$, let E^p_τ denote the number of (true) positive instances in E_τ. The second number in the triple is number of problems for which $\frac{|E^p_\tau|}{|E_\tau|} \geq \tau$ (the ratio of positives is greater than desired threshold τ).

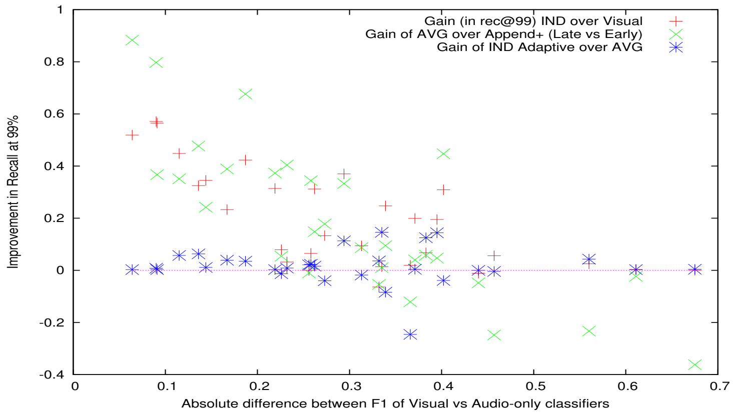

Figure 2: Each point corresponds to one problem. The x-coordinate for all points is the absolute difference in max F1 performance of audio and visual-only base classifiers. For the first two plots, the y-coordinate is the gain, *i.e.*, the difference in recall at 99% (rec@99). The first plot shows the gains of IND (in rec@99) over the visual classifier, the 2nd is the gain of AVG over the Append$^+$ classifier, and the 3rd is the gain of IND Adaptive over average. In general, the closer the performance of the two base classifiers, the higher the gain when using late fusion. For many of the problems, the difference in rec@99 is substantial.

Threshold $\tau \rightarrow$	$\geq$ 0.99	$\geq$ 0.95
Audio	(0, 0, 0)	(8, 4, 0.32)
Visual	(8, 3, 0.653)	(24, 20, 0.56)
Append (early fuse)	(3, 1, 0.826)	(26, 16, 0.50)
Append$^+$ (early fuse)	(7, 3, 0.60)	(23, 20, 0.63)
IND	(24, 18, 0.35)	(29, 22, 0.56)
AVG	(0, 0, 0)	(13, 13, 0.19)
calibrated AVG	(17, 12, 0.65)	(30, 26, 0.62)
IND+AVG	(24, 22, 0.322)	(28, 26, 0.45)
IND Adaptive	(29, 22, 0.43)	(30, 25, 0.59)

Table 2: For each classifier and threshold combination, we report three numbers: The number of problems (out of 30), where some test instances obtained a probability $\geq$ the threshold τ, the number of "valid" problems, *i.e.*, those problems on which the ratio of positives with score exceeding τ to all such instances is at least τ, and the average recall at threshold τ (averaged over the valid problems only).

Note that, due to variance, the estimated true positive proportion may fall under the threshold τ for a few problems. There are two types of variance. For each bin (score range), we extract a probability estimate, but the true probability has a distribution around this estimate.[11] Another variation comes from our test data: while the true probability may be equal or greater than a bin's estimate, the estimate from test instances may indicate otherwise due to sampling variance.[12] The last number in the triple is the average recall at threshold τ for those problems on which $\frac{|E_\tau^p|}{|E_\tau|} \geq \tau$.

Fusion using IND substantially increases the number of classes on which we reach or surpass high thresholds, compared to early fusion and base classifiers, and is superior to AVG based on this measure. As expected, plain AVG does not do well specially for threshold $\tau = 0.99$, because its scores are not calibrated. However, once we learn a mapping of (calibrate) its scores (performed on the validation set), calibrated AVG improves significantly on both thresholds. IND being based on an upperbound on false-positive errors, is conservative: on many of the problems where some test instances scored above the 0.99 threshold, the proportion of true positives actually was 1.0. On problems that both calibrated AVG and IND vairants reach 0.99, calibrated AVG yields a substantially higher recall. IND is a simple technique and the rule of thumb in using it would be that if calibration of AVG does not reach the desired (99%) threshold, then use IND (see also IND+AVG in Section 3.7). We note that in practice, with many 100s to 1000s of classes, it can be the case that the validation may not provide sufficient evidence that AVG reaches 99% (in general, a high precision), and IND can be superior.

