# Gradient Regularized Budgeted Boosting

**Zhixiang (Eddie) Xu**
Washington University in St. Louis
St. Louis, MO 63130
xuzx@cse.wustl.edu

**Matt J. Kusner**
Washington University in St. Louis
St. Louis, MO 63130
mkusner@wustl.edu

**Kilian Q. Weinberger**
Washington University in St. Louis
St. Louis, MO 63130
kilian@wustl.edu

**Alice Zheng**
Microsoft Research
alicez@microsoft.com

## Abstract

As machine learning transitions increasingly towards real world applications controlling the test-time cost of algorithms becomes more and more crucial. Recent work, such as the Greedy Miser and Speedboost, incorporate test-time budget constraints into the training procedure and learn classifiers that provably stay within budget (in expectation). However, so far, these algorithms are limited to the supervised learning scenario where sufficient amounts of labeled data are available. In this paper we investigate the common scenario where labeled data is scarce but unlabeled data is available in abundance. We propose an algorithm that leverages the unlabeled data (through Laplace smoothing) and learns classifiers with budget constraints. Our model, based on gradient boosted regression trees (GBRT), is, to our knowledge, the first algorithm for *semi-supervised budgeted learning*.

## 1   Introduction

The number and variety of real-world settings in which machine learning is used is astounding, with settings such as web-search ranking [1], predicting hepatitis B [2], and ad placement [3]. Classification in these real-world settings many times is constrained by (a) the cost to extract features and (b) the (CPU-)cost to evaluate the classifier. The notion of feature cost may be quite diverse. Often this cost is CPU-time (*i.e.,* it may take 10 milliseconds to load user's personal search history) or in medical settings the cost of features may be the price of patient tests. Reducing CPU-cost is particularly crucial for application domains on embedded devices [4] and very large-scale industrial applications, where it directly translates to energy usage and $CO_2$ emissions. Recently, the field of *budgeted learning* has begun to address these costs by training a classifier to stay within a cost budget during test-time [5; 6; 7; 8; 9; 10; 11; 12; 13; 14; 15; 16; 17].

Gradient Boosted Regression Trees (GBRT), originally proposed in 2001 [18], inspires the majority of budgeted learning models [6; 19; 8; 10; 12; 20; 15; 21]. The additive nature of GBRT classifiers lends itself naturally to this budget setting: The evaluation of a boosted tree classifier can be stopped prematurely if the result is already sufficiently accurate or unlikely to lead to promising results [22]. This provides a natural trade-off mechanism between classifier accuracy and energy usage. Furthermore, in several real-world learning settings, and especially in web-search ranking, GBRT is now generally regarded as the state-of-the-art learning approach. Notably, it was used by all eight winning teams of the 2010 Yahoo! Learning to Rank competition (across both competition tracks) [23].

For most of these real-world applications, obtaining labeled data is expensive and may require expert knowledge, but unlabeled data is often available in abundance. For example in web-search ranking,

query-page pairs need to be labeled by hand from experts for each country in which a search engine operates [24], but unlabeled data can be downloaded at virtually no cost.

Semi-supervised learning [25] aims to incorporate unlabeled data during training. There are multiple paradigms, but one of the most successful is Laplacian (/Manifold) regularization [26]. Specifically, Laplacian regularization enforces predictions of classifiers to vary smoothly along the geodesics of the data manifold of the unlabeled data. Due to its simplicity, incorporating Laplacian regularization with many existing classification algorithms is straight-forward, and the resulting methods [26; 27; 28] all show significant classification improvement by employing unlabeled data. Although Laplacian regularization can also be incorporated into GBRT, it necessarily leads to very large model sizes, which makes it impractical for the purpose of budgeted learning.

In this paper, we propose a novel technique to incorporate Laplacian regularization alongside budget regularization [10] via *gradient regularization*. We demonstrate that our method leads to cheap compact models, requiring few trees to leverage information from unlabeled data to improve generalization performance in cases where labeled data is sparse. As far as we know, the resulting algorithm, *Gradient Regularized Budgeted Boosting (GRBB)*, is the first successful combination of budgeted learning and semi-supervised learning. We evaluate GRBB on one synthetic and six data sets to demonstrate how it generates compact models with few trees. We then demonstrate the efficacy of GRBB on two real-world budgeted learning datasets: the Yahoo! LTR budgeted learning benchmark [29] and the Scene15 image classification dataset [30]. The results demonstrate that its reduction in model size directly translates into substantial cost savings at test-time.

## 2  Background

Our data consists of a small number of input vectors $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \in \mathcal{R}^d$ with labels $\{y_1, \ldots, y_n\} \in \mathcal{Y}$. For the sake of clarity, we focus on binary classification scenario, where $\mathcal{Y} \in \{0, 1\}$. In addition to labeled inputs, we also have a larger set of unlabeled inputs $\{\mathbf{x}_{n+1}, \ldots, \mathbf{x}_{n+m}\} \in \mathcal{R}^d$, with $n \ll m$. Throughout this paper we use $\mathbf{x}$ to refer to an arbitrary *labeled* input and $\bar{\mathbf{x}}$ to refer to an arbitrary *unlabeled* input.

