Learning Monotonic Linear Functions

Adam Kalai

TTI-Chicago
kalai@tti-c.org

Abstract. Learning *probabilities* (p-concepts [13]) and other real-valued concepts (regression) is an important role of machine learning. For example, a doctor may need to predict the probability of getting a disease $P[y|x]$, which depends on a number of risk factors.
Generalized additive models [9] are a well-studied nonparametric model in the statistics literature, usually with monotonic link functions. However, no known efficient algorithms exist for learning such a general class. We show that regression graphs *efficiently* learn such real-valued concepts, while regression trees *inefficiently* learn them. One corollary is that any function $E[y|x] = u(w \cdot x)$ for u *monotonic* can be learned to arbitrarily small squared error ϵ in time polynomial in $1/\epsilon$, $|w|_1$, and the Lipschitz constant of u (analogous to a margin). The model includes, as special cases, linear and logistic regression, as well as learning a noisy half-space with a margin [5, 4].
Kearns, Mansour, and McAllester [12, 15], analyzed decision trees and decision graphs as boosting algorithms for classification accuracy. We extend their analysis and the boosting analogy to the case of real-valued predictors, where a small positive *correlation coefficient* can be boosted to arbitrary accuracy. Viewed as a noisy boosting algorithm [3, 10], the algorithm learns both the target function and the asymmetric noise.

1 Introduction

One aim of machine learning is predicting probabilities (such as p-concepts [13]) or general real values (regression). For example, Figure 1 illustrates the standard prediction of relapse probability for non-Hodgkin's lymphoma, given a vector of patient features. In this application and many others, probabilities and real-valued estimates are more useful than simple classification.

A powerful statistical model for regression is that of generalized *linear* models [16], where the expected value of the dependent variable y can be written as $E[y|x] = u(w \cdot x)$, an arbitrary *link function* $u : \mathbb{R} \to \mathbb{R}$ of a linear function of the feature vector $x \in \mathbb{R}^n$. Our results apply to *mono-linear functions*, where u is monotonic and Lipschitz continuous.[1]

Linear and logistic regression both learn mono-linear functions. The model also captures (noisy) linear threshold functions with a margin [5, 4].[2]

[1] A function u is Lipschitz continuous with constant L if $|u(a) - u(b)| \leq L|a - b|$ for all $a, b \in \mathbb{R}$. (For differentiable u, $|u'(a)| \leq L$.)

[2] For a linear threshold function, $L = 1/\text{margin}$.

# Risk Factors	complete response rate	relapse-free 2-year survival	relapse-free 5-year survival	2-year survival	5-year survival
0,1	87%	79%	70%	84%	73%
2	67%	66%	50%	66%	51%
3	55%	59%	49%	54%	43%
4,5	44%	58%	40%	34%	26%

Risk Factors: $x_1 \geq 60, x_2 \geq 2, x_3 \geq 2, x_4 \geq$ normal, and $x_5 \geq 3$.
($x_1 =$ age, $x_2 =$ # extranodal sites, $x_3 =$ performance status, $x_4 =$ LDH, $x_5 =$ stage.)

Fig. 1. Non-Hodgkin's lymphoma International Prognostic Index probabilities [21]. Each probability (column) can be written in the form $u(I(x_1 \geq 60) + \ldots + I(x_5 \geq 3))$ for monotonic u, but does not fit a linear or logistic (or threshold) model.

In fact, our results apply to the more general *generalized additive models.* Random examples are seen from a distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, where $\mathcal{X} = \mathbb{R}^n$ and $\mathcal{Y} \subset \mathbb{R}$. ($Y = \{0, 1\}$ corresponds to probability learning [13].) The assumption is that $f(x) = E[y|x] = u(\sum_i v_i(x_i))$, where u is a continuous monotonic *link function* and each $v_i : \mathbb{R} \to \mathbb{R}$ is an arbitrary function of bounded total variation[3].

A *regression tree* is simply a decision tree with real (rather than binary) predictions in the leaves. A *decision graph* (also called branching program, DAG, or binary decision diagram) is a decision tree where internal nodes may be merged. We suggest the natural *regression graph*, which is a decision graph with real-valued predictions in the leaves (eq. a regression graph with merging). We give an algorithm for learning these functions that is derivative of Mansour and McAllester [15]. We show that, for error of h defined as $\epsilon(h) = E_{\mathcal{D}}[\big(h(x) - f(x)\big)^2]$, the error of regression graphs decreases quickly, while regression trees suffer from the "curse of dimensionality."

Theorem 1. *Let $\mathcal{D}$ be a distribution on $\mathcal{X} \times \mathcal{Y}$, where $\mathcal{X} \subseteq \mathbb{R}^n$ and $\mathcal{Y} \subseteq [0, 1]$. Suppose $f(x) = E[y|x] = u(\sum v_i(x_i))$, where u is monotonic (nondecreasing or nonincreasing). Let L be the Lipschitz constant of u and $V = \sum V_{v_i}$ is the sum of the total variations of v_i.*
1. Natural top-down regression graph learning, with exact values of leaf weights and leaf means, achieves $\epsilon(R) \leq \epsilon$ with size$(R) \leq L^3V^3/(10\epsilon^4)$.
2. For regression trees *with exact values, $\epsilon(R) \leq \epsilon$ with size$(R) \leq 2(1.04)^{L^2V^2/\epsilon^3}$.*

While the above assumes knowing the exact values of parameters, standard tools extend the analysis to the case of estimation, as described in Section 5.3. Also, notice the Winnow-like dependence on V. In the case where each $v_i(x_i) = w_i x_i$ and $\mathcal{X} = [0, 1]^n$, $V = W = \sum |w_i|$. If $f(x)$ is a linear threshold function of boolean $\mathcal{X} = \{0, 1\}$, and $w_i \in \mathbb{Z}$, then $V = W$ and u can be chosen with $L = 1$, since the increase from $u(z) = 0$ to $u(z) = 1$ happens between integer z's. Since the sample complexity depends only logarithmically on the n, if there are only a

[3] The total variation of v is how much "up and down" it goes. For differentiable functions, it's $\int_{-\infty}^{\infty} |v'(a)| da$. For monotonic functions it's $\sup_a v(a) - \inf_a v(a)$.

few relevant dimensions (with small W) then the algorithm will be very attribute efficient.

