

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

Omid Madani, Manfred Georg, and David Ross

Google Inc.
1600 Amphitheatre Parkway
Mountain View, CA 94043
{madani|mgeorg|dross}@google.com

Abstract

Consider learning tasks where the precision requirement is very high, for example a 99% precision requirement for a video classification application. 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 less – in some cases significantly less – than the product of individual probabilities of conditional false-positive mistakes (a Noisy-OR combination). Our analysis highlights a simple key criterion for this boosted precision phenomenon and justifies 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.

1 Introduction

In many classification scenarios, *e.g.*, in surveillance or in medical domains, 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), very

¹In binary classification, given a set of (test) instances, let T denote the set of truly 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.

high recall is required. In other applications, for instance for the safe application of a treatment or high quality user experience, 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, specifically 99%. This has applications to improved user experience and advertising, but can also benefit self-training (bootstrapping) systems during the automatic labeling of the unlabelled data, when a relatively low false-positive rate is sought. Achieving high precision raises a number of challenges: features may be too weak or the labels may be too noisy to allow the classifiers to robustly reach the required precision levels. Furthermore, verifying whether the classifier has achieved high precision can require expensive manual labeling.

Many applications, particularly in multimedia, provide diverse feature families and different ways of processing the different signals. For example, YouTube videos contain audio, video, and speech streams, as well as text-based (e.g., title, tags) attributes, and each such facet (or “view”) can be processed in distinct ways to create predictive features, such as color, texture, gradient and motion-related histogram features extracted from the visual signal. Given access to such rich set of 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 and then combining their output, aka late fusion² or fusion in classifier/semantic space [27]. 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 yield independent sources of evidence and complement one another. The pattern of false-positive errors that the base classifiers make, each trained on a single feature family, may therefore be nearly independent. Using an independence assumption on false-positive mistakes of the base classifiers and an additional positive correlation assumption, we derive a simple upper bound, essentially the product of individual conditional false-positive probabilities, via Bayes’

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

formula, on *joint* false-positive mistakes (in case of two classifiers, the event of both classifiers making a mistake, given both classify positive). This bound is equivalent to the Noisy-OR model [13]. Our subsequent analysis relaxes the assumptions and reveals a single alternative condition that needs to hold for the substantial drop in the probability of joint mistakes. Furthermore, such criteria 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 substantially increase at high precision levels via late fusion of nearly-independent base classifiers. We summarize our contributions as follows:

1. We report the phenomenon of boosted precision at the beginning of the precision-recall curve when combining independent feature families via late fusion.³ We present analyses that explain the observations and suggest ways for fusing classifiers as well as methods 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.

The paper is organized as follows. Section 2 contains our analysis and Section 3 presents experiments. Section 4 discusses related work, and Section 5 concludes. This paper extends our prior work [20], in particular with additional experiments on fusion of classifiers trained on subfamilies of audio and visual features, and experiments on a multiview paper-classification dataset [23].

2 Analyzing Fusion Based on False-Positive Independence

We focus on the binary classification setting and on the two-classifier case for the most part. 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 the probability of false-positive events. To show that the probability of joint false-positive mistakes can be significantly reduced when different feature families are available, in addition to an independence assumption on (conditional) false-positive events, we need another positive-correlation criterion (see below). Our first analysis uses these two assumptions, and derives an upper bound on the probability of joint false-positive mistake. This bound is equivalent to the Noisy-OR model [13]. We then discuss these assumptions, and subsequently present a relaxation that yields a single intuitive criterion for significant reduction in false positive probability.

The two assumptions are:

³In 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

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 (“fires”), 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 (“misfires”), and $(y_x = 0, f_i(x) = 1)$ is the conjunction of two events (the true class is negative, while $f_i(x) = 1$). We note that the assumption of the independence of conditional false-positive mistakes is a subset (milder form) of the full “view-independence” assumptions made in the original co-training work [4] (see related work, Section 4).

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

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

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

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

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

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). Extension to $k > 2$ classifiers is straightforward generalization of the two classifier case, making use of the two assumptions: 1) Independence of false positive mistakes: $\forall k \geq 2, P(f_k(x) = 1|y_x = 0, \wedge_{i < k} f_i(x) = 1) = P(f_k(x) = 1|y_x = 0)$, and 2) positive correlation: $\forall k \geq 2, P(f_k(x) = 1| \wedge_{i < k} f_i(x) = 1) \geq P(f_k(x) = 1)$:

$$P(y_x = 0| \wedge_{i \geq 1} f_i(x) = 1) \leq P(y_x = 0)^{-(k-1)} \prod_i (1 - P_i),$$

The bound has the form of a Noisy-OR model, where the prior probability is the “leak” probability.

