A Continuum from Mixtures to Products: Aggregation under Bias

Amos J. Storkey
Zhanxing Zhu
Jinli Hu
School of Informatics, University of Edinburgh

Abstract

This is a preprint, and does not constitute publication, but is provided for the benefit of attendees to accompany a talk at the ICML Workshop on Divergence Methods for Probabilistic Inference. If you wish to reference this paper, please reference the final published version.

Machine learning models rely heavily on two compositional methods: mixtures and products. Probabilistic aggregation also commonly uses forms of linear opinion pools (which are effectively mixtures), or log opinion pools (which are effectively products). In this paper, we introduce a complete spectrum of compositional methods, Rényi mixtures, that interpolate between mixture models and product models, and hence between log opinion pools and linear opinion pools. We show that these compositional methods are maximum entropy distributions for aggregating information from agents subject to individual biases, with the Rényi divergence parameter dependent on the bias. We also demonstrate practically that Rényi mixtures can provide better performance than log and linear opinion pools, with the optimal limit of log opinion pools when all agents are unbiased and see the same data. We infer that log opinion pools are the most appropriate aggregator for machine learning competitions. We designed, ran and analysed a machine learning Kaggle competition, the results of which confirmed this expectation. Finally we relate Rényi mixtures to recent work on machine learning markets, showing that Rényi aggregators are directly implemented by machine learning markets with isoelastic utilities, and so can result from autonomous self interested decision making by individuals contributing different predictors.

1. Introduction

Aggregation of predictions from different agents or algorithms is becoming increasingly necessary in distributed, large scale or crowdsourced systems. Much previous focus is on aggregation of classifiers or point predictions. However, aggregation of probabilistic predictions is also of particular importance, especially where quantification of risk matters, generative models are required or where probabilistic information is critical for downstream analyses. In this paper we focus on aggregation of probability distributions (including conditional distributions).

Any practical aggregation method can be decomposed into two components: an aggregation model and an expert model. In expert systems (see e.g. (Jacobs et al., 1991; Hinton, 2002)) different model components are weighted differently for different covariate values. The choice of reweighting is given by the expert model, and there are many ways that this can be done. On the other hand the weighting given by an expert model says nothing about how items are aggregated given that weighting: an aggregation model is still needed. Any aggregation model can be combined with any expert model. Typically, though there are many forms of expert model, and choices of experts have received significant attention, aggregation models for distributions typically fall in two simple classes: linear and log opinion pools (including restrictions of these). This paper focuses on examining aggregation models.

The main novel contributions of this paper are

- Introducing the class of Rényi divergence based aggregators, and showing these aggregators interpolate the whole spectrum between linear opinion pools and log opinion pools.
- Giving maximum entropy arguments for the full class of Rényi divergence based aggregators in the context of different biases in the individual predictors.
- Empirically demonstrating the validity of Rényi divergence based aggregators on simulated systems and real data with sample selection bias.
- Building, and running a competition environment for
multi-class probabilistic prediction, and testing probabilistic aggregation methods on the submissions, demonstrating the superiority of log opinion pools in a real competition environment.

- Directly relating Machine Learning Markets and maximum-entropy aggregators.

As a result of this paper, we establish a practical continuum of aggregation methods between mixture models (linear opinion pools) and product models (log opinion pools), along with a demonstration that log opinion pooling is the most appropriate in a competition environment.

2. Background

Aggregation methods have been studied for some time, and have been discussed in a number of contexts. Aggregation methods differ from ensemble approaches (see e.g. (Dietterich, 2000)), as the latter also involves some control over the form of the individuals within the ensemble: with aggregation, the focus is entirely on the method of combination. In addition, most aggregation methods focus on aggregating hard predictions (classifications, mean predictive values etc.) (Breiman, 1996; Domingos, 1997). Some, but not all of those are suitable for aggregation of probabilistic predictions (Dani et al., 2006; Ottaviani & Sørensen, 2007), where full predictive distributions are given. This issue has received significant attention in the context of aggregating Bayesian or probabilistic beliefs (West, 1984; Dietterich, 2010; Maynard-Reid & Chajewska, 2001; Pennock & Wellman, 1997; Storkey, 2011). Full predictive distributions are generally useful for a Bayesian analysis (where the corresponding loss function is computed from the posterior predictive distribution), in situations where full risk computations must be done, or simply to get the most information from the individual algorithms. Wolpert (Wolpert, 1992) describes a general framework for aggregation, where an aggregator is trained using the individual predictions on a held out validation set as inputs, and the true validation targets as outputs. This requires specification of the aggregation function. The work in this paper fits within this framework, with Rényi mixtures as the aggregator. In crowdsourcing settings, issues of reliability in different contexts come into play. Log opinion pools have been generalized to weighted log opinion pools using Bayesian approaches with an event-specific prior (Kahn, 2004). This emphasises that expert models can work with aggregators at many different levels, from individual data points to whole datasets within a corpus.