1.1 Real-valued boosting

In learning a regression graph or tree, one naturally searches for binary splits of the form $x_i \geq \theta$. We first show that there always exists such a split with positive *correlation coefficient*. We then show that a positive correlation leads to a reduction in error.

This is clearly similar to boosting, and we extend the analyses of Kearns, Mansour, and McAllester, who showed that decision trees and more efficiently decision graphs can perform a type of boosting [20]. Rather than a weakly accurate hypothesis (one with accuracy $P[h(x) = f(x)] \geq 1/2$), we use weakly correlated hypotheses that have correlation bounded from 0. This is similar to the "okay" learners [10] designed for noisy classification.[4]

2 Related work

While generalized additive models have been studied extensively in statistics [9], often with monotonic link functions, to the best of our knowledge no existing algorithm can efficiently guarantee $\epsilon(h) < \epsilon$ for arbitrarily small ϵ, even though such guarantees exist for much simpler single-variable problems.

For example, an algorithm for efficiently learning a monotonic function of a *single variable* $x \in \mathbb{R}$, $f(x) = E[y|x]$ was given by Kearns and Schapire [13]. Statisticians also have efficient learning algorithms for this *scatterplot smoothing* problem.

For the important special case of learning a linear threshold function with classification noise, Bylander showed that Perceptron-like algorithms are efficient in terms of a margin [5]. This would correspond to $u = \eta$ for negative examples, $u = 1 - \eta$ for positive examples, and linearly increasing at a slope of $(1 - 2\eta)/\text{margin}$ in between, where η is the noise rate. Blum et. al. removed the dependence on the margin [4]. Bylander also proved efficient *classification* in the case with a margin and random noise that monotonically and *symmetrically* decreased in the margin. It would be very interesting if one could extend these techniques to a non-symmetric noise rate, as symmetric techniques for other problems, such as learning the intersection of half-spaces with a symmetric density [1], have not been extended.

3 Definitions

We use the Kearns and Schapire's definition of efficient learnability in a real-valued setting [13]. There is a distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$. Kearns and Schapire

[4] As observed in [10], correlation is arguably a more popular and natural measure of weak association between two random variables than accuracy, e.g. the boolean indicators $f(x) =$"person x lives in Chicago" and $h(x) =$"person x lives in Texas" are negatively correlated, but have high accuracy $P[h(x) = f(x)]$.

take binary labels $\mathcal{Y} = \{0, 1\}$ in the spirit of learning probabilities and PAC learning [22]. In the spirit of regression, we include real labels $\mathcal{Y} \subseteq \mathbb{R}$, though the theory is unchanged. The target function is $f(x) = E[y|x]$.

An algorithm A *learns* concept class $\mathcal{C}$ of real-valued functions from $\mathcal{X}$, if, for every $\epsilon, \delta > 0$ and every distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$ such that $E_{\mathcal{D}}[y|x] = f(x) \in \mathcal{C}$, given access to random labelled examples from $\mathcal{D}$, with probability $1 - \delta$, A outputs hypothesis h with error,

$$\epsilon(h) = E_{\mathcal{D}}[\big(h(x) - f(x)\big)^2] \leq \epsilon.$$

It *efficiently learns* if it runs in time polynomial in $1/\epsilon, 1/\delta$, and size(f).[5]

While $\epsilon(h)$ cannot directly be estimated, $E[(h(x)-y)^2]$ can be and is related:

$$E_{\mathcal{D}}[(h(x) - y)^2] = E_{\mathcal{D}}[(f(x) - y)^2] + E_{\mathcal{D}}[\big(h(x) - f(x)\big)^2].$$

Let the indicator function $I(P) = 1$ if predicate P holds and 0 otherwise. Recall various statistical definitions for random variables $u, v \in \mathbb{R}$.

$$\begin{aligned}
\mu_u &= E[u] \\
\mathrm{cov}(u, v) &= E[(u - \mu_u)(v - \mu_v)] = E[uv] - \mu_u \mu_v \\
\mathrm{var}(u) &= \sigma_u^2 = \mathrm{cov}(u, u) = E[(u - \mu_u)^2] = E[u^2] - \mu_u^2 \\
\sigma_u &= \sqrt{\mathrm{var}(u)} \\
\mathrm{cor}(u, v) &= \rho_{uv} = \frac{\mathrm{cov}(u, v)}{\sigma_u \sigma_v}
\end{aligned}$$

In most of the analysis, the random variables $f, h : \mathcal{X} \to \mathbb{R}$ can either be thought of as functions or the induced random variables for x from $\mathcal{D}$. We use ρ_{fh} or $\rho_{f(x)h(x)}$, as is convenient. We will use a few properties of covariance. It is shift invariant, i.e. $\mathrm{cov}(u + c, v) = \mathrm{cov}(u, v)$ for a constant c. It is symmetric and *bilinear*, i.e.

$$\mathrm{cov}(c_1 u_1 + c_2 u_2, v) = c_1 \mathrm{cov}(u_1, v) + c_2 \mathrm{cov}(u_2, v),$$

for constants c_1, c_2.

The (possibly infinite) *Lipschitz constant* of a function $u : \mathbb{R} \to \mathbb{R}$ is,

$$L = \sup_{a \neq b} \frac{|u(a) - u(b)|}{|a - b|}.$$

Let V_g be the *total variation* of a function $g : \mathbb{R} \longrightarrow \mathbb{R}$, which can be defined as the following maximum over all increasing sequences of $a_i \in \mathbb{R}$.

$$V_g = \sup_{k \in \mathbb{Z}} \sup_{a_1 < a_2 < \ldots < a_k} \sum_{i=1}^{k-1} |g(a_{i+1}) - g(a_i)|.$$

[5] In our example size$(f) = LV$, where L is a Lipschitz constant and V is total variation.

4 Top-down regression graph learning

For our purposes, a *regression tree* R is a binary tree with boolean split predicates, functions from $\mathcal{X}$ to $\{0,1\}$, at each internal node. The leaves are annotated with real numbers. A *regression graph* R is just a regression tree with merges. More specifically, it's a directed acyclic graph where each internal node again has a boolean split predicate and two labelled outgoing edges, but children may be shared across many parents. The internal nodes determine a partition of $\mathcal{X}$ into the leaves. The weight of a leaf is $w_\ell = P[x \in \ell]$. The value of a leaf ℓ is $q_\ell = E[y|x \in \ell]$. We define the prediction $R(x)$ to be the value of the leaf that x falls into. (These quantities are exact; estimation is discussed in the next section.) This enables us to discuss the correlation coefficient and other quantities relating to R. We also define the distribution $\mathcal{D}_\ell$, which is the distribution $\mathcal{D}$ restricted to the leaf ℓ.