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% false-positive error for each of two classifiers, to 1% for the combination. This (potential) near-geometric reduction in false-positive probability is at the core of the possibility of substantial increase in precision, via late fusion in particular. In our setting, we seek such boosts in precision specially for relatively high probability ranges. Our focus in this work is on 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 the two classifiers’ outputs are mutually exclusive (e.g., when 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 other. 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 ambiguities or similarities among instances in one feature domain that tend to 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, and the presence of these terms can lead to confusion by a text-based classifier between videos about the movie versus the game. However it is easy to tell such videos apart by the visual appearance or the audio. There can of course be exceptions. Consider the task of learning to distinguish two games in a video game series (such as “Uncharted 2” and “Uncharted 3”), and more generally, 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:

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

$$P(y_x = 0 | f_2(x) = 1, f_1(x) = 1) = \frac{r_{fp}}{r_p} (1 - P_2)(1 - P_1). \quad (5)$$

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 upper bound when $r \leq 1$. The ratios r_p and r_{fp} can be rewritten in conditional form⁴ as:

$$r_p = \frac{P(f_2(x) = 1 | f_1(x) = 1)}{P(f_2(x) = 1)}, r_{fp} = \frac{P(f_2(x) = 1, y_x = 0 | f_1(x) = 1, y_x = 0)}{P(f_2(x) = 1, y_x = 0)} \quad (6)$$

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 3.5 and 3.7). In our experiments we observe that indeed, often, $r_{fp} > 1$. This implies that false-positive events are *NOT* necessarily independent (in particular when $P(y_x = 0) \approx 1$, see inequality 4), even for very different feature families. However, we also observe that $r_p > r_{fp}$. 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 may yield high precision at top rankings as well. Simple aggregation techniques are competitive in a variety of tasks [28, 11, 9, 17]. We also experiment with the fusion-via-summation technique.

2.2.1 Conditions for (a Mild) Boost in Confidence (Lift in Precision-Recall Curve)

A natural question is whether it is always the case that precision (confidence) tends to increase, or $P(y_x = 1 | f_1(x) = f_2(x) = 1) \geq \max(P_1, P_2)$, even if not substantially, given the mild easily

⁴The ratios can be seen as essentially the pointwise mutual information quantities (without the log function) [22].

understandable assumptions that the classifiers are positively correlated, and that both somewhat agree⁵ on x being positive (i.e., $P_i \gg P(y_x = 1)$). This can help us understand when late fusion can help lift the precision-recall curve, albeit modestly, even when the feature families or classifiers are similar (such as two different classifiers trained on the same feature family). In particular, assume $P_1 \geq P_2$, and we want to understand when it is the case that: $P(y_x = 0 | f_2(x) = 1, f_1(x) = 1) \leq 1 - P_1$ (or, using equation 5, $\frac{r_{fp}}{r_p}(1 - P_1)(1 - P_2) \leq 1 - P_1$). Simplifying, we get whenever $1 - P_2 \leq \frac{r_p}{r_{fp}}$. As expected, because $P_2 \geq 0$, this is a weaker condition than the condition for the product upper bound to hold (which requires $1 \leq \frac{r_p}{r_{fp}}$). However, a remaining question is whether we can replace the extra dependence on false-positive correlation r_{fp} with the very mild positive correlation assumption, $r_p \geq 1$ (together with the assumption that the two classifiers agree). In the extreme case that the classifiers are duplicates, it can be verified that the condition holds, i.e., $1 - P_2 = \frac{r_p}{r_{fp}}$ (as expected, the overall confidence stays the same). But in intermediate cases, we find that counter examples exist, i.e., the overall confidence can in fact degrade (lower than the minimum of P_1 and P_2). Imagine $P(y_x = 1) = 0.01$ (generally, very low positive-prior), and $P(f_2(x) = 1 | f_1(x) = 1) = 0.5$ thus there is positive correlation, and $P_i = 0.5$ (conditional false-positive rate of 50%), but that $P(y_x = 0 | f_1(x) = f_2(x) = 1) = 1$ (their “intersection”, when both output positive, are exactly the two classifiers’ false positives). Therefore, it appears that we still need to take into account the false-positive correlation ratio r_{fp} (and in particular the ratio: r_p/r_{fp}).

3 Experiments

We focus on video classification, where the problem is classifying whether a video depicts mostly gameplay footage of a particular video game.⁶ We also include brief experiments, in Section 3.9, on the Cora dataset, which is a text (research paper) classification dataset enjoying multiple views [23].

Our objective here is to maximize recall at a very high precision, such as 99%. For evaluation and comparison, 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 decision theoretic scenarios where the system, once deployed, should make binary (committed) decisions or provide good probabilities on each instance (irrespective of other instances). We begin by describing the experimental setting, then provide comparisons under the two evaluations. Most of our experiments focus on visual and audio feature families. We report on the extent of dependencies among the two, and present some results that include other feature families (text), as well as sub-families of audio and visual features, and explore several variants of stacking.

⁵In the experiments, we will observe that when P_1 is very low (e.g., $P_1 \leq P(y_x = 1)$), while $P_2 \gg P(y_x = 1)$ i.e., when the two classifiers *disagree* on x , the overall joint confidence can be lower than the max P_2 (and $r_p < r_{fp}$).

⁶These “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.

For the video experiments in this paper, we chose 30 game titles at random, from amongst the more popular games. We treat each game classification as a binary 1-vs-rest problem. For each game, we collected around 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 an extra 30,000 videos from other game titles to serve as negatives for all 30 labels. The data, of about 120,000 instances, was split into 80% training, 10% validation, and 10% test.

3.1 Video Features and Classifiers

The video content features used span several different feature families, both audio (Audio Spectrogram, Volume features, Mel Frequency, ..) and visual features (Global visual features such as 8x8 hue-saturation, and PCA of patches at spatio-temporal interest points, etc.) [31, 34, 19, 29]. 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. 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 feature vector is fairly dense (only about 50% are zero-valued). We also include experiments with two text-based feature families, which we describe in Section 3.6.