**Budgeted Learning.** In budgeted learning [10; 19] each feature $\alpha$ has an on-demand extraction cost $c_\alpha$. Learning a classifier $H(\cdot)$ incurs a feature extraction cost $c_f(H) > 0$ and a classifier evaluation cost $c_e(H) \geq 0$. The goal of budgeted learning is to constrain this total classifier cost to be under a pre-specified budget $B$. Because of the small number of training labels our approach will be to integrate Laplacian regularization into a GBRT-based budgeted learning algorithm, the Greedy Miser [10]. We give a background description for each of these methods.

**Gradient Boosted Regression Trees.** Gradient Boosted Regression Trees (GBRT) [18] learns a classifier $H : \mathcal{R}^d \to \mathcal{Y}$ that (approximately) minimizes a continuous and differentiable loss function $\ell(\cdot)$.[1] The classifier is an additive ensemble of CART trees, $H(\mathbf{x}) = \sum_{t=1}^{T} \eta_t h_t(\mathbf{x})$, where $\eta_t > 0$ is the learning rate and $h_t(\cdot)$ are limited depth regression trees [31]. In iteration $t+1$, GBRT greedily adds a new tree to approximately minimize $\ell(H + \eta h_{t+1})$ with the CART algorithm [31].

**Greedy Miser.** Xu et al. [10] proposed a modified version of GBRT that selects features within a cost budget. Specifically, they define an indicator function $\mathcal{F}_\alpha(h_t)$ that is 1 if tree $h_t$ uses feature $\alpha$ and 0 otherwise. They propose an iterative algorithm that at each step $t$ selects a CART tree $h_t(\cdot) \in \mathcal{H}$ for the current ensemble $H_{t-1}(\cdot) = \sum_{j=1}^{t-1} \eta_j h_j(\cdot)$ by the following minimization,

$$\min_{h_t \in \mathcal{H}} \sum_{i=1}^{n} \left( -\frac{\partial \ell}{\partial H_{t-1}(\mathbf{x}_i)} - h_t(\mathbf{x}_i) \right)^2 + \mu \sum_{\alpha=1}^{d} c_\alpha \mathcal{F}_\alpha(h_t) \tag{1}$$

where $\mu$ controls the trade-off between accuracy and feature extraction cost (if we remove the right-hand term we arrive at the original GBRT formulation). The classifier evaluation cost is simply proportional to the number of CART trees $h_t$ used in $H$.

**Laplacian regularization.** Belkin and Niyogi [26] propose Laplacian regularization as a powerful method to improve classification accuracy using unlabeled inputs. The algorithm takes the underlying data manifold in account and encourages the classifier to make similar predictions on unlabeled

---

[1]We use the logistic loss throughout.

inputs that are close to each other. Let $k_{ij} = 1$ be a measure of similarity (*e.g.* $k_{ij} = 1$ if $\mathbf{x}_i, \mathbf{x}_j$ are close and $k_{ij} = 0$ otherwise). The algorithm enforces locally similar predictions with a regularization term $\sum_{i=1}^{n+m} \sum_{j=i+1}^{n+m} k_{ij} \left[ H(\mathbf{x}_i) - H(\mathbf{x}_j) \right]^2$. This regularization term can be represented in matrix form using the Laplacian matrix $\mathbf{L} \subseteq \mathcal{R}^{(n+m)\times(n+m)}$ [32]. Specifically, Laplacian regularization algorithms solve the following,

$$\min_H \ell(H(\mathbf{X})) + \frac{\lambda}{2}[H(\mathbf{X}), H(\bar{\mathbf{X}})]\mathbf{L}[H(\mathbf{X}), H(\bar{\mathbf{X}})]^\top, \tag{2}$$

where $\mathbf{X} \subseteq \mathcal{R}^{d\times n}$, $\bar{\mathbf{X}} \subseteq \mathcal{R}^{d\times m}$ are matrices of the labeled and unlabeled input features and $H(\mathbf{X})$ is a row vector of $n$ predictions on $\mathbf{X}$ (similarly for $H(\bar{\mathbf{X}})$). Finally, $\lambda$ is the regularization trade-off parameter. The regularization term encourages predictions to vary smoothly over all inputs (labeled and unlabeled), and thus avoids over-fitting to labeled inputs when they are few. Belkin and Niyogi [26] demonstrate that Laplacian regularized algorithms (LapSVM and LapRLS) are competitive with a variety of other semi-supervised learning models.