It is straightforward to verify that $\mu_y = \mu_f = \mu_R = \sum_\ell w_\ell q_\ell$. Most decision tree algorithms work with a potential function, such as $\epsilon(R) = E[\big(R(x) - f(x)\big)^2]$, and make each local choice based on which one decreases the potential most. In Appendix C, we show that all of the following potential functions yield the same ordering on graphs:

$$\epsilon(R),\ -\sum w_\ell q_\ell^2,\ -\rho_{Rf},\ -\sigma_R^2,\ -\sum_\ell w_\ell\big(a(q_\ell - b)\big)^2,\ \sum_\ell w_\ell 4 q_\ell(1-q_\ell)$$

We use the second one, $G(R) \overset{\text{def}}{=} -\sum_\ell w_\ell q_\ell^2$, because it is succ. in terms of w_ℓ, q_ℓ. However, the (a,b) formulation (for $a \neq 0, b \in \mathbb{R}$) illustrates that minimizing $G(R)$ is scale-invariant (and shift-invariant), which mean that the algorithm can be run as-is even if Y is larger than $[0,1]$ (and the guarantees scale accordingly). Also, the last quantity shows that it is equivalent to the Gini splitting criterion used by CART [6].

A natural top-down regression graph learning algorithm with stopping parameter $\Delta_{\min}$ is as follows. We start with a single leaf ℓ_1 and repeat:

1. Sort leaves so that $q_{\ell_1} \leq q_{\ell_2} \leq \ldots \leq q_{\ell_N}$. ($N = \#$ of leaves.)
2. Merge leaves $\ell_a, \ell_{a+1}, \ldots, \ell_b$ into a single internal node. Split this node into two leaves with a split of the form $(x_i \leq \theta)$. Choose $\theta \in \mathbb{R}$, $i \in \mathbb{Z}$, and $1 \leq a \leq b \leq L$ that minimize $G(R)$.
3. Repeat until the change in $G(R)$ is less than Δ_{min}.

Every author seems to have their own suggestion about which nodes to merge. Our merging rule above is in the spirit of decision trees. Several rules have been proposed [15, 14, 18, 7, 2, 19], including some that are bottom-up. Mansour and McAllester's algorithm [15] is more computationally efficient than ours, has the same sample complexity guarantees, but requires fixed-width buckets of leaves. The *regression tree* learner is the same without merges, i.e. $a = b$. The size(R) is defined to be the number of nodes.

The following lemma serves the same purpose as Lemma 5 of [12] (using correlation rather than classification error).

Lemma 1. *Let $h : \mathcal{X} \to \{0, 1\}$ be a binary function. The split of ℓ into leaves $\ell_0 = \{x \in \ell | h(x) = 0\}$ and $\ell_1 = \{x \in \ell | h(x) = 1\}$ has score (reduction in $G(R)$) of $w_{\ell_0} w_{\ell_1} (q_{\ell_0} - q_{\ell_1})^2 / w_\ell = w_\ell (cor_{\mathcal{D}_\ell}(f, h))^2 var_{\mathcal{D}_\ell}(f)$.*

The proof is in Appendix A. We move the buckets of Mansour and McAllester [15] into our analysis, like [10].

Lemma 2. *The merger of leaves $\ell_a, \ell_{a+1}, \ldots, \ell_b$ with $q_{\ell_a} \leq \ldots \leq q_{\ell_b}$ into a single leaf can increase $G(R)$ by at most $(w_{\ell_{a+1}} + w_{\ell_{a+2}} + \ldots + w_{\ell_b})(q_{\ell_b} - q_{\ell_a})^2$.*

Proof. Proof by induction on b. The case $b = a$ is trivial. Let $\ell_{<b} = \ell_a \cup \ldots \cup \ell_{b-1}$ be the merger of all leaves except b. Then clearly $q_{\ell_a} \leq q_{\ell_{<b}} \leq q_{\ell_b}$. In terms of change in $G(R)$, the merger of ℓ_b and $\ell_{<b}$ is exactly the opposite of a split, and thus by Lemma 1, it increases $G(R)$ by an additional,

$$\frac{w_{\ell_b} w_{\ell_{<b}}}{w_{\ell_b} + w_{\ell_{<b}}} (q_{\ell_b} - q_{\ell_{<b}})^2 \leq w_{\ell_b} (q_{\ell_b} - q_{\ell_a})^2.$$

5 Mono-linear and mono-additive learning

Lemma 4 will show that for any mono-linear or mono-additive function, there is a threshold of a single attribute that has sufficiently large covariance with the target function. Then, using Lemmas 1 and 1 above, Lemma 5 shows that $\epsilon(R)$ will become arbitrarily small.

5.1 Existence of a correlated split

Lemma 3. *Let $u : \mathbb{R} \to \mathbb{R}$ be a monotonically nondecreasing L-Lipschitz function. Then for any distribution over $z \in \mathbb{R}$, $cov(u(z), z) \geq \sigma_u^2 / L$.*

Proof. By the bilinearity of covariance, and since $\sigma_u^2 = \mathrm{cov}(u, u)$, the statement of the lemma can be rewritten as $\mathrm{cov}(u, t) \geq 0$ for $t(z) = z - u(z)/L$. Note that $t(z)$ is nondecreasing as well. To see this, $t(z) - t(z') = z - z' - (u(z) - u(z'))/L$ which is nonnegative for $z > z'$, by definition of L-Lipschitz.