3.4. Score Spread and Dependencies

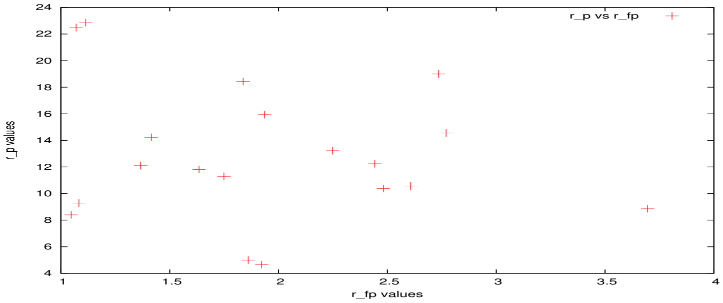

Figure 3: The positive correlation (good) ratios, r_p (y axis), versus dependency ratios r_{fp}, on 19 games, for threshold $\tau = 0.2$ (see Section 3.4), measured on test ($f_i(x) = 1$ if $P_i \geq \tau$). Note that for all the problems, the bad-to-good ratio $r_{fp}/r_p < 1$.

For a choice of threshold τ, let the event $f_i(x) = 1$ mean that the score of classifier i exceeds that threshold. For assessing extent of positive correlation, we looked at the ratios r_p (eq. 5, Section 2.2), where f_1 is the visual classifier and f_2 is the audio classifier. For $\tau \in \{0.1, 0.2, 0.5, 0.8\}$, r_p values (median or average) were relatively high (≥ 14). Figure 3 shows the spread for $\tau = 0.2$. We also looked at false-positive dependence and in particular

[11] This variance could be estimated and used for example for a more conservative probability estimation, though we don't pursue that here.

[12] Note also that many test instances may obtain higher probabilities than τ, and thus the expected proportion of positives can be higher than τ.

r_{fp}. For relatively high $\tau \geq 0.5$, we could not reliably test whether independence was violated: while we observed 0 false positives in intersection, the prior probability of false positive is also tiny. However, for $\tau \geq 0.2$, we could see that for many problems (but not all), the null hypothesis that the false positives are independent could reliably be rejected. This underscores the importance of our deriviations of Section 2.2: Eventhough the feature families may be very different, some dependence of false positives may still exist. We also pooled the data over all the problems and came to the same conclusion. However, r_{fp} is in general relatively small, and $r_p \geq r_{fp}$ for all the problems and thresholds we looked at (Figure 3). In contrast, see Section 3.6 and Table 4, when features families are close.

Note that if the true rec@99 of the classifier is x, and we decide to require y many positive instances ranked highest to verify 99% precision (eg $y = 100$ is not overly conservative), then in a standard way of verification, we require to sample and label y/x many positive instances for the validation data. In our game classification experiments, we saw that base classifiers' rec@99 were rather low (around 10 to 15% on test data, Table 1). This would require much labeled data to reliably find a threshold at or close to 99%. Yet with fusion, we achieved that precision on more than a majority of the problems (Table 2).

3.5. Learning a Weighting (Stacking)

We can take a stacking approach (Wolpert, 1992) and learn on top of classifier outputs and other features derived from them. We evaluated a variety of learning algorithms (linear SVMs, perceptrons, decision trees, and random forests), comparing max F1 and rec@99. On each instance, we used as features the probability output by the video and audio classifiers, p_1 and p_2, as well as 5 other features: the product p_1p_2, $\max(p_1, p_2)$, $\min(p_1, p_2)$, $\frac{p_1+p_2}{2}$, and gap $|p_1 - p_2|$. We used the validation data for training and the test data for test (each 12k). For the SVM, we tested with the regularization parameters $C = 0.1, 1, 10,$ and 100, and looked at the best performance on the test set. We found that, using the best of the learners (*e.g.*, SVM with C=10) when compared to simple averaging, recall at high precision, rec@99, did not change, but max $F1$ improved by a small 1% on average (averaged over the problems). Pairing the F1 performances on each problem shows that this small improvement is significant, using the binomial sign test, at 90% confidence.[13] SVMs with C=10 and random forests tied in their performance. Because the input probabilities are calibrated (extracted on heldout data), and since the number of features is small (all are a function of p_1 and p_2), there is not much to gain from plain stacking. However, with more base classifiers, stacking may show an advantage in achieving high precision. Section 3.7 explain another kind of stacking (estimating r_p, r_{fp}) that is beneficial.