Recently, prediction markets, and methods derived from securities market settings (Storkey, 2011; Storkey et al., 2012; Lay & Barbu, 2010; Barbu & Lay, 2011; Pennock & Wellman, 1997; Dani et al., 2006; Chen & Wortman Vaughan, 2010), have provided a particular foundation for belief aggregation. Belief aggregation of this form of importance in crowdsourcing settings, or settings combining information from different autonomous agents. In such settings, the beliefs of different agents can be subject to various biases.

One other area that aggregation has shown importance is in machine learning competitions, including the Netflix Challenge (Green, 2006), the PASCAL Visual Object Classes challenge (Everingham et al., 2006), and many challenges set in the Kaggle challenge environment (Goldbloom, 2010). Many workshops (e.g. KDD) also run a variety of machine learning challenges. One of the most consistent take-home messages from all the challenges is that aggregation of individual entries provides a performance benefit. The final winning Netflix submission was itself a large scale aggregation of 107 different methods (Robert M. Bell & Volinsky, 2010).

3. Weighted Divergence Aggregation

We consider a setting where there are $N_A$ agents, indexed by $i$ each with a distribution $P_i$, which is the agent belief. We consider aggregate distributions $P$ that pull together the beliefs of the different agents. Typically a small validation set can be made available for estimating aggregation parameters.

Weighted divergence-based aggregation was proposed in (Garg et al., 2004). The idea was, given individual distributions $P_i$, to choose an aggregate distribution $P$ given by

$$P = \arg \min_Q \sum_i w_i D(P_i, Q), \tag{1}$$

where $w_i$ is a weight and $D(P_i, Q)$ represents a choice of divergence between $P_i$ and $Q$, where $D(A, B) \geq 0$, with equality iff $A = B$. This framework generalizes several popular opinion pooling methods, e.g., linear opinion pooling when $D(P_i, Q) = KL(P_i || Q)$ is a KL divergence, and log opinion pooling when $D(P_i, Q) = KL(Q || P_i)$. Concretely, a linear opinion pool is given by $P(y|\cdot) = \sum_{j=1}^{N_A} w_j P_j(y|\cdot)$, where $w_j \geq 0 \forall j$ and $\sum_{j=1}^{N_A} w_j = 1$. The weight vector $w$ can be optimized by maximizing the log likelihood with simplex constraints, or alternatively via an expectation maximization procedure. By convexity, the solution of both approaches is equivalent.

On the other hand a logarithmic opinion pool is given by $P(y|\cdot) = \frac{1}{Z(w)} \prod_{j=1}^{N_A} P_j(y|\cdot)^{w_j}$ where $w_j \geq 0 \forall j$. The logarithmic opinion pool is more problematic to work with due to the required computation of the normalization constant, which is linear in the number of states. The value of $w$ can be obtained using a gradient-based optimizer. Others (see e.g. (Heskes, 1998)) have used various approximate
schemes for log opinion pools when the state space is a product space.

Weighted Divergence aggregation is very general but to make it practical use of these methods, we need to choose a particular form of divergence. In this paper we utilize the family of Rényi divergences for weighted divergence aggregation. This choice is motivated by a number of benefits we establish through this paper:

- The family of Rényi divergence aggregators still include linear opinion pools and log opinion pools as special cases, and interpolate between them with a single parameter.
- Rényi divergence aggregators satisfy maximum entropy arguments for the aggregator class under highly relevant assumptions about the biases of individual agents.
- Rényi divergence aggregators are implemented by machine learning markets, and hence can result from autonomous self interested decision making by the individuals contributing different predictors without centralized imposition. Hence this approach can incentivize agents to provide their best information for aggregation.

These diverse benefits of weighted Rényi divergence based aggregators make it a particular interesting form of aggregation methods. In much of the analysis that follows we will drop the conditioning (i.e. write $P(y)$ rather than $P(y|x)$) for the sake of clarity, but without loss of generality as all results follow through in the conditional setting.

3.1. Weighted Rényi Divergence Aggregation

The space of all possible weighted divergence aggregation methods is large, and for any particular choice of divergence there is the issue of how it should be implemented. Here we introduce an important simplified subspace, the family of Rényi divergence aggregators.