Now imagine picking $\hat{z}$ independently from the same distribution as z. Then, since $\mathrm{sign}\big(u(z) - u(\hat{z})\big) = \mathrm{sign}\big(t(z) - t(\hat{z})\big)$ always,

$$\begin{aligned} E[\big(u(z) - u(\hat{z})\big)\big(t(z) - t(\hat{z})\big)] &\geq 0 \\ E[u(z)t(z)] + E[u(\hat{z})t(\hat{z})] - E[u(z)t(\hat{z})] - E[u(\hat{z})t(z)] &\geq 0 \\ 2E[u(z)t(z)] - 2E[u(z)]E[t(z)] &\geq 0 \end{aligned}$$

The last line follows from independence and is equivalent to $\mathrm{cov}(u, t) \geq 0$. □

Lemma 4. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be of the form $f(x) = u(\sum_{i=1}^n v_i(x_i))$, where u is monotonic and L-Liptschitz, each $v_i : \mathbb{R} \to \mathbb{R}$ is a function of bounded variation V_{v_i}, and $V = \sum V_{v_i}$. Then there exists $i \in \{1, 2, \ldots, n\}$, $\diamond \in \{<, >, \leq, \geq\}$, and $\theta \in \mathbb{R}$, such that*

$$cov(I(x_i \diamond \theta), f) \geq \frac{\sigma_f^2}{LV}.$$

Proof. WLOG u is monotonically nondecreasing. A theorem from real analysis states that every function v of bounded variation V_v can be written as the sum of a monotonically nondecreasing function v_1 and a monotonically nonincreasing function v_2 with $V_v = V_{v_1} + V_{v_2}$ [17]. Thus, we can write,

$$\sum_{i=1}^{n} v_i(x_i) = \sum_{i=1}^{n} v_{i1}(x_i) + v_{i2}(x_i),$$

for *monotonic* v_{ij}, and $V = \sum_{j=1}^{j=2} \sum_{i=1}^{i=n} V_{v_{ij}}$. Let $c_{ij} = \inf_{x_i} v_{ij}(x_i)$ (so $v_{ij} : \mathbb{R} \to [c_{ij}, c_{ij} + V_{v_{ij}}]$).

Now we argue that a random threshold function of a random attribute will have large covariance. Observe that for any $z \in [0,1]$, $E_{\alpha\in[0,1]}[I(z \geq \alpha)] = z$, where α is uniform over $[0,1]$. Then, since $(v_{ij}(x_i) - c_{ij})/V_{v_{ij}} \in [0,1]$,

$$\begin{aligned} \frac{v_{ij}(x_i) - c_{ij}}{V_{v_{ij}}} &= E_{\alpha\in[0,1]}\left[I\left(\frac{v_{ij}(x_i) - c_{ij}}{V_{v_{ij}}} \geq \alpha\right)\right] \\ v_{ij}(x_i) - c_{ij} &= V_{v_{ij}} E_{\alpha\in[0,1]}\left[I(v_{ij}(x_i) \geq c_{ij} + \alpha V_{v_{ij}})\right]. \end{aligned}$$

Choose i, j from the distribution $P(i,j) = V_{ij}/V$. Then,

$$\begin{aligned} E_{i,j\leftarrow P,\alpha\in[0,1]}[I(v_{ij}(x_i) \geq c_{ij} + \alpha V_{ij})] &= \sum_{j=1}^{2}\sum_{i=1}^{n} \frac{V_{v_{ij}}}{V} E_\alpha[I(v_{ij}(x_i) \geq c_{ij} + \alpha V_{ij})] \\ &= \sum_{j=1}^{2}\sum_{i=1}^{n} \frac{v_{ij}(x_i) - c_{ij}}{V} \\ &= \frac{1}{V}\sum_{i=1}^{n} v_i(x_i) - c, \end{aligned}$$

for some constant $c \in \mathbb{R}$. By the bilinearity of covariance, the above, and the fact that covariance is immune to shifts,

$$\begin{aligned} E_{i,j,\alpha}\left[\operatorname{cov}(f, I(v_{ij}(x_i) \geq c_{ij} + \alpha V_{ij}))\right] &= \operatorname{cov}(f, E_{i,j,\alpha}[I(v_{ij}(x_i) \geq c_{ij} + \alpha V_{ij})]) \\ &= \operatorname{cov}(f, \frac{1}{V}\sum v_i(x_i) - c) \\ &= \frac{\operatorname{cov}(f, \sum v_i(x_i))}{V} \end{aligned}$$

From the previous lemma, the last quantity is at least $\sigma_f^2/(LV)$. Since the above holds in expectation, there must be an i, j, and α for which it holds instantaneously. Finally, since v_{ij} is monotonic, $I(v_{ij}(x_i) \geq c_{ij} + \alpha V_{v_{ij}}) \equiv I(x_i \diamond \theta)$ for some $\diamond \in \{<, >, \leq, \geq\}$ and $\theta \in \mathbb{R}$. □

The dependence on σ_f in the above lemma is necessary. If $\sigma_f = 0$, then $\operatorname{cov}(h, f)$ must also be 0. But the lemma does gives us the following guarantee on correlation in terms of σ_f,

$$\rho_{hf} = \frac{\operatorname{cov}(h,f)}{\sigma_h \sigma_f} \geq \frac{\sigma_f}{\sigma_h LV} \geq \frac{4\sigma_f}{LV}. \quad (1)$$

5.2 The implications for $\epsilon(R)$

Anticipating some kind of correlation boosting, we state the following lemma in terms of a guaranteed correlation $\rho(\sigma_f^2)$. In the above case $\rho(z) = 4\sqrt{z}/LV$.

Lemma 5. *Suppose $\rho : \mathbb{R} \to \mathbb{R}_+$ is a nondecreasing guarantee function such that, for each leaf ℓ, there exists a split predicate $h : \mathcal{X} \to \{0,1\}$ of correlation $cor_{\mathcal{D}_\ell}(h,f) \geq \rho(var_{\mathcal{D}_\ell}(f))$. Suppose $E[y|x] \in [0,1]$. Then with the regression graph learner with $\Delta_{min} = \epsilon^{2.5}\big(\rho(\epsilon/2)\big)^3/4$, error $\epsilon(R) \leq \epsilon$ with at most $\epsilon^{-2.5}(\rho(\frac{\epsilon}{2}))^{-3}$ splits. For the regression tree learner, after* $\exp\big(1/(4(\rho(\frac{\epsilon}{2}))^2\epsilon^2)\big)$ *splits, $\epsilon(R) \leq \epsilon$.*

Proof. By definition of leaf variance $\mathrm{var}_{\mathcal{D}_\ell}(f)$ and error $\epsilon(R)$,

$$\epsilon(R) = E_{\mathcal{D}}[\big(R(x) - f(x)\big)^2] = \sum\nolimits_\ell w_\ell E_{\mathcal{D}_\ell}[(q_\ell - f(x))^2] = \sum\nolimits_\ell w_\ell \mathrm{var}_{\mathcal{D}_\ell}(f).$$

Let N be the current number of leaves. As long as $\epsilon(R) > \epsilon$, there must be some leaf ℓ with both $w_\ell \geq 2\epsilon/N$ and $\mathrm{var}_{\mathcal{D}_\ell}(f) \geq \epsilon/2$. Otherwise, the contribution to $\epsilon(R)$ from leaves with $\mathrm{var}_{\mathcal{D}_\ell}(f) < \epsilon/2$ would be $< \epsilon/2$ and from the rest of leaves would be at most $N(2\epsilon/N)(1/4) = \epsilon/2$, since $\mathrm{var}_{\mathcal{D}_\ell}(f) \leq 1/4$ (since $f(x) \in [0,1]$).