3.6. Experiments with Text-Based Features

Our training data comes from title matches, thus we expect classifiers based on text features to do well. Here, as features, we used a 1000-topic Latent Dirichlet Allocation (LDA) model (Blei et al., 2003), where the LDA model was trained on title, tags, and descriptions of a large corpus of gaming videos. Table 3 reports on the performance of this model, and its fusion with video content classifiers (using IND). We observe LDA alone does very well (noting

[13] Even using only $p1$ and $p2$ as features, gives a slight improvement in Max F1 over simple averaging, but using all the features gives additional improvement.

Table 3: Average recall, over 30 classes, for several precision thresholds on the test set, comparing classifiers trained solely on LDA (1000 topics using text features), Append (LDA, audio, visual), fusion of LDA with Append on audio-visual features (LDA+Append), and fusion of all three feature types (LDA+audio+visual). While LDA feature alone perform very well, fusion, in particular of audio, video, and LDA features, does best.

Prec. →	99%	95%	90%	Max F1
LDA	0.58	0.79	0.85	0.94
Append	0.65	0.86	0.91	0.93
LDA+Append AudioVis	0.73	0.85	0.92	0.95
LDA+Audio+Visual	0.76	0.88	0.94	0.95

Table 4: Average values of r_{fp} and r_p for several paired classifiers (at $\tau = 0.1$). Tag and LDA (LDAvsTag) classifiers are highly dependent in their pattern of false positives, and $\frac{r_{fp}}{r_p} \gg 1$. We observe a high degree of independence in the other pairings.

PAIR →	LDAvsTag	LDAvsVis	TagVsVis	VisVsAudio
r_{fp}	101	6	3	2
r_p	30	18	17	14

that our training data is biased). Still, the performance of the fusion shows improvements, in particular, when we fuse visual, audio, and LDA classifiers. Another text feature family, with high dimensionality of 11 million, is features extracted from description and tags of the videos, yielding "tags" classifiers. Because we are not extracting from the title field, the tags classifiers are also not perfect,[14] yielding an average F1 performance of 90%. Table 4 shows the r_{fp} and r_p values when we pair tag classifiers with LDA, etc. We observe very high r_{fp} values, indicating high false-positive dependence between the text-based classifiers.

3.7. Improved IND: Independence as a Function of Scores

Further examination of the bad-to-good ratio $r = r_{fp}/r_p$, both on individual per class problems, as well as pooled (averaged over) all the problems, suggested that the ratio varies as a function of the probability estimates and in particular: 1) $r \gg 1$ (far from independence), when the classifiers "disagree", *i.e.*, when one classifier assigns a probability close to 0 or the prior of the positive class, while the other assigns a probability significantly higher, and 2) $r \in [0, 1]$, *i.e.*, the false-positive probability of the joint can be significantly lower than the geometric mean, when both classifiers assign a probability significantly higher

[14] Note that combining these classifiers is still potentially useful to increase the coverage. Only a fraction of game videos's titles contain the game titles.

than the prior. Figure 4 shows two slices of the two-dimensional curve learnt by averaging the ratios over the grid of two classifier probability outputs, over the 30 games. These ratios are used by IND Adaptive to estimate the false-positive probability.[15] Note that, it makes sense that independence wouldn't apply when one classifier outputs a score close to the positive class prior: Our assumption that the classifier false-positive events are independent is not applicable when one classifier doesn't "think" the instance is positive to begin with! Inspired by this observation, a simple modification is to take an exception to the plain IND technique when one classifier's probability is close to the prior. In IND+AVG, when one classifier outputs below 0.05 (close to the prior), we simply use the average score. As seen in Tables 1 and 2, its performance matches or is superior to the best of IND and AVG. We also experimented with learning the two-dimensional curves *per game*. The performance of such, with some smoothing of the curves, was comparable to IND+AVG. The performance of IND Adaptive indicates that learning has potential to significantly improve over the simpler techniques.[16]

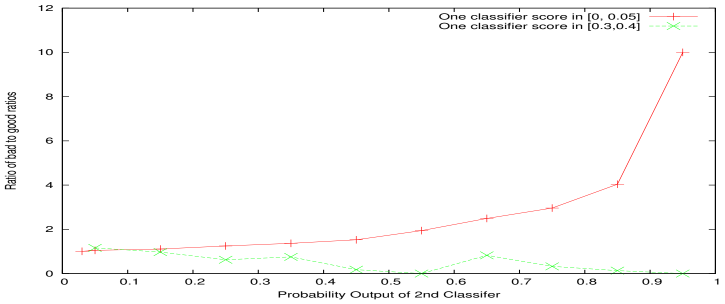

Figure 4: The bad-to-good ratio r as a function of individual classifier score ranges. When the classifiers 'disagree' (one output is near the positive prior, 0.03, while the other is higher), $r \gg 1$. But $r \approx 1$, or $r \ll 1$, when both 'agree', *i.e.*, when both outputs are higher than the positive prior (lower curve).