**Definition 1 (Rényi Divergence).** Let $y$ be a random variable taking values $y = 1, 2, \ldots, K$. The Rényi divergence of order $\gamma$ ($\gamma > 0$) from a distribution $P$ to a distribution $Q$ is defined\(^\dagger\) as

$$D_\gamma^R[P||Q] = \frac{1}{\gamma - 1} \log \left( \sum_{y=1}^{K} P(y)^\gamma Q(y)^{1-\gamma} \right).$$  \hspace{1cm} (2)

The Rényi divergence has two relevant special cases: $\lim_{\gamma \to 1}(1/\gamma)D_\gamma^R(P||Q) = KL(P||Q)$, and $\lim_{\gamma \to 0}(1/\gamma)D_\gamma^R(P||Q) = KL(Q||P)$ (which can be seen via L'Hôpital's rule).

**Definition 2 (Weighted Rényi Divergence Aggregation).** The weighted Rényi divergence aggregation is a weighted divergence aggregation given by (1), where each divergence $D(P_i, Q) = \gamma_i^{-1}D_\gamma^R[P_i||Q]$.

Note that each component $i$ in (1) can have a Rényi divergence with an individualized parameter $\gamma_i$. Sometimes we will assume that all divergences are the same, and refer to a single $\gamma = \gamma_i \forall i$ used by all the components.

3.2. Properties of Weighted Rényi Divergence Aggregation

**Proposition 1.** Weighted Rényi divergence aggregation satisfies the implicit equation for $P(y)$ of

$$P(y) = \frac{1}{Z} \sum_i w_i \gamma_i^{-1} P_i(y)^{\gamma_i} P(y)^{-\gamma_i} \left( \sum_{y'} P_i(y')^{\gamma_i} P(y')^{1-\gamma_i} \right)^{-1}$$  \hspace{1cm} (3)

where $w_i$ are given weights, and $Z = Z(\{\gamma_i\}) = \sum_i w_i \gamma_i^{-1}$ is a normalisation constant, and $\{\gamma_i\}$ is the set of Rényi divergence parameters.

**Proof.** Outline: Use $D(P_i, Q) = \gamma_i^{-1}D_\gamma^R[P_i||Q]$ from (2) in Equation (1), and build the Lagrangian incorporating the constraint $\sum_y Q(y) = 1$ with Lagrange multiplier $Z$. Use calculus of variations w.r.t. $Q(y)$ to get $K$ equations

$$\sum_i w_i \gamma_i^{-1} P_i(y)^{\gamma_i} P(y)^{-\gamma_i} \left( \sum_{y'} P_i(y')^{\gamma_i} P(y')^{1-\gamma_i} \right) - Z = 0$$  \hspace{1cm} (4)

for the optimum values of $P(y)$. Multiply each equation with $P(y)$ and find $Z = \sum_j w_j \gamma_j^{-1}$ by summing over all equations. Rearrange to obtain the result. \hfill $\Box$

**Proposition 2.** Weighted Rényi divergence aggregation interpolates between linear opinion pooling ($\gamma \to 1$) and log opinion pooling ($\gamma \to 0$).

**Proof.** Outline: Set $\gamma_i = 1$ in (3) to obtain a standard linear opinion pool. For log opinion pool, set $\gamma_i = \gamma$, and take $\gamma \to 0$. Note (3) can be written $Z = \sum_i w_i \gamma_i^{-1} \frac{\partial}{\partial \gamma_i} D_\gamma^R[P_i||Q]$. Using L'Hôpital's rule on each element in the sum and switching the order of differentiation $\left( \partial/\partial \gamma_i \right) \left( \partial/\partial Q \right) = \left( \partial/\partial Q \right) \left( \partial/\partial \gamma_i \right)$ gives the result. \hfill $\Box$

In Section 4 we show that Rényi divergence aggregation provides the maximum entropy distribution for combining together agent distributions where the belief of each agent is subject to a particular form of bias. Two consequences that are worth alerting the reader to ahead of that analysis are:

\dagger\dagger The divergence for $\gamma = 1$ can be obtained by analytic continuation and is equivalent to $KL(P||Q)$.
1. If all agents form beliefs on data drawn from the same (unbiased) distribution then the maximum entropy distribution is of the form of a log opinion pool.

2. If all agents form beliefs on unrelated data then the maximum entropy distribution is of the form of a linear opinion pool.