By Lemma 1, using $\mathrm{cor}_{\mathcal{D}_\ell}(f,h) \geq \rho(\epsilon/2)$ correlation, splitting this leaf ℓ gives a reduction in $G(R)$ of at least,

$$\Delta G \geq w_\ell\big(\rho(\epsilon/2)\big)^2 \mathrm{var}_{\mathcal{D}_\ell}(f) \geq \big(\rho(\epsilon/2)\big)^2\epsilon^2/N.$$

Now $\epsilon(R) = \sigma_f^2$ at the start and decreases in each step, but never goes below 0. Also, the change in $G(R)$ is equal to the change in $\epsilon(R)$ since $\epsilon(R) = G(R) + E_{\mathcal{D}}[f(x)^2]$. Thus the total change in $G(R)$ is at most $\sigma_f^2 \leq 1/4$. In the case of regression trees, where we do splits and no merges, each split increases the number of leaves by 1. Thus, after T splits,

$$\sum_{N=1}^{T} \frac{\big(\rho(\epsilon/2)\big)^2\epsilon^2}{N} \leq \frac{1}{4}.$$

Since $\sum_1^T 1/N \geq \ln(T)$, we get the regression tree half of the lemma.

For regression graphs, say at some point there are N leaves with values $q_\ell \in [0,1]$. Now bucket the leaves by value of q_ℓ into $1/s$ intervals of width $s = \rho(\epsilon/2)\sqrt{\epsilon}/2$. For the moment, imagine merging all leaves in every bucket. Then there would be at most $1/s$ leaves, and by the above reasoning, there must be one of these merged leaves $\ell = \ell_a \cup \ell_{a+1} \cup \ldots \cup \ell_b$ with $w_\ell \geq 2\epsilon s$ and $\mathrm{var}_{\mathcal{D}_\ell}(f) \geq \epsilon/2$ (the error $\epsilon(R)$ can only have increased due to the merger). Now imagine merging *only* the leaves in this bucket and not any of the others. By Lemma 2, the increase in $G(R)$ due to the merger at most $w_\ell(q_{\ell_b} - q_{\ell_a})^2 \leq w_\ell s^2$. Using Lemma 1 as well, the total decrease in $G(R)$ is at least

$$\Delta G \geq w_\ell\big(\rho(\epsilon/2)\big)^2 \mathrm{var}_{\mathcal{D}_\ell}(f) - w_\ell s^2$$

$$\begin{aligned}&\geq w_\ell\big(\rho(\epsilon/2)\big)^2\epsilon/2 - w_\ell\big(\rho(\epsilon/2)\big)^2\epsilon/4\\ &\geq (2\epsilon s)\big(\rho(\epsilon/2)\big)^2\epsilon/4\\ &= \epsilon^{2.5}\big(\rho(\epsilon/2)\big)^3/4\end{aligned}$$

Thus there exists a merge-split that reduces $G(R)$ by at least $\epsilon^{2.5}\big(\rho(\epsilon/2)\big)^3/4$ as long as $\epsilon(R) \geq \epsilon$, and by choice of Δ_{min} we will not stop prematurely. Using that the total reduction in $G(R)$ is at most $1/4$, completes the lemma. □

We are now ready to prove the main theorem.

Proof (of Theorem 1). For part 1, we run the regression graph learning algorithm (getting exact values of p_ℓ and q_ℓ). By (1), we have $\rho(z) = 4\sqrt{z}/LV$. Since size(R) increases at most 2 per split, by Lemma 5, $\epsilon(R) \leq \epsilon$ with

$$\text{size}(R) \leq 2\epsilon^{-2.5}\left(\frac{4\sqrt{\epsilon/2}}{LV}\right)^{-3} = \epsilon^{-4}(LV)^3/8\sqrt{2} \leq \epsilon^{-4}(LV)^3/10.$$

We use $\Delta_{\text{min}} = 4\sqrt{2}\epsilon^4/(LV)^3$ to guarantee we get this far and don't run too long. Similarly, for regression trees in part 2, by Lemma 5, since $\rho(\epsilon/2)^2 = 8\epsilon/(LV)^2$, size$(R) \leq 2\exp((LV)^2/\epsilon^3/32)$. Finally, $e^{1/32} < 1.04$.

5.3 Estimations via sampling

Of course, we don't have exact values of $g_\ell = w_\ell q_\ell^2$ for each leaf, so one must use estimates. For simplicity of analysis, we use fresh samples to estimate this quantity (the only quantity necessary) for each leaf. (Though a more sophisticated argument could be used, since the VC dimension of splits is small, to argue that one large sample is enough.) It is not difficult to argue that if each estimate of g_ℓ, for each potential leaf ℓ encountered, is accurate to within, say $\tau = \Delta_{\text{min}}/10$, the algorithm will still have the same asymptotic guarantees.

While it is straightforward to estimate w_ℓ to within fixed additive tolerance, estimating q_ℓ to within fixed additive tolerance is not necessarily easy when w_ℓ is small. However, if w_ℓ is very small, then g_ℓ is also small. More precisely, if $\hat{w}_\ell < \tau/2$ and the estimate is accurate to within tolerance $\tau/10$, then we can safely estimate $\hat{g}_\ell = 0$ and still be accurate to within τ. On the other hand, if $w_\ell > \tau$, then it takes only $1/\tau$ samples to get one from leaf w_ℓ, and we can estimate q_ℓ to additive accuracy $\tau/10$ and thus g_ℓ to additive accuracy τ.

To have failure probability $1/\delta$, the number of samples required depends polynomially on $1/\epsilon, \log(n/\delta)$, and size$(R)$. The $poly - \log(n)$ dependence on n can be good in situations where there are only a few relevant attributes and LV is small.