We used the passive-aggressive online algorithm as the learner [8]. 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 F1,⁷ on validation data for the classifier trained on *all features* (audio and visual) appended together. This is our *early fusion* approach. We call this classifier Append. The parameters were 7 passes, and aggressiveness of 0.1, though the differences, *e.g.*, between aggressiveness of 1 and 0.01 were negligible at Max F1 0.774 and 0.778 respectively. 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 algorithms to improve the base classifiers.

We have experimented with 2 basic types of late fusion: 1) fusion using the bound 4 of Section 2 (NoisyOR), where false-positive probability is simply the product of the false-positive probabilities of base classifiers, *i.e.*, the Noisy-OR combination, and 2) fusion using the average of base classifier probability scores (AVG). For NoisyOR, we set the negative prior $P(y_x = 0) = 0.97$, since the

⁷F1 is the harmonic mean of precision and recall. The maximum is taken over the curve for each problem.

positives, for each label, are roughly 3% of data.⁸ In Section 3.8, we also report on learning a weighting on the output of each classifier (stacking), and we describe another stacking variant, NoisyOR Adaptive, as well a simpler hybrid technique, NoisyOR+AVG in Section 3.7.

3.2 Events Definitions and Score Calibration

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.*, [24, 35, 20]). 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 [25]. We instead used the simple non-parametric technique of binning (pooling) 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. Our binning technique is a variant of the (pool-adjacent violators) PAV algorithm for isotonic regression [26, 36]. Briefly, instances are processed by classifier score from highest to lowest, and bins are created when at least 20 instances are inside a bin, and there is at least one positive and one negative instance inside a bin (except for the lowest bin which may have only negatives). The minimum-size condition controls for statistical significance, and the latter condition ensures that the probability estimates for the high scoring ranges are somewhat conservative. Repeatedly, pairs of adjacent bins that violate the monotonicity condition are then merged. Note that in typical isotonic regression, initially each bin contains a single point, which can lead to the last (highest) bin with 1.0 positive proportion or a very high probability estimate. Figure 1 shows the mapping for one classifier, for plain isotonic regression and our parameter setting in this paper (minimum bin size set to 20, and the diversity constraint). The main significant difference tends to be at the top of the probability range.

3.3 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, NoisyOR and AVG. Figure 2 shows the precision-recall curves for a few classifiers on one problem. We observe that late fusion substantially improves performance (“lifts” the curve up) at the high precision regions

⁸The prior does not change the ranking experiments (Table 1), but affects threshold selection (Table 2).

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

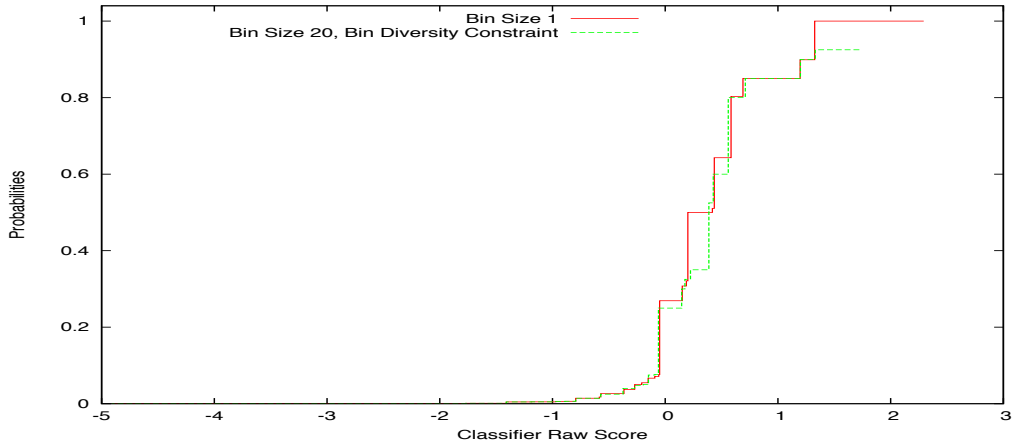


Figure 1: The score to probability mapping learned for one classifier using two parameter settings, one typical isotonic regression (minimum bin size of 1), and another requiring a minimum bin size of 20 and both negative and positive instances within a bin (the diversity constraint), as a simple way to obtain more conservative probability estimates.

of the curve. Note 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 boosting precision (see also the comparisons of [11]). 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 major 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 $\text{rec}@99$ for recall at 99% precision. When we pair the $\text{rec}@99$ values for

Prec. \rightarrow	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
NoisyOR	0.33	0.55	0.66	0.82
AVG (SUM)	0.45	0.62	0.70	0.82

Table 1: Ranking performance, *i.e.*, recall at several precision thresholds (averaged over 30 classes), on the test set ($\text{rec}@99$, $\text{rec}@95$, etc).

each problem, at the 99% precision threshold, AVG beats all other methods above it in the table, and NoisyOR beats Append and the base classifiers (at 99% confidence level). As we lower the precision threshold or when we compare Max F1 scores, the improvements from late fusion decrease.