4. Maximum Entropy Arguments

Consider the problem of choosing an aggregator distribution $P(.)$ given a number of individual distributions $P_i(.)$. These individual distributions are assumed to be learnt from data by a number of individual agents. We will assume the individual agents did not (necessarily) have access to data drawn from $P(.)$, but instead the data seen by the individual agents was biased, and instead sampled from distribution $Q_i(.)$. In aggregating the agent beliefs, we neither know the target distribution $P(.)$, nor any of the individual bias distributions $Q_i(.)$. However we do have validation estimates of the test log likelihood performance of the individual distributions in their own domains, and assume this test log likelihood is accurate in that domain, to all intents and purposes. As far as the individual agents are concerned they trained and evaluated their methods on their individual data, unconcerned that their domains were biased with respect to the domain we care about. We can think of this scenario as convergent dataset shift (Storkey, 2009), where there is a shift from the individual training to a common test scenario.

The individual agent data is biased, not unrelated, and so we make the assumption that the individual distributions $Q_i(.)$ are related to $P(.)$ in some way. Typically the support of $Q_i$ will not exceed the support of $P$ (the data is biased, not contaminated). Specifically we make the assumption that $Q_i(.)$ is close to $P(.)$ in the sense that $KL(Q_i||P)$ is subject to some bound. Choosing this form of $KL$ divergence ensures the support conditions are met.

Given this scenario, a reasonable ambition is to find maximum entropy distributions $Q_i$ that capture the performance of the individual distributions $P_i$, while at the same time being related via an unknown distribution $P$. We write the test performance as constraints:

$$\sum_y Q_i(y) \log P_i(y) = a_i, \quad (5)$$

where we have used our assumption that the validation estimates of test log-likelihood were accurate estimates of the generalisation error.

The nearness constraints\(^2\) for $Q_i$ are written as

$$KL(Q_i(.)||P(.)) \leq A_i, \quad (6)$$

$$\Rightarrow \sum_y Q_i(y) \log \frac{Q_i(y)}{P(y)} \leq A_i \text{ for some } P(.). \quad (7)$$

Given these constraints, the maximum entropy (minimum negative entropy) Lagrangian optimisation can be written as $\arg \min_{\{Q_i\}, P} L(\{Q_i\}, P)$, where

$$L(\{Q_i\}, P) = \sum_i \sum_y Q_i(y) \log Q_i(y) + \sum_i \rho_i \left( \sum_y Q_i(y) \frac{Q_i(y)}{P(y)} - A_i + s_i \right)$$

$$- \sum_i \lambda_i \left( \sum_y Q_i(y) \log P_i(y) - a_i \right) + \sum_i b_i (1 - \sum_y Q_i(y)) + c (1 - \sum P(y)) \quad (8)$$

where $s_i$ are slack variables $s_i \geq 0$, and $\rho_i, \lambda_i, b_i$ and $c$ are Lagrange multipliers. This minimisation chooses maximum entropy $Q_i$, while ensuring there is a distribution $P(.)$ for which the nearness constraints are met. The final two terms of (8) are normalisation constraints for $Q_i$ and $P$.

Taking derivatives with respect to $Q_i(y)$ and setting to zero gives

$$Q_i(y) = \frac{1}{Z_i} P(y)^{\frac{\rho_i}{\lambda_i}} P_i(y)^{\frac{\gamma_i}{\lambda_i}} \quad (9)$$

where $Z_i$ is a normalisation constant.

Given these $Q_i$, we can find an optimal $P$. Taking derivatives with respect to $P(y)$ and setting to zero gives

$$P(y) = \sum_i \frac{\rho_i}{\sum \rho_i} Q_i(y)$$

$$= \sum_i w_i \frac{1}{Z_i} (P_i(y)^{\lambda_i})^{\gamma_i} P(y)^{1-\gamma_i} \quad (10)$$

where $w_i = \rho_i / \sum \rho_i$, and $\gamma_i = 1/(1 + \rho_i)$, and $Z_i = \sum (P_i(y)^{\lambda_i})^{\gamma_i} P(y)^{1-\gamma_i}$. Comparing this with (3) we see that this form of maximum entropy distribution is equivalent to the Renyi divergence aggregator of annealed forms of $P_i$. The maximum entropy parameters of the aggregator could be obtained by solving for the constraints or estimated using test data from $P(y)$. Empirically we find that, if all the $P_i$ are trained on the same data, or on data subject to sample-selection bias (rather than say an annealed form of the required distribution), then $\nu_i \approx 1$.