3.8. Discussion

The analysis that led to IND explains the success of both IND and AVG for high precision classification when feature families are nearly independent. Furthermore, in many practical scenarios, the learning problems can be so challenging that individual classifiers may fall far short of the desired precision on validation data. In other cases, the distributions of labeled data, in particular the proportion of the positive instances, may be very different in deployment versus the training/validation phase.[17] In such cases, a simple conservative approach such as IND, or IND+AVG, that substantially increases precision with decent recall may be preferred over more elaborate techniques, such as stacking, that attempt to

[15] Given p_1 and p_2, the map is used to obtain $r_{p_1p_2}$, and the product $r_{p_1p_2}(1-p_1)(1-p_2)$ is the false-positive probability. To learn the map, the domain $[0,1] \times [0,1]$, is split into grids of width 0.05, and ratio r is estimated for each grid cell for each problem, then averaged over all problems.

[16] Note that IND Adaptive has a potential advantage in that the map is estimated using multiple games.

[17] For example, often the sampling process to obtain the labeled data may have certain biases in it.

fit the validation data more closely. When we expect that the deployment data is close to the validation data and one has sufficient validation data, then estimating the parameters r_p and r_{fp} may improve performance (Section 3.7).

4. Related Work

The work of Kittler *et. al.* explores a number of classifier combination techniques (Kittler et al., 1998). In that work, the product rule has a superficial similarity to the plain IND technique. However, the product rule is more similar to a conjunction, while IND is more similar to a disjunction (or noisy-OR, see product of experts below). There are more basic difference between that work (and much subsequent work) and ours: their focus is not achieving high precision, and the treatment is for a more general setting where classifier outputs can be very correlated.[18] The literature on benefits of multiple-views, multi-classifier systems (ensembles), and fusion, with a variety of applications, is vast (*e.g.*, Ho et al. (1994); Blum and Mitchell (1998); Jain et al. (2005); Long et al. (2005); Snoek et al. (2005); Brown (2009); Gehler and Nowozin (2009)). Some explicitly consider independence (Tulyakov and Govindaraju, 2005), but in a stronger and less precise sense that classifier output distributions are independent. Often other performance measures, such as average precision over the whole precision-recall curve, equal error rate, or max F1, are reported. We are not aware of work that specifically focuses on high precision, in particular on the problem of maximizing recall at a high precision threshold, with a careful analysis of near independence of the false-positive events, *explaining* the phenomenon of increased precision at the beginning of the precision-recall curve via late fusion.

Multikernel learning is an attractive approach to early fusion, but in our setting, efficiency (scalability to millions of very high dimensional instances) is a crucial consideration, and we observed that a simple scaling variation is inferior. Prior work has found combination rules very competitive compared to multikernel learning with simplicity and efficiency advantages (Tulyakov and Govindaraju, 2005).

Fusion based on independence has a similarity to the Product of Experts (PoE) (Hinton, 2002), which combines probabilistic expert models by multiplying their outputs together and renormalizing. The product operation in PoE is a conjunction, requiring that all constraints be simultaneously satisfied. In contrast, since IND fusion considers the product of failure probabilities, it has the semantics of a noisy-OR model (Henrion, 1987); the predicted confidence is always as strong as the least confident expert, and when multiple experts agree the confidence increases sharply.

A number of techniques are somewhat orthogonal to the problems addressed here. Cost-sensitive learning (*e.g.*, Elkan (2001)) allows one to emphasize certain errors, for example on certain types of instances or classes. In principle, it can lead the learner to focus on improving part of the precision-recall curve. In our case, we seek to minimize false-positive errors, but at high ranks. If formulated naively, this would lead to weighting or supersampling the negative instances. However, negative instances are already a large majority in many applications, as is the case in our experiments, and thus weighting them more is unlikely to improve performance significantly. It has been found that changing the balance of negative and positive classifier had little effect on the learned classifier (in that case, decision trees

[18] We also note that in much of past work, the classifier outputs are not calibrated probabilities.

and naive Bayes) (Elkan, 2001). Other work mostly focuses on oversampling the positives or downsampling the negatives (*e.g.*, Batista et al. (2004)). Area under curve (AUC) optimization is a related technique for improved ranking, though the techniques may be more appropriate for improving measures such as max F1, and we are not aware of algorithms that substantially improve at very high precision over standard learning technique (*e.g.*, see Cortez and Mohri (2004); Calders and Jaroszewicz (2007)).