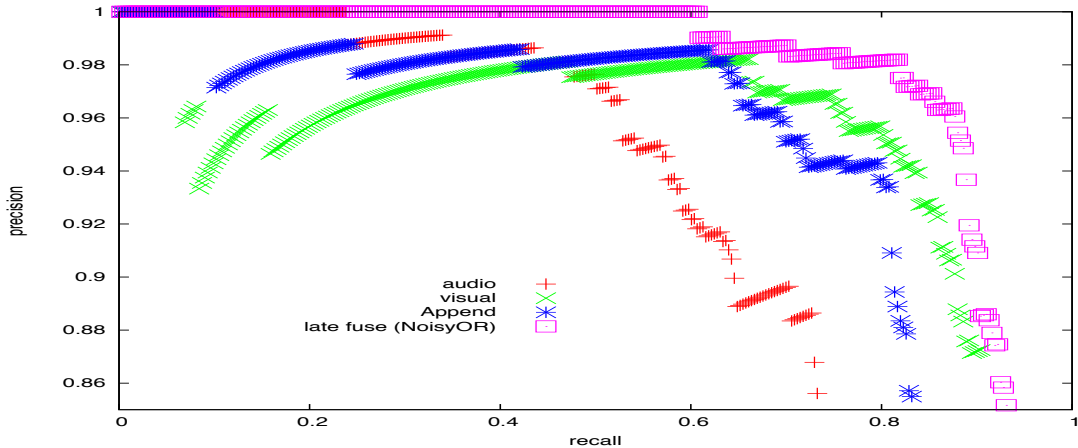


Figure 2: Precision vs. recall curves, on one of the 30 game classes, for the classifiers trained on visual only, audio only, the union of the two features (Append), and late fusion. Fusion substantially increases recall at high precisions.

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 extremely well, so there would be room for improvement. Figure 3 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

3.4 Threshold Picked using Validation Data

We now focus on the setting where a threshold should be picked using the validation data, *i.e.*, the classifier has to decide on the class of each instance in isolation during testing. Table 2 presents the findings. In contrast to Table 1, in which the best threshold was picked on test instances, here, we assess how the probabilities learned on validation data “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 (Section 3.2). Therefore

¹⁰An interesting trend appears to be that Append⁺ (early fusion) gains an advantage when the performances of one feature family dominates the other (high x values).

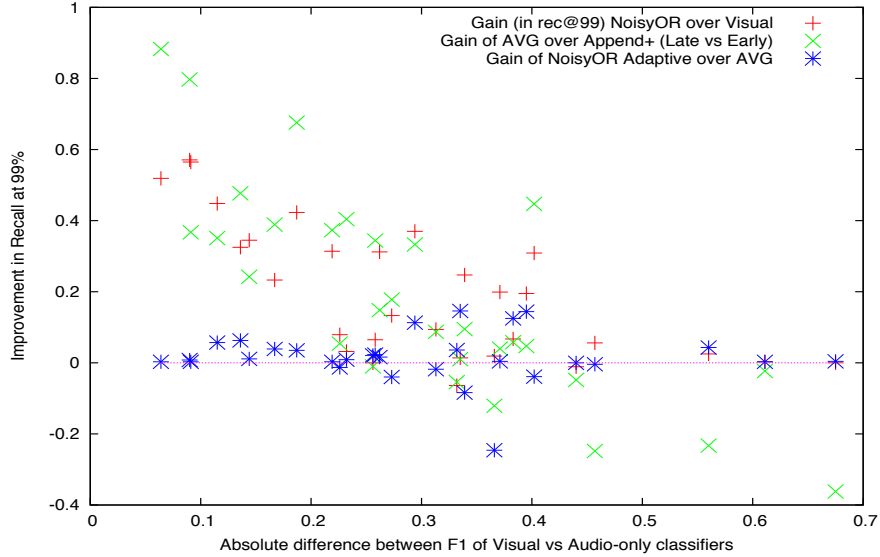


Figure 3: Each point corresponds to one problem. The x-coordinate for all points is the absolute difference in max F1 performance of audio and visual-only base classifiers. For the first two plots, the y-coordinate is the gain, *i.e.*, the difference in recall at 99% (rec@99) when instances are ranked. The first plot shows the gains of NoisyOR (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 NoisyOR Adaptive (Section 3.7) 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.

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 “passing” problems (out of 30), *i.e.*, those for which $|E_\tau| > 0$ (the set is not empty). For such passing problems, let E_τ^p denote the number of (true) positive instances in E_τ . The second number in the triple is number of “valid” problems, *i.e.*, those for which $\frac{|E_\tau^p|}{|E_\tau|} \geq \tau$ (the ratio of positives is greater than desired threshold τ).

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

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

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 , averaged over valid problems only.

Threshold $\tau \rightarrow$	≥ 0.99	≥ 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)
NoisyOR	(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)

Table 2: For each classifier and threshold combination (threshold picked using validation data), we report three numbers: The number of “passing” problems (out of 30), where some test instances obtained a probability no less than the threshold τ , the number of “valid” problems, *i.e.*, those passing problems for which the ratio of (ture) positive test instances 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 if we average the recall over all problems, at $\tau = 0.99$ Append⁺ gets 0.06 (*i.e.*, $0.6 \times \frac{3}{30.0}$, since Append⁺ achieves 3 valid problems), while NoisyOR and AVG get respectively 0.21 and 0.26. Both the number of valid problems and recall are indicative of performance.

Fusion using NoisyOR 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. NoisyOR 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 NoisyOR variants reach 0.99, calibrated AVG yields a substantially higher recall. NoisyOR 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 NoisyOR (see also NoisyOR+AVG in Section 3.7). We note that in practice, with many 100s to 1000s of classes, validation data may not provide sufficient evidence that AVG reaches 99% (in general, a high precision), and NoisyOR can be superior.