\(^2\)We could work with a nearness penalty of the same form rather than a nearness constraint. The resulting maximum entropy solution would be of the same form.
Note that the parameter $\rho_i$ controls the level of penalty there is for a mismatch between the biased distributions $Q_i$ and the distribution $P$. If all the $\rho_i$ are zero for all $i$ then this penalty is removed and the $Q_i$ can bear little resemblance to the $P$ and hence to one another. In this setting (10) becomes a standard mixture and the aggregator is a linear opinion pool. If however $\rho_i$ tends to a large value for all $i$, then the distributions $Q_i$ are required to be much more similar. In this setting (10) becomes like a log opinion pool.

5. Implementation

Rényi-mixtures can be implemented with direct optimization, stochastic gradient methods, or using a variational optimization for the sum of weighted divergences, which is described here. The weighted Rényi Divergence objective given by Definition 2) can be lower bounded using

$$\sum_i w_i D(P_i, Q) \geq \sum_i \frac{w_i \gamma_i}{\gamma_i - 1} \sum_{y=1}^K Q_i(y) \left( \log[P_i(y)^{\gamma_i} Q(y)^{1-\gamma_i}] - \log Q_i(y) \right)$$

(11)

where we have introduced variational distributions $Q_i$, and used Jensen’s inequality. Note equality is obtained in (11) for $Q_i(y) \propto P_i(y)^{\gamma_i} Q(y)^{1-\gamma_i}$. Optimizing for $Q$ gives $P(y) = Q_{\text{opt}}(y) = \sum_i w_i^* Q_i(y)$ with $w_i^* = w_i \gamma_i^{-1} / \sum_i w_i \gamma_i^{-1}$.

This leads to an iterative variational algorithm that is guaranteed to optimize (11): iteratively set $Q_i(y) \propto P_i(y)^{\gamma_i} Q(y)^{1-\gamma_i}$, and then set $Q(y) \propto \sum_i w_i^* Q_i(y)$. The optimization of the parameters $w_i^*$ also naturally fits within this framework. $Q(y)$ is a simple mixture of $Q_i(y)$. Hence given $Q_i(y)$, the optimal $w_i^*$ are given by the optimal mixture model parameters. These can be determined using a standard inner Expectation Maximization loop. In practice, we get faster convergence if we use a single loop. First set $Q_i(y) \propto P_i(y)^{\gamma_i} Q(y)^{1-\gamma_i}$. Second compute $q_{in} = w_i^* Q_i(y_{in}) / \sum_i w_i^* Q_i(y_{in})$. Third set $w_i^* = \sum_n q_{in} / \sum_{in} q_{in}$. Finally set $Q(y) \propto \sum_i w_i^* \gamma_i Q_i(y)$. This is repeated until convergence. All constants of proportionality are given by normalisation constraints. Note that where computing the optimal $Q$ may be computationally prohibitive, this process also gives rise to an approximate divergence minimization approach, where $Q_i$ is constrained to a tractable family while the optimizations for $Q_i$ are performed.

6. Demonstrations

To show the practical validity of the maximum entropy arguments we demonstrate the result on a simple aggregation test bed. We need to be able to test the variation of the aggregator performance as the bias of the agent datasets is gradually changed. This requires that the data does not dramatically change across tests of different biases. We tested this process using a number of bias generation procedures, all with the same implication in terms of results.

The details of the data generation and testing is given in Algorithm 1. We used $N_A = 10$, $K = 64$, $N_{V_{\text{a}}} = 100$, $P^*$ was a discretized $U([0, 1])$ to generate the artificial data that gave the results displayed here. Equivalent results were found for all (non-trivial) parameter choices we tried, as well as using completely different data generation procedures generating biased agent data. Full code for all the tests will be made available at point of publication.

Algorithm 1 Generate test data for agents with different biases, and test aggregation methods.

1. Select a target discrete distribution $P^*(\cdot)$ over $K$ values. Choose $N_A$, the number of agents.
2. Sample IID a large number $N_{V_{\text{a}}}$ of values from the target distribution to get a validation set $D_{V_{\text{a}}}$
3. Sample IID a large number $N$ of values \{\{y_n; n = 1, 2, 3, \ldots, N\} from the target distribution to get the base set $D$ from which agent data is generated.
4. Sample bias probabilities $f_i(y)$ for each agent to be used as a rejection sampler.
5. for annealing parameter $\beta = 0$ TO 4 do
6. for each agent $i$ do
7. Anneal $f_i$ to get $f_i^*(y) = f_i(y)^{\beta} / \max_y f_i(y)^{\beta}$.
8. For each data point $y_i$, reject it with probability $(1 - f_i^*(y_i))$.
9. Collect the first 10000 unrejected points, and set $P_i$ to be the resulting empirical distribution.
10. This defines the distribution $P_i$ for agent $i$ given the value of $\beta$.
11. end for
12. Find aggregate $P(\cdot)$ for different aggregators given agent distributions $P_i$ and an additional $P_0$ corresponding to just the uniform distribution, using the validation dataset $D_{V_{\text{a}}}$ for any parameter estimation.
13. Evaluate the performance of each aggregator using the KL Divergence between the target distribution $P^*(\cdot)$ and the aggregate distribution $P(\cdot)$: $KL(P^* || P)$.
14. end for