6 Correlation boosting

Lemma 5 is clearly hiding a statement about boosting. Recall that in classification boosting, a weak learner, is basically an algorithm that output a boolean hypothesis h with accuracy $P[h(x) = f(x)] \geq 1/2 + \gamma$ (for any distribution), where $1/\gamma$ is polynomial in size(f). Then the result was that the accuracy could be "boosted" to $1 - \epsilon$ in time $poly(1/\epsilon, \text{size}(f))$. We follow the same path, replacing accuracy with correlation. We define a *weak correlator*, also similar to an "okay" learner [10].

Definition 1. *Let $\rho : [0,1] \rightarrow [0,1]$ be a nondecreasing function. An efficient ρ weak correlator for concept $\mathcal{C}$ is an algorithm (that takes inputs δ and samples from $\mathcal{D}$) such that, for any $\delta > 0$, any distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$ with $\mathcal{Y} = [0,1]$ and $f(x) = E[y|x] \in \mathcal{C}$, with probability $1 - \delta$ it outputs a hypothesis $h : \mathcal{X} \rightarrow \mathbb{R}$ with $\rho_{fh} \geq \rho(\sigma_f^2)$. It must run in time polynomial in $1/\delta, 1/\sigma_f^2$, and size(f), and $1/\rho$ must be polynomial in $1/\sigma_f^2$, and size(f).*

The algorithm is very similar. We start with a single leaf ℓ. Repeat:

1. Sort leaves so that $q_{\ell_1} \leq q_{\ell_2} \leq \ldots \leq q_{\ell_N}$. ($N = \#$ of leaves.)
2. For each $\ell_a, \ell_{a+1}, \ldots, \ell_b$, run the weak correlator (for a maximum of T time) on the distribution $\mathcal{D}_{\ell_{ab}}$ where ℓ_{ab} would be the merger of $\ell_a \ldots \ell_b$. If it terminates, the output will be some predictor $h_{ab} : \mathcal{X} \rightarrow \mathbb{R}$. Choose $1 \leq a \leq b \leq N$ and θ such that the merge-split of $\ell_a \ldots \ell_b$ with split $(h_{ab}(x) \geq \theta)$ gives the smallest $G(R)$.
3. Repeat until the change in $G(R)$ is less than $\Delta_{\min}$.

The point is that such a weak correlator can be used to get an arbitrarily accurate regression graph R with $\epsilon(R) \leq \epsilon$ for any $\epsilon > 0$ (efficiently in $1/\epsilon$). Appendix C shows,

$$\rho_{Rf} = \sqrt{1 - \frac{\epsilon(R)}{\sigma_f^2}} \geq 1 - \frac{\epsilon(R)}{\sigma_f^2}.$$

Thus, reducing $\epsilon(R)$ to arbitrary inversely polynomial ϵ is equivalent to "boosting" correlation from inversely polynomial to $1 - \epsilon/\sigma_f^2$. Appendix C also shows $\rho_{Ry} = \rho_{Rf}\rho_{fy}$. Thus ρ_{Ry}, the correlation coefficient reported in so many statistical studies, also becomes arbitrarily close to ρ_{fy}, the optimal correlation coefficient.

Theorem 2. *Given a ρ weak correlator, with probability $1-\delta$, the learned regression graph R has $\epsilon(R) \leq \epsilon$, with runtime polynomial in $1/\epsilon, 1/\delta$, and $1/\rho(\epsilon/2)$.*

Proof (sketch). The proof follows that of Lemma 5. There are three differences.

First, we must have a maximum time restriction on our weak correlators. If a leaf has tiny $\text{var}_{\mathcal{D}_\ell}(f)$, then the weak correlator will have to run for a very long time, e.g. if in one leaf there are only two types of x, one with $f(x) = 0.5$ and the other with $f(x) = 0.49999$, then it could easily take the weak correlator a long time to correlate with them. However, as seen in the proof of Lemma 5, we can safely ignore all leaves with $\text{var}_{\mathcal{D}_\ell}(f) < \epsilon/2$. Since we can't identify them, we simply stop each one after a certain amount of time running, for if we've gone longer than T time (which depends on the runtime guarantees of the weak correlator, but is polynomial in $1/\epsilon$ and size(f)), then we know that leaf has low variance anyway.

Second, we estimate weights and values for each different leaf with fresh samples. This makes the analysis simple.

Third, h_{ab} is not necessarily a boolean attribute. Fortunately, there is some threshold so that $I(h_{ab}(x) \geq \theta)$ also has large correlation. The arguments of Lemma 5 show there exists an h_{ab} with $\rho_{h_{ab}f} \geq \rho(\epsilon/2)$ and $\sigma_f^2 \geq \epsilon/2$, which are polynomial in $1/\epsilon$ and size(f) by definition of weak correlator. Lemma 6 in Appendix B implies that there will be some such threshold indicator h with,

$$\rho_{hf} > \frac{\rho_{h_{ab}f}}{2 + 2\sqrt{2\log(2/(\rho_{h_{ab}f}\sigma_f))}},$$

where quantities are measured over $\mathcal{D}_{\ell_{ab}}$. This is nearly $\rho_{h_{ab}f}/2$ and its reciprocal is certainly inverse polynomial in $1/\epsilon$ and size(f). □

7 Conclusions

While generalized additive models have been studied extensively in statistics, we have proven the first efficient learning guarantee, namely that regression graphs efficiently learn a generalized additive model (with a monotonic link function) to within arbitrary accuracy.

In the case of classification boosting, most boosting algorithms are parametric and maintain a linear combination of weak hypotheses. In fact, if a function is boostable, then it is writable as a linear threshold of weak hypotheses (just imagine running AdaBoost sufficiently long). We have shown that the class of boostable functions in the real valued setting is much richer. It includes at least the mono-linear functions of base hypotheses.

It would be especially nice to remove the dependence on the Lipschitz constant. (The bounded variation condition does not seem too restrictive.) For the related problem of learning a linear threshold function with uniform classification noise, Blum et. al. [4] were able to remove the dependence on a margin that was in Bylander's original work [5].

It would be nice to relax the assumption that $f(x) = E[y|x]$ is exactly distributed according to a mono-additive function. While it seems difficult to provably get as far as one can get in linear regression, i.e. find the best fit linear predictor, it may be possible to do something in between. For any given distribution there are often several mono-additive functions $f(x)$ that are *calibrated* with the distribution, i.e. $f(x) = E[y|f(x)]$. For example, the historical probability of white winning in a game of chess is almost certainly monotonic in the quantity $w_1 \cdot x =$ (#white pieces)$-$(#black pieces). But it should also be monotonic in terms of something like $w_2 \cdot x =$ (#white pawns $+ \ldots +$ 3#white bishops) $-$ (#black pawns$+\ldots+$3#black bishops). Can one do as well as the best calibrated mono-additive function without assumptions on $\mathcal{D}$?