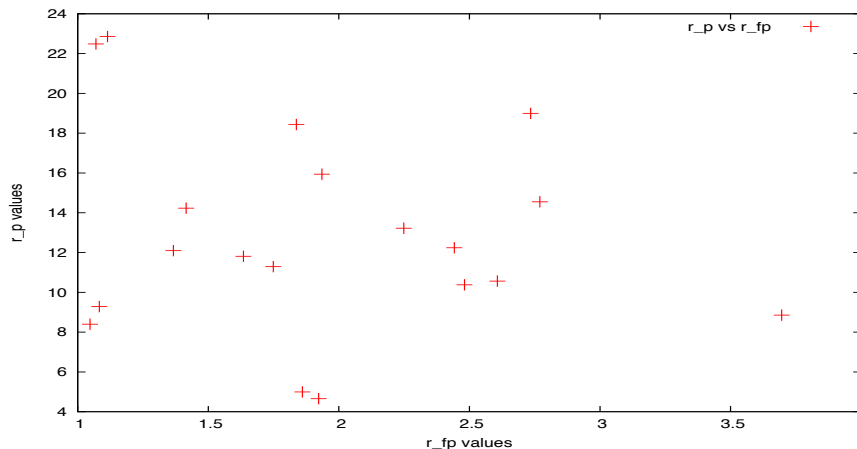


Figure 4: The positive correlation (good) ratios, r_p (y axis), versus dependency ratios r_{fp} , on 19 games, for threshold $\tau = 0.2$ (see Section 3.5), measured on test ($f_i(x) = 1$ if $P_i \geq \tau$). Note that for all the problems, the bad-to-good ratio $r_{fp}/r_p < 1$.

3.5 Score Spread and Dependencies

For a choice of threshold τ , let the event $f_i(x) = 1$ mean that the score of classifier i exceeds that threshold (the classifier outputs positive or “fires”). For assessing extent of positive correlation, we looked at the ratios r_p (eq. 6, 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 4 shows the spread for $\tau = 0.2$. We also looked at false-positive dependence and in particular 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 derivations of Section 2.2: Even though 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, that the NULL hypothesis could be rejected. However, r_{fp} is in general relatively small, and $r_p \geq r_{fp}$ for all the problems and thresholds $\tau \geq 0.1$ that we looked at. Note that the choice of threshold that determines the event (when the rule fires), makes a difference in the bad-to-good ratios (see Section 3.7).

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 performance verification, we require to sample and label y/x many positive

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

PREC. \rightarrow	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	0.73	0.85	0.92	0.95
LDA+AUDIO+VISUAL	0.76	0.88	0.94	0.95

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.

instances for the validation data. In our game classification experiments, we saw that base classifiers’ $\text{rec}@99$ were rather low (around 10 to 15% on test data from 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 level on more than a majority of the problems (Table 2).

3.6 Text-Based Features and Further Exploration of Dependencies

Our training data comes from title matches, thus we expect classifiers based on text features to do rather well. Here, as features, we used a 1000-topic Latent Dirichlet Allocation (LDA) model [3], 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 NoisyOR). We observe LDA alone does very well (noting 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,¹³ yielding an average Max 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. This is not surprising, as the instances LDA was trained on contained words from tags and description.¹⁴ We also compared pairs of feature subfamilies from either visual or audio features respectively. The bad to good ratios remained less than one (for $\tau = 0.1$). The table includes the ratios for video HOG (histogram of gradients) and motion histogram subfamilies.

¹³Note that combining these classifiers is still potentially useful to increase the coverage. Only a fraction of game videos’s titles contain the game titles.

¹⁴We later compared tag-only and description-only classifiers and observed $r_{fp}/r_p \leq 1$, even though both are based on bag-of-words text features.

PAIR →	LDAvsTAG	LDAvsVIS	TAGvsVIS	VISvsAUDIO	HOGvsMOTIONHIST
r_{fp}	101	6	3	2	2.7
r_p	30	18	17	14	4.5

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.

3.7 Improved NoisyOR: Independence as a Function of Scores

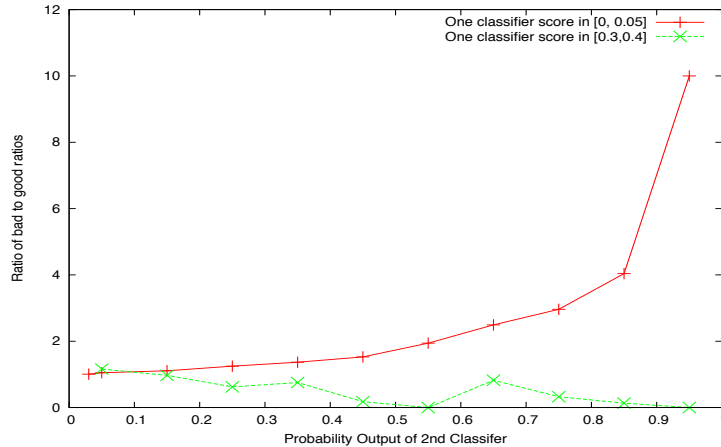


Figure 5: The bad-to-good ratio r as a function of individual classifier output-probability ranges. When the classifiers ‘disagree’ (one output is near the positive prior, 0.03, while the other is higher), $r \gg 1$. But $r \approx 1$, or $r \ll 1$, when both ‘agree’, *i.e.*, when both outputs are higher than the positive prior (lower curve).