A second test was then made on real data, consisting of accurately predicting distributions of chords from Bach chorales (Bache & Lichman, 2013). The Bach chorales data was split equally and randomly into training and test distributions. Then training data from half of the chorales was chosen to be shared across all the agents. After that each agent received additional training data from a random
half of the remaining chorales. Each agent was trained using a mixture of Bernoulli’s with a randomized number of mixture components between 5 and 100, and a random regularisation parameter between 0 and 1. 10 agents were used and after all 10 agents were fully trained, the Rényi mixture weights were optimized using the whole training dataset. Performance results were computed on the held out test data.

\begin{figure}
\centering
\includegraphics[width=0.5\textwidth]{figure1.png}
\caption{Plot of the KL divergence against $\log \gamma$ for one dataset with $\beta = 0$ (lower lines, blue) through to $\beta = 4$ (upper lines, red) in steps of 0.5. Note that, unsurprisingly, more bias reduces performance. However the optimal value of $\gamma$ (lowest KL), changes as $\beta$ changes. For low values of $\beta$ the performance of $\gamma = 0$ (log opinion pools) is barely distinguishable from other low $\gamma$ values. Note that using a log opinion pool (low $\gamma$) when there is bias produces a significant hit on performance.}
\end{figure}

Figure 1 is a graph of the test performance on different biases for different values of $\log(\gamma_i)$ in (10), where all $\gamma_i$ are taken to be identical and equal to $\gamma$. Figure 2 shows how the optimal value of $\gamma$ changes, as the bias parameter $\beta$ changes. Parameter optimization was done using a conjugate gradient method. The cost of optimization for Rényi mixtures is comparable to that of log opinion pools. Figure 3 shows the performance on the Bach chorales with 10 agents, with the implementation described in Section 5. Again in this real data setting, the Rényi mixtures show improved performance.

These demonstrations show that when agents received a biased subsample of the overall data then Rényi-mixtures perform best as an aggregation method, in that they give the lowest KL divergence. As the bias increases, so the optimal value of $\gamma$ increases. In the limit that the agents see almost the same data from the target distribution, Rényi-mixtures with small $\gamma$ perform the best, and are indistinguishable from the $\gamma = 0$ limit. Rényi mixtures are equivalent to log opinion pools for $\gamma \to 0$.

In (Storkey et al., 2012) the authors show, on a basket of UCI datasets, that market aggregation with agents having isoelastic utilities performs better than simple linear opinion pools (markets with log utilities) and products (markets with exponential utilities) when the data agents see is biased. We shall also see in Section 8 that such markets implement Rényi mixtures, and so this provides additional evidence that Rényi mixtures are appropriate when combining biased predictors.

\section{7. Aggregation Test on Kaggle Competition}

When all agents see the same data, the maximum-entropy aggregate is the log opinion pool. One setting of particular interest is in machine learning competitions, where the same training data is made publicly available to everyone. In this section we compare a number of aggregation methods on a real competition, and confirm that log opinion pools are the aggregation method of choice.

To analyze the use of combination methods in a realistic competition setting, we need data from an appropriate competitive setup. For this purpose we designed and ran the Kaggle-in-Class competition described in this section. The competition consisted of a critical problem in low-level image analysis: the image coding problem, which is fundamental in image compression, inlling, super-resolution and denoising. We used data consisting of images from van Hateren’s Natural Image Dataset\(^\text{\footnote{http://bethgelab.org/datasets/vanhateren/}}\) (Hateren & Schaaf, 1998). The data was preprocessed using Algorithm 2 to put it in a form suitable for a Kaggle competition, and ensure the data sizes were sufficient for use on student machines.
and that submission files were suitable for uploading (this is the reason for the 6 bit grayscale representation).