5. Summary

Fusing classifiers trained on different sources of evidence, via a Noisy-OR model, increases recall at high precisions. When one seeks robust classifier probabilities, or in a threshold that achieves high precision, one can substantially save on labeling held-out data, compared to the standard way of verifying high precision. In such nearly-independent cases, the probability of a joint false-positive is close to the product of individual (conditional) false-positive probabilities, therefore an instance receiving high probabilities from multiple classifiers is highly likely a true positive. This property also partly explains our observation that simply summing the base classifier probabilities does very well when the objective is high precision at top rankings. As the number of classifiers increase, addressing the interdependencies of classifier outputs via a learning (stacking) approach could become beneficial. We showed promising results in that direction. Exploring the multiclass case and developing further understanding of the tradeoffs between early and late fusion are fruitful future directions.

Acknowledgments

Many thanks to Kevin Murphy, Emre Sargin, Fernando Preira, Yoram Singer, and the perception group for discussions and pointers, and to the anonymous reviewers for their valuable feedback.

References

G. E. Batista, R. C. Prati, and M. C. Monard. A study of the behavior of several methods for balancing machine learning training data. *SIGKDD Explorations*, 2004.

D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3, 2003.

A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *COLT: Proceedings of the Workshop on Computational Learning Theory*, pages 91–100, 1998.

G. Brown. An information theoretic perspective on multiple classifier systems. 2009.

T. Calders and S. Jaroszewicz. Efficient AUC optimization for classification. In *Proceedings of the 11th European conference on principles and practice of knowledge discovery in databases (PKDD)*, 2007.

C. Cortez and M. Mohri. AUC optimization vs. error rate minimization. In *Advances in Neuram Information Processing Systems (NIPS)*, 2004.

K. Crammer, O Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer. Online passive-aggressive algorithms. *Journal of Machine Learning Research*, 7, 2006.

C. Elkan. The foundations of cost-sensitive learning. In *IJCAI*, 2001.

P. Gehler and S. Nowozin. On feature combination for multiclass object classification. In *ICCV*, 2009.

Max Henrion. Practical issues in constructing a Bayes' belief network. In *Proceedings of the Third Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-87)*, pages 132–139, New York, NY, 1987. Elsevier Science.

G.E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002.

T. K. Ho, J.J. Hull, and S. N. Srihari. Decision combination in multiple classifier systems. *Pattern Analysis and Machine Intelligence*, 16, 1994.

A. Jain, K. Nandakumara, and A. Ross. Score normalization in multimodal biometric systems. *J. of Pattern Recognition Society*, 38, 2005.

J. Kittler, M. Hatef, R. Duin, and J. Matas. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):226–239, 1998.

P. Long, V. Varadan, S. Gilman, M. Treshock, and R. A. Servedio. Unsupervised evidence integration. In *ICML*, 2005.

C. D. Manning and H. Schutze. *Foundations of Statistical Natural Language Processing.* The MIT Press, 1999.

A. Niculescu-Mizil and R. Caruana. Predicting good probabilities with supervised learning. In *ICML*, 2005.

J. Platt. Probabilities for support vector machines and comparisons to regularized likelihood methods. In A. Smola, P. Bartlett, B. Schlkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*, pages 61–74. MIT Press, 1999.

C. Snoek, M. Worring, and A. Smeulders. Early versus late fusion in semantic video analysis. In *ACM Conference on Multimedia*, 2005.

G. Toderici, H. Aradhye, M. Pasca, L. Sbaiz, and J. Yagnik. Finding meaning on youtube: Tag recommendation and category discovery. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3447–3454. IEEE, 2010.

S. Tulyakov and V. Govindaraju. Using independence assumption to improve multimodal biometric fusion. *Lecture Notes in Computer Science*, 2005.

D. H. Wolpert. Stacked generalization. *Neural Networks*, 5, 1992.

B. Zadrozny and C. Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *KDD*, 2002.