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 than the prior. Figure 5 shows two slices of the two-dimensional surface learned by averaging the ratios over the grid of two classifier probability outputs, over the 30 games. These ratios are used by NoisyOR 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 NoisyOR technique when one classifier's probability is close to the prior. In NoisyOR+AVG, when one classifier outputs below 0.05 (close to the prior), we simply use the average score. As seen in Tables 5 and 6, its performance matches or is superior to the best of NoisyOR 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 NoisyOR+AVG. The performance of NoisyOR Adaptive indicates that learning has potential to significantly improve over the simpler techniques.¹⁶

Table 5: Ranking performance experiments (Table 1) using NoisyOR+AVG and NoisyOR Adaptive. The rows for (plain) NoisyOR and AVG are copied from Table 1 for ease of comparison.

PREC. \rightarrow	99%	95%	90%	MAX F1
AVG (SUM)	0.45	0.62	0.70	0.82
NOISYOR+AVG	0.45	0.62	0.72	0.83
NOISYOR ADAPTIVE	0.47	0.65	0.72	0.83

THRESHOLD $\tau \rightarrow$	≥ 0.99	≥ 0.95
NOISYOR	(24, 18, 0.35)	(29, 22, 0.56)
CALIBRATED AVG	(17, 12, 0.65)	(30, 26, 0.62)
NOISYOR+AVG	(24, 22, 0.322)	(28, 26, 0.45)
NOISYOR ADAPTIVE	(29, 22, 0.43)	(30, 25, 0.59)

Table 6: Threshold experiments (Table 2) repeated for NoisyOR+AVG and NoisyOR Adaptive. The rows for (plain) NoisyOR and calibrated AVG are copied from Table 2 for ease of comparison.

3.8 Learning a Weighting (Stacking)

We can take a stacking approach [33] and learn on top of classifier outputs and other features derived from them. We evaluated a variety of learning algorithms (linear SVMs, perceptrons, decision

¹⁵Given p_1 and p_2 , the map is used to obtain $r_{p_1 p_2}$, and the product $r_{p_1 p_2}(1 - p_1)(1 - p_2)$ 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.

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

trees, and random forests), comparing Max F1 and rec@99. On each instance, we used as features the probability output of 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 $C = 0.1, 1, 10, \text{ and } 100$, 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 roughly 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.¹⁷ 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, as we observe in the next section, with additional base classifiers, stacking can show a convincing advantage for further boosting precision.

3.8.1 Late Fusing Classifiers Trained on Subfamilies

There are several feature subfamilies within Audio and Visual features. A basic question is whether training individual classifiers on each family separately (14 classifiers), then calibrating and fusing the output, can further boost precision. As we split the features, individual classifiers get weaker, but their fusion may more than make up for the lost ground. In particular, we observed in Section 3.6 that the bad-to-good ratios for each subfamily pair were lower than 1 for the pairs we checked, indicating the potential for precision boost. For training the 14 classifiers, we used the same algorithm with exact parameters as above (7 passes of passive-aggressive). Calibration of the classifiers was performed on all of validation data, as before. We used 2-fold validation on the validation data for parameter selection for several stacking algorithms we tested, as in the previous section (random forests, linear SVMs, committees of perceptrons). The features are the outputs of the 14 classifiers (probabilities) on each instance. For SUM (simply sum the feature values, akin to AVG), SVMs, and perceptrons (but not random forests), we found that including the products of pairs and triples of outputs as extra features was very useful. For efficiency, we kept a product feature for an instance as long as the value passed a minimum threshold of 0.001. Both on the 2-fold validation data, and on test data, random forest of 200 trees performed best in achieving a rec@99 of 0.53 on test. Max F1 did not noticeably improve (compared to using two classifiers). Table 7 presents the performance results. The superior performance of random forests on rec@99, compared to SVMs, perceptron committees, and fusing two classifiers (e.g., AVG) is statistically significant using a paired sign test (e.g., 21 wins vs. 9 losses when comparing to SVMs).

¹⁷Even using only p_1 and p_2 as features gives a slight improvement in Max F1 over simple averaging, but using all the features gives noticeable additional improvement.

PREC. \rightarrow	99%	95%	90%	MAX F1
RANDOM FORESTS (200 TREES)	0.53	0.65	0.73	0.84
PERCEPTRON COMMITTEE (40)	0.50	0.64	0.72	0.83
LINEAR SVMs, $C = 10$	0.49	0.63	0.72	0.83
SUM	0.47	0.59	0.67	80

Table 7: Average ranking test performance (over 30 classes), when late fusing individual classifiers trained on sub-feature families of Audio and Visual features (14 many subfamilies), where late fusion is achieved by learning on the validation data (no learning for SUM). We observe a significant boost in rec@99, in particular via random forests.

3.9 Analysis on the Cora dataset

The Cora Research Paper Classification dataset consists of about 31k research papers, where each paper is described by a number of views, including author names, title, abstract, and papers cited [23]. Each paper is classified into one of 11 high level subject categories (Artificial Intelligence, Information Retrieval, Operating Systems,...). We used two views, author and citations, and partitioned the data into a 70-15-15 train-validation-test split. Each paper has on average 2.5 authors and 21 citations. We trained and calibrated the scores of linear SVM classifiers (trained on each view separately and on both appended), using the best parameter $C = 100$ for early fusion, after trying $C \in \{1, 10, 100\}$ on validation (all had close performance). Same C was used for single-view classifiers.