The competition problem was to infer $P(y|x,i)$, the predictive distribution for the grayscale value of the pixel at a given location, where $y$ takes one of 64 possible values. The information given was the image number $i$ and a raster scan $x$ of an image patch above and up to a given pixel location (see Figure 4a, which clarifies the form of the data). Patches were taken randomly from a large image corpus. Competitors were provided with three files specified in Algorithm 2. The competition submissions were unnormalized log probabilities at the test set points: log $P(y = k|x,i)$, with one column for each $k$. The normalization was computed by the competition evaluation mechanism to prevent any room for cheating by false normalization. The test cases were split into a public set and

A private set. The competitor was given the perplexity on the public set at submission time, but the final ranked ordering was on the private set. The perplexity is given by perplexity $= \exp \left( -\frac{1}{N_t} \sum_{j=1}^{N_t} \log(P(y_j = c_j|x_j,i_j)) \right)$, where $N_t$ is the number of test pixels and $c_j$ is the true class the $j$th pixel belongs to, and $x_j$ and $i_j$ are the provided covariates. Note that perplexity is equivalent to test probability up to a monotonic transformation.

There were 46 competitors, with a total of 440 submissions. Some submissions were highly erroneous (submitting probabilities instead of log probabilities etc.), but competitors quickly fixed these issues for future submissions. A uniform prediction was used as a dummy baseline which has perplexity 64. We chose as agent distributions the 269 submissions that had perplexity greater than 64.

7.1. Analysis of the Competition

The following aggregation methods were tested: weighted Rényi divergence aggregators, including linear opinion pools and log opinion pools, simple averaging of the top submissions (with an optimized choice of number), and a form of heuristic Bayesian model averaging, via an annealed likelihood: $P(y|\cdot) \propto \sum_j P_j(y|\cdot) (P_j|\mathcal{D}_w)\alpha$, where $\alpha$ is an aggregation parameter choice. The weighted Rényi divergence aggregators were optimized using stochastic gradient methods, until the change between epochs became negligible. The validation set (20,000 pixels) is used for learning the aggregation parameters. The test set (also 20,000 pixels) is only used for the test results.

In order to show the generalization performance of all the aggregation methods, we split the private set into 10 subsets and apply each method to each subset to obtain the mean perplexity and the statistics for the difference in perplexity for all methods. The mean perplexity values and standard deviation for all the methods tested can be seen in Figure 4b and Table 1. Table 1 also shows the difference in log perplexity between each approach and shows the estimated standard deviation of those differences (log perplexity values appear approximately Gaussian), and corresponding single-tailed $t$-test sample probability (p-value) under the null assumption that the method is equivalent to the linear opinion pool. There is a statistically significant performance benefit of using logarithmic opinion pooling over linear pooling. The parameter-based pooling methods perform better than simple averages and all forms of heuristic model averaging as these are inflexible methods.

---

4The heuristic model averaging, includes Bayesian Model Averaging as a special case. However we emphasize that Bayesian Model Averaging, though discussed in the context of aggregation (Dietterich, 2000; Domingos, 1997), is not formulated as an aggregation method: it assumes only one of the submissions is actually correct (Minka, 2002).
A Continuum from Mixtures to Products: Aggregation under Bias

Figure 4. (a) Competition form: competitors had to infer the probabilities for a pixel taking each of 64 values, given values for pixels above and to the left of that pixel. Note that i labels the original image source: iml00004.imk. The image patch is 35 pixels (horiz) × 30 pixels (vert). (b) Perplexity on the test set of all the compared aggregation methods against η = 1/γ. For each method, the best performance is plotted: simple averaging on top 10 submissions, Bayesian averaging (and with power heuristics) with α = 0.0049, and Rényi mixture with η = 1/30 = 30 confirming that Rényi mixtures with small γ are equivalent to log opinion pools. Log opinion pools perform best as suggested by the maximum entropy arguments.

All results have been tested for reproducibility using multiple initializations.

In this context, the ‘value’ of a submission is not the same as its performance: rather the importance of the contribution to the overall aggregate probability depends on how it is combined. A single good contribution, that is different from the others, is usually more valuable than a slightly higher scoring contribution that is very similar to all the others. Figure 5 shows the weights of the contributions for the log opinion pools and the isoeleastic markets with η = 30 (for η ≥ 30 the results are sufficiently similar to the log opinion pool). Further analysis (omitted here for space reasons) shows that the obvious spikes in weight are due to particular contributions that are noticeably different for the bulk of the high scoring contributions.

Figure 5. Weight (wealth) distribution over 269 ranked submissions (agents) for the log opinion pool.