7.1 Acknowledgments

I would like to thank Marc Coram for identifying the model as a generalized additive model (before it was too late), Ehud Kalai for suggesting the use of a Lipschitz condition, David McAllester and Rob Schapire for insightful discussions, and the anonymous referees for pointing out the disorganization.

References

1. E. Baum. A polynomial time algorithm that learns two hidden unit nets. *Neural Computation* 2:510-522, 1991.
2. L. Bahl, P. Brown, P. deSouze, and R. Mercer. A Tree-based statistical language model for natural language speech recognition. *IEEE Transactions on Acoustics, Speec, and Signal Processing,* 37:1001-1008, 1989.
3. J. Aslam and S. Decatur. Specification and simulation of statistical query algorithms for efficiency and noise tolerance. *Journal of Computer and System Sciences*, 56:191–208, 1998.
4. A. Blum, A. Frieze, R. Kannan, and S. Vempala. A polynomial time algorithm for learning noisy linear threshold functions. *Algorithmica*, 22(1/2):35–52, 1997.
5. T. Bylander. Polynomial learnability of linear threshold approximations. In *Proceedings of the Sixth Annual ACM Conference on Computational Learning*, 297–302, 1993.
6. L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Wadsworth International Group, 1984.
7. P. Chou. *Applications of Infromation Theory to Pattern Recognition and the Design of Decision Trees and Trellises.* PhD thesis, Department of Electrical Engineering, Stanford University, June 1988.
8. J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: A statistical view of boosting. The Annals of Statistics, 28:337 – 374, 2000.
9. T. Hastie and R. Tibshirani. *Generalized Additive Models.* London: Chapman and Hall, 1990.
10. A. Kalai and R. Servedio. Boosting in the presence of Noise. *Proceedings of the thirty-fifth ACM symposium on theory of computing,* pages 195–205, 2003.
11. M. Kearns. Efficient noise-tolerant learning from statistical queries. *Journal of the ACM*, 45(6):983–1006, 1998.
12. M. Kearns and Y. Mansour. On the boosting ability of top-down decision tree learning algorithms. *Journal of Computer and System Sciences*, 58(1):109–128, 1999.
13. M. Kearns and R. Schapire. Efficient distribution-free learning of probabilistic concepts. *Journal of Computer and Systems Sciences*, 48:464-497, 1994.
14. R. Kohavi. Wrappers for Performance Enhancement and Oblivious Decision Graphs. Ph.D. dissertation, Comput. Sci. Depart., Stanford Univ., Stanford, CA, 1995.
15. Y. Mansour and D. McAllester. Boosting using branching programs. *Journal of Computer and System Sciences*, 64(1):103–112, 2002.
16. P. McCullagh and J. Nelder. *Generalized Linear Models,* Chapman and Hall, London, 1989.
17. H. Royden. *Real Analysis*, 3rd edition. Macmillan, New York, 1988.

18. J. Oliver. Decision graphs – an extension of decision trees. In *Proceedings of the Fourth International Workshop on Artificial Intelligence and Statistics*, pp. 334-350, 1993.
19. J. Oliver, D. Dowe, and C. Wallace. Inferring decision graphs using the minimum message length principle. In *Proceedings of the 5th Austrailian Conference on Artificial Intelligence*, pp. 361-367, 1992.
20. R. Schapire. The strength of weak learnability. *Machine Learning*, 5(2):197–227, 1990.
21. M. Shipp, D. Harrington, J. Anderson, J. Armitage, G. Bonadonna, G. Brittinger, et al. A predictive model for aggressive non-Hodgkin's lymphoma. The International Non-Hodgkin's Lymphoma Prognostic Factors Project. *New England Journal of Medicine* 329(14):987-94, 1993.
22. L. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134-1142, 1984.
23. B. Zadrony and C. Elkan. Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 609–616, 2001.

A Proof of Lemma 1

Using the facts that $w_\ell = w_{\ell_0} + w_{\ell_1}$ and $q_\ell = (w_{\ell_0} q_{\ell_0} + w_{\ell_1} q_{\ell_1})/w_\ell$, the change in G is,

$$\begin{aligned}\Delta G &= w_{\ell_0} q_{\ell_0}^2 + w_{\ell_1} q_{\ell_1}^2 - w_\ell \left(\frac{w_{\ell_0} q_{\ell_0} + w_{\ell_1} q_{\ell_1}}{w_\ell} \right)^2 \\ &= \frac{(w_{\ell_0} + w_{\ell_1})(w_{\ell_0} q_{\ell_0}^2 + w_{\ell_1} q_{\ell_1}^2) - (w_{\ell_0} q_{\ell_0} + w_{\ell_1} q_{\ell_1})^2}{w_\ell} \\ &= \frac{w_{\ell_0} w_{\ell_1} (q_{\ell_0} - q_{\ell_1})^2}{w_\ell}.\end{aligned}$$

Next,

$$\begin{aligned}\mathrm{cov}_{\mathcal{D}_\ell}(f,h) &= E_{\mathcal{D}_\ell}[f(x)h(x)] - E_{\mathcal{D}_\ell}[f(x)] E_{\mathcal{D}_\ell}[h(x)] \\ &= \frac{w_{\ell_1}}{w_\ell} q_{\ell_1} - \frac{w_{\ell_0} q_{\ell_0} + w_{\ell_1} q_{\ell_1}}{w_\ell} \frac{w_{\ell_1}}{w_\ell} \\ &= \frac{(w_{\ell_0} + w_{\ell_1}) w_{\ell_1} q_{\ell_1} - (w_{\ell_0} q_{\ell_0} + w_{\ell_1} q_{\ell_1}) w_{\ell_1}}{w_\ell^2} \\ &= \frac{w_{\ell_0} w_{\ell_1} (q_{\ell_1} - q_{\ell_0})}{w_\ell^2}.\end{aligned}$$