We expect the authors and citations views to be roughly independent, but exceptions include papers that cross two (or more) fields (e.g., both Artificial Intelligence and Information Retrieval): the citations may include papers crossing both fields and the authors may also have published papers in both. Table 8 presents the good (r_p) and bad ratios and ranking performances for a few algorithms. The median bad-to-good ratio slightly exceeds 1 (it is 1.2). Thus we observed weaker patterns of independence compared to the video data, but the near 1 ratios suggest that late fusion techniques such as AVG and NoisyOR+AVG should still perform relatively well at high precision requirements, as seen in Table 8. Note that the positive proportion of the various classes is high compared to the video dataset, therefore, considering inequality 4, the factor $P(y_x = 0)^{-1}$ can be high (1.5 for AI and ≈ 1.1 for several other classes).

4 Related Work

The literature on benefits of multiple views, multi-classifier systems (ensembles), and fusion, applied to a variety of tasks, is vast [12, 15, 4, 16, 18, 27, 5, 11, 37]. The work of Kittler *et. al.* explores a number of classifier combination techniques [17]. There are several differences between

Table 8: The Cora Research Paper Classification dataset. Left: The good (r_p) and bad ratios (for $\tau = 2$), using the two linear SVM classifiers trained on Citations or Authors only for the 11 top level classes. The percentage of positive instances is shown in parentheses for each class. Right: Ranking performance (recall at two precision thresholds and Max F1), using SVM classifiers, averaged over the 11 problems.

	r_p	r_{fp}		r_p	r_{fp}	Prec. \rightarrow	99%	95%	Max F1
AI (35%)	1.7	3.3	HW Arch (4%)	10	11	Author Only	0.02	0.03	0.54
IR (2%)	17	9.2	Theory (10%)	5.5	8	Citations Only	0.03	0.13	0.71
DB (4%)	11	15	Prog. (13%)	4.5	6.7	Append	0.04	0.19	0.73
Encr. (4%)	12	10	HCI (5%)	11.1	11.7	AVG	0.09	0.19	0.73
OS (8%)	5.9	7	Data (8%)	8.1	8.7	NoisyOR	0.08	0.18	0.72
Netw. (5%)	6.1	6.9				NoisyOR+AVG	0.09	0.21	0.73

that work (and much related work) and ours: the treatment is for a more general setting where classifier outputs can be very correlated.¹⁸ Often other performance measures, such as average precision over the entire precision-recall curve, equal error rate, or max F1, are reported. We are not aware of work that 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 early in the precision-recall curve via late fusion.

Near-independence relates to classifier diversity, and these and close properties such as (dis)agreement rate, have been studied in work on classifier ensembles as well as co-training and semisupervised learning settings [12, 4, 1, 30, 21, 32]. The original view-independence assumptions in co-training [4] are strong, in that they assume conditional independence assumptions for all the possible combinations of class values and output values of the classifiers (similar to the Naive Bayes assumption). Abney describes an example scenario (two classifiers on a data set) where the classifier outputs remain correlated after conditioning on the class [1]. Later work has sought to relax the assumptions, and make them more realistic and directly relevant (*i.e.*, find sufficient and/or necessary conditions) to the success of co-training [1, 32]. For instance, Abney gives a condition for weak dependence (which remains a function of all possible class values), and shows that co-training can succeed if only weak dependence holds. Similarly, early work on ensembles pointed to their potential in reducing error (*e.g.*, via majority vote), by making ideal independence assumptions. Our initial analysis is similar in nature, but with our focus on conditional false-positive events, requires a milder independence assumption (plus an unconditional positive correlation) than full view independence.

¹⁸We also note that in much of past work on ensembles, the classifier outputs, even if they are interpretable as probabilities, are not carefully calibrated probabilities learnt from heldout data.

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. 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 [11, 28].

Fusion based on Noisy-OR variants has a similarity to the Product of Experts (PoE) in that it involves a product [14]. POE 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 NoisyOR fusion considers the product of failure probabilities, it is akin to a disjunction [13]; the predicted confidence is always as strong as the least confident expert, and when multiple experts agree the confidence increases sharply. The product rule for two classifiers is simply $p_1 p_2$, while the Noisy-OR is $p_1 + p_2 - p_1 p_2$ (ignoring the priors). Kittler et. al. [17] study the product rule and compare it to variants such as sum, and find sum to be more robust, due to the higher sensitivity of the product rule to variance in the output of the classifiers. We briefly experimented with ranking evaluation using the product rule (*i.e.*, the set up of Table 1). Recall at 99% precision was high, but a percent lower than AVG (SUM), and the average Max F1 was lower at 0.79 (several techniques obtain 0.82). Note that for the product rule to work well, in general the low probabilities need to be estimated fairly well too. For example, at an extreme, if very low scores, from one classifier, are rounded to 0 by the calibration technique, the probability output of the other classifier loses its influence completely (*e.g.*, on ranking such instances). Tamraker et. al. and Gehler et. al. find competitive results with late fusion using simple sum and product techniques [28, 11].

A number of techniques are somewhat orthogonal to the problems addressed here. Cost-sensitive learning allows one to emphasize certain errors [10], 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 observed that changing the balance of negative and positive classifier can have little effect on the learned classifier (in that work, decision trees and naive Bayes) [10]. Other work mostly focuses on oversampling the positives or downsampling the negatives (*e.g.*, [2]). 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 [7, 6]).