8. Machine Learning Markets and Rényi Divergence Aggregation

Machine learning markets with isoeleastic utilities (Storkey et al., 2012) are an information market based aggregation method. Independent agents with different beliefs trade in a securities market. The equilibrium prices of the goods in that securities market can then be taken as an aggregate probability distribution, aggregating the individual agent beliefs. From (Storkey et al., 2012), agents indexed by i with belief $P_i(y)$, wealth $W_i$ and utility function $U_i(.)$ trade in Arrow-Debreu securities derived from each possible outcome of an event. Given the agents maximize expected utility, the market equilibrium price of the securities $c(y)$ is used as an aggregate model $P(y) = c(y)$ of the agent beliefs. When each agent’s utility is an isoeleastic utility of the form $U_i(W) = W^{1-\eta_i}/(1-\eta_i)$ with a risk-averseness parameter $\eta_i$, the market equilibrium $P(y)$ is implicitly given by

$$P(y) = \sum_i W_i \frac{P_i(y)\gamma_i}{\sum_j W_j \sum_{y'} P_j(y')\gamma_j P(y')^{1-\gamma_j}}$$

with $\gamma_i = \eta_i^{-1}$ (generalising (10) in (Storkey et al., 2012)). This shows the isoeleastic market aggregator linearly mixes together components that are implicitly a weighted product of the agent belief and the final solution. Simple compar-

---

Table 1. Top: The perplexity on the test set for all the aggregation methods. The log opinion pool and the Rényi mixture for $\eta = (1/30)$ are fairly equivalent. Bottom: Relative log perplexity using linear pool as the baseline, and corresponding p-value.

<table>
<thead>
<tr>
<th>Methods</th>
<th>SimpleAvgBest</th>
<th>Heuristic</th>
<th>LogOP</th>
<th>Rényi Mixture</th>
</tr>
</thead>
<tbody>
<tr>
<td>Perplexity: (LinearOP: 2.894 ± 0.060) mean±std</td>
<td>2.931 ± 0.065</td>
<td>2.929 ± 0.067</td>
<td>2.837 ± 0.067</td>
<td>2.836 ± 0.061</td>
</tr>
<tr>
<td>Log Perplexity Difference from LinearOP: mean±std</td>
<td>0.013 ± 0.021</td>
<td>0.012 ± 0.021</td>
<td>-0.020 ± 0.024</td>
<td>-0.020 ± 0.021</td>
</tr>
<tr>
<td>p value</td>
<td>1</td>
<td>1</td>
<td>8.0 × 10^{-7}</td>
<td>5.3 × 10^{-7}</td>
</tr>
</tbody>
</table>
ison of this market equilibrium with the Rényi Divergence aggregator (3) shows that the market solution and the Rényi Divergence aggregator have the same form.

This implies that the process of obtaining the market equilibrium is a process for building the Rényi Divergence aggregator, and hence machine learning markets provide a method of implementation of weighted Rényi divergence aggregators. The benefit of market mechanisms for machine learning is that they are incentivized. There is no assumption that the individual agents behave cooperatively, or that there is an overall controller who determines agents’ actions. Simply, if agents choose to maximize their utility then the result is weighted Rényi Divergence aggregation.

9. Discussion

We demonstrate empirically that log opinion pooling provides better performance in a real competition setting than other pooling methods, including linear pooling. This matches the fact that log opinion pooling can be seen as a maximum entropy aggregator in the circumstance that all entries are derived from the same dataset.

However when agents are training and optimising on different datasets than one another, log opinion pooling is no longer a maximum entropy aggregator. Instead, under certain assumptions, the weighted Rényi divergence aggregator is the maximum entropy solution, and tests confirm this practically. The weighted Rényi divergence aggregator can be implemented using isoelastic machine learning markets.

Though there is some power in providing aggregated prediction mechanisms as part of competition environments, there is the additional question of the competition mechanism itself. With the possibility of using the market-based aggregation mechanisms, it would be possible to run competitions as prediction market or collaborative scenarios (Abernethy & Frongillo, 2011), instead of as winner takes all competitions. This alternative changes the social dynamics of the system and the player incentives, and so it is an open problem as to the benefits of this. We recognize the importance of such an analysis as an interesting direction for future work.

References


Goldbloom, A. Data prediction competitions – far more than just a bit of fun. In IEEE International Conference on Data Mining Workshops, 2010.


Robert M. Bell, Yehuda Koren and Volinsky, Chris. All together now: A perspective on the NETFLIX PRIZE. Chance, 24, 2010.

Storkey, A.J. When training and test sets are different: Characterising learning transfer. In Lawrence, Candela