Meanwhile, since h is boolean,

$$\mathrm{var}_{D_\ell}(h) = P_{\mathcal{D}_\ell}[h(x)=0] P_{\mathcal{D}_\ell}[h(x)=1] = \frac{w_{\ell_0}}{w_\ell} \frac{w_{\ell_1}}{w_\ell} = \frac{w_{\ell_0} w_{\ell_1}}{w_\ell^2}$$

Finally, $\Delta G = w_\ell \mathrm{cov}_{D_\ell}(f,h)^2 / \mathrm{var}_{D_\ell}(h) = w_\ell \mathrm{cor}_{D_\ell}(f,h)^2 \mathrm{var}_{D_\ell}(f)$. □

B Thresholds

Lemma 6. *Let $u \in [0,1]$ be a random variable and $v \in \mathbb{R}$ be a positively correlated random variable. Then there exists some threshold $t \in \mathbb{R}$ such that the indicator random variable $v_t = I(v \geq t)$ has correlation near $\rho_{uv} > 0$,*

$$\rho_{uv_t} > \frac{\rho_{uv}}{2 + 2\sqrt{2\log(2/(\rho_{uv}\sigma_u))}}.$$

Proof. WLOG let v be a standard random variable, i.e. $\mu_v = 0$ and $\sigma_v = 1$. The main idea is to argue, for $\tau = 2/\sigma_{uv}$, that

$$\int_{-\tau}^{\tau} \sigma_{uv_t} dt > \frac{\sigma_{uv}}{2 + 2\sqrt{2\log(1/\tau)}} \int_{-\tau}^{\tau} \sigma_{v_t} dt. \tag{2}$$

This implies that there exists a $t \in [-\tau, \tau]$ for which the above holds instantaneously, i.e.,

$$\sigma_{uv_t} > \frac{\sigma_{uv}}{2 + 2\sqrt{2\log(1/\tau)}} \sigma_{v_t}$$
$$\frac{\sigma_{uv_t}}{\sigma_u \sigma_{v_t}} > \frac{\sigma_{uv}}{\sigma_u(2 + 2\sqrt{2\log(1/\tau)})}$$

The above is equivalent to the lemma for $\tau = 2/\sigma_{uv} = 2/(\rho_{uv}\sigma_u)$. Thus it suffices to show (2).

First, a simple translation can bring $\mu_u = 0$. This will not change any correlation, so WLOG let us assume that $\mu_u = 0$ and that $u \in (-1, 1)$. This makes calculations easier because now $\sigma_{uv_t} = E[uv_t] - \mu_u \mu_{v_t} = E[uv_t]$ for all t. Define the random variable w by,

$$w = \int_{-\tau}^{\tau} v_t dt - \tau = \begin{cases} \tau & \text{if } v \geq \tau \\ v & \text{if } v \in (-\tau, \tau) \\ -\tau & \text{if } v \leq -\tau \end{cases}$$

Then we have, by linearity of expectation,

$$\int_{-\tau}^{\tau} \sigma_{uv_t} dt = \int_{-\tau}^{\tau} E[uv_t] dt = E\left[u \int_{-\tau}^{\tau} v_t dt\right] = E[u(w + \tau)] = E[uw].$$

Next, notice that $|v - w| \leq |v|$ and, if $v - w \neq 0$ then $|v| \geq \tau$. This means that $|v - w| \leq v^2/\tau$. Consequently, $E[u(v - w)] < E[|v - w|] \leq E[v^2/\tau] = 1/\tau$, so,

$$\int_{-\tau}^{\tau} \sigma_{uv_t} dt = E[uw] = E[uv] - E[u(v - w)] > E[uv] - \frac{1}{\tau} = \frac{\sigma_{uv}}{2}. \tag{3}$$

For the second part, by the Cauchy-Schwartz inequality,

$$\int_1^{\tau} \sigma_{v_t} dt = \int_1^{\tau} \sigma_{v_t} \sqrt{t} \cdot \frac{1}{\sqrt{t}} dt \leq \sqrt{\int_1^{\tau} \sigma_{v_t}^2 t dt \int_1^{\tau} \frac{1}{t} dt}.$$

Now, $\sigma_{v_t}^2 = E[v_t^2] - E[v_t]^2 = P[v \geq t]P[v < t]$. The above is at most:

$$\sqrt{\int_1^\tau P[v \geq t]tdt \cdot \log(1/\tau)} = \sqrt{\int_1^{\tau^2} P[v \geq \sqrt{y}]\frac{1}{2}dy \cdot \log(1/\tau)}.$$

For a nonnegative random variable A, $E[A] = \int_0^\infty P[A \geq y]dy$. Thus

$$\int_1^{\tau^2} P[v \geq \sqrt{y}]dy \leq \int_0^\infty P[v^2 \geq y]dy = E[v^2] = 1.$$

By symmetry, we get

$$\int_{-\tau}^{\tau} \sigma_{v_t} dt \leq \int_{-1}^{1} \sigma_{v_t} dt + 2\sqrt{\frac{\log(1/\tau)}{2}} \leq 1 + \sqrt{2\log(1/\tau)}. \tag{4}$$

Equations (4) and (3) imply (2), and we are done. □

C Facts about regression graphs

It is easy to see that $\mu_y = \mu_f = \mu_R = \sum_\ell w_\ell q_\ell$. Also,

$$\begin{aligned}\epsilon(R) &= E[f(x)^2] + E[R(x)^2] - 2E[f(x)R(x)] \\ &= E[f(x)^2] - E[R(x)^2] = \sigma_f^2 - \sigma_R^2 \\ &= E[f(x)^2] - \sum_\ell w_\ell(q_\ell^2 - 2q_\ell^2) \\ &= \sigma_f^2 + \mu_f^2 - \sum_\ell w_\ell q_\ell^2.\end{aligned}$$

Since $\sum w_\ell = 1$, we have $\sum aw_\ell + bw_\ell q_\ell$ is constant across graphs. So $\sum aw_\ell + bw_\ell q_\ell - cw_\ell q_\ell^2$ for $c > 0$ as an objective function is equivalent to using $-\sum w_\ell q_\ell^2$. Finally, $\text{cov}(R, f) = \sigma_R^2 = \sum w_\ell q_\ell^2 - \mu_R^2 = \sum w_\ell q_\ell^2 - \mu_f^2$, implying that $\rho_{rf} = \sigma_R/\sigma_f = \sqrt{\sum w_\ell q_\ell^2 - \mu_f^2}/\sigma_f = \sqrt{1 - \epsilon(R)/\sigma_f^2}$