5 Summary

Fusing classifiers trained on different sources of evidence, via a Noisy-OR model and its extensions, can substantially increase recall at high precisions. When one seeks robust class probabil-

ities, or a threshold that achieves high precision, one can significantly save on manually labeling held-out data compared to the standard way of verifying high precision. For classifiers trained on very different features, we showed how the probability of a (conditional) joint false-positive can be upper bounded by the product of individual (conditional) false-positive probabilities, therefore, in such scenarios, 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, and other simple variants such as product, can do very well when the objective is improving precision at top rankings. As the number of classifiers grows, addressing the interdependencies of classifier outputs via a learning (stacking) approach becomes beneficial. We showed promising results in that direction. Investigating the multiclass case and developing a further understanding of the tradeoffs between early and late fusion are fruitful future directions.

Acknowledgements

Many thanks to Tomas Izzo, Kevin Murphy, Emre Sargin, Fernando Pereira, and Yoram Singer, for discussions and pointers, and to the anonymous reviewers for their valuable feedback.

References

- [1] S. Abney. Bootstrapping. In *40th Annual Meeting of the Association for Computational Linguistics*, 2002.
- [2] 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.
- [3] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3, 2003.
- [4] 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.
- [5] G. Brown. An information theoretic perspective on multiple classifier systems. 2009.
- [6] 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.
- [7] C. Cortez and M. Mohri. AUC optimization vs. error rate minimization. In *Advances in Neural Information Processing Systems (NIPS)*, 2004.

- [8] K. Crammer, O Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer. Online passive-aggressive algorithms. *Journal of Machine Learning Research*, 7, 2006.
- [9] V. Dani, O. Madani, D. Pennock, S. Sanghai, and B. Galebach. An empirical comparison of expert aggregation techniques. In *UAI*, 2006.
- [10] C. Elkan. The foundations of cost-sensitive learning. In *IJCAI*, 2001.
- [11] P. Gehler and S. Nowozin. On feature combination for multiclass object classification. In *ICCV*, 2009.
- [12] L. K. Hansen and P. Salamon. Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1990.
- [13] M. 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.
- [14] G. E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002.
- [15] T. K. Ho, J.J. Hull, and S. N. Srihari. Decision combination in multiple classifier systems. *Pattern Analysis and Machine Intelligence*, 16, 1994.
- [16] A. Jain, K. Nandakumara, and A. Ross. Score normalization in multimodal biometric systems. *J. of Pattern Recognition Society*, 38, 2005.
- [17] 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.
- [18] P. Long, V. Varadan, S. Gilman, M. Treshock, and R. A. Servedio. Unsupervised evidence integration. In *ICML*, 2005.
- [19] R. F. Lyon, M. Rehn, S. Bengio, T. C. Walters, and G. Chechik. Sound retrieval and ranking using sparse auditory representations. *Neural computation*, 22(9):2390–2416, 2010.
- [20] O. Madani, M. Georg, and D. A. Ross. On using nearly-independent feature families for high precision and confidence. In *Asian Machine Learning Conference*, 2012.
- [21] O. Madani, D. Pennock, and G. Flake. Co-validation: Using model disagreement on unlabeled data to validate classification algorithms. In *NIPS*, 2004.
- [22] C. D. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. The MIT Press, 1999.

- [23] A. McCallum, K. Nigam, J. Rennie, and K. Seymore. Automating the construction of internet portals with machine learning. *Information Retrieval Journal*, 3, 2000. www.research.whizbang.com/data.
- [24] A. Niculescu-Mizil and R. Caruana. Predicting good probabilities with supervised learning. In *ICML*, 2005.
- [25] 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.
- [26] T. Robertson, F. Wright, and R. Dykstra. *Order Restricted Statistical Inference*. John Wiley & Sons, 1988.
- [27] C. Snoek, M. Worring, and A. Smeulders. Early versus late fusion in semantic video analysis. In *ACM Conference on Multimedia*, 2005.
- [28] A. Tamrakar, S. Ali, Q. Yu, J. Liu, O. Javed, A. Divakaran, H. Cheng, and H. S. Sawhney. Evaluation of low-level features and their combinations for complex event detection in open source videos. In *CVPR*, 2012.
- [29] 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)*, pages 3447–3454. IEEE, 2010.
- [30] S. Tulyakov and V. Govindaraju. Using independence assumption to improve multimodal biometric fusion. *Lecture Notes in Computer Science*, 2005.
- [31] T. C. Walters, D. A. Ross, and R. F. Lyon. The intervalgram: An audio feature for large-scale melody recognition. In *Proc. of the 9th International Symposium on Computer Music Modeling and Retrieval (CMMR)*, 2012.
- [32] W. Wang and Z.-H. Zhou. A new analysis of co-training. In *ICML*, 2010.
- [33] D. H. Wolpert. Stacked generalization. *Neural Networks*, 5, 1992.
- [34] W. Yang and G. Toderici. Discriminative tag learning on YouTube videos with latent sub-tags. In *Computer Vision and Pattern Recognition (CVPR)*, 2011.
- [35] B. Zadrozny and C. Elkan. Learning and making decisions when costs and probabilities are both unknown. In *KDD*, 2001.
- [36] B. Zadrozny and C. Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *KDD*, 2002.

[37] Z.-H. Zhou. *Ensemble Methods: Foundations and Algorithms*. Chapman & Hall, 2012.