Solving the optimization in (2) is commonly done using gradient descent. Let $\mathcal{L}(\cdot)$ denote the entire objective in eq. (2) (loss + Laplacian regularization). By encouraging the predictions of unlabeled inputs $\bar{\mathbf{x}}$ to be similar to nearby labeled inputs $\mathbf{x}$ we are encouraging their gradients $\frac{\partial \mathcal{L}}{\partial H_t(\bar{\mathbf{x}})}$ and $\frac{\partial \mathcal{L}}{\partial H_t(\mathbf{x})}$ to be similar. In effect, the gradient *propagates* out from labeled inputs to unlabeled inputs.

# 3 Laplacian Regularized Boosting

We begin by considering the simple combination of Laplacian regularization with GBRT via eq. (2) (we first examine GBRT without the feature cost modifications of Greedy Miser for the sake of exposition). We show through experiments on a synthetic dataset that this leads to large boosted models, having a significant evaluation cost. Because Greedy Miser does not explicitly control evaluation cost, we propose a method to substantially reduce the evaluation cost via *gradient regularization*, which allows the model to sucessfully incorporate unlabeled inputs in a budgeted setting.

## 3.1 Gradient propagation

Imagine we combine Laplacian regularization with GBRT by using the proposed objective function in eq. (2). GBRT would select new CART trees $h_t$ to match the gradient of this Laplacian regularized objective $\mathcal{L}(\cdot)$, via minimizing the left-hand term of eq. (1). As described above, we would like the gradient of labeled inputs to propagate to unlabeled inputs. However, if the labeled set is small relative to the unlabeled set this propagation may take a long time. For other Laplacian regularized models, *e.g.* LapSVM and LapRLS [26], this may be only a minor nuisance in the form of increased training time, but for LapGBRT this leads to *very large models*. Gradient boosting stores the entire optimization path from initialization to final solution as part of the model and *each tree represents one gradient step*. If the initialization is far from the final solution, this results in many CART trees. For the case of budgeted learning this means a significant evaluation cost.

Figure 1 (3rd row) shows the performance of LapGBRT on a simple simulation dataset. The dataset contains two classes (red and blue) and contains two labeled inputs (shown in all plots). We construct the Laplacian matrix $\mathbf{L}$ from 9 nearest neighbors for all inputs and cross-validate over the learning rate $\eta$ to ensure fastest convergence. We plot results for different numbers of learned CART trees. For this small dataset LapGBRT is eventually able to achieve $97.8\%$ accuracy after 1000 trees. This is a large number of trees for a dataset with obvious manifold structure.

# 4 Gradient Regularized Boosting

To overcome this problem, we encourage gradient propagation by explicitly *regularizing* the gradients of nearby inputs to be similar. The idea is to reconsider Laplacian regularization as such,

$$[H(\mathbf{X}), H(\bar{\mathbf{X}})]\mathbf{L}[H(\mathbf{X}), H(\bar{\mathbf{X}})]^\top \implies [\nabla_L, \nabla_U]\mathbf{L}[\nabla_L, \nabla_U]^\top.$$

where $\nabla_L$ and $\nabla_U$ are gradients of labeled and unlabeled inputs, which we define below. Intuitively we are push neighboring inputs to have similar gradients *immediately*. To achieve this we first define
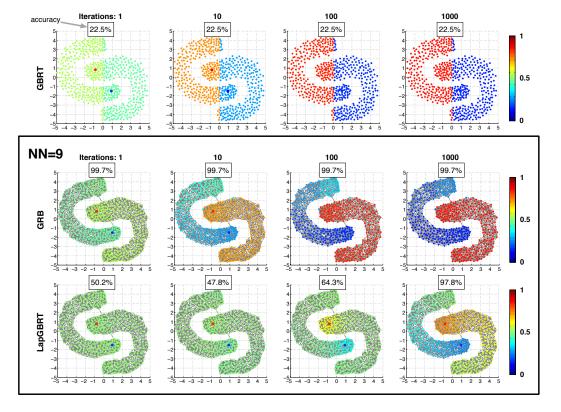
3

Figure 1: GRBB algorithm is evaluated on a synthetic data set (see text for details).

$\nabla_L$ as the gradient of the loss $\ell(\cdot)$ with Laplacian regularization *for labeled inputs only*,

$$\nabla_L = \frac{\partial \ell}{\partial H^t(\mathbf{x})} + \frac{\lambda}{2} \frac{\partial [H^t(\mathbf{x}), H^t(\bar{\mathbf{x}})]^\top \mathbf{L}[H^t(\mathbf{x}), H^t(\bar{\mathbf{x}})]}{\partial H^t(\mathbf{x})}. \tag{3}$$

We would like to encourage close unlabeled inputs to have similar gradients. We solve for the gradients of unlabeled inputs $\nabla_U$ by minimizing the difference between them and nearby labeled gradients, via the Laplacian matrix,

$$\min_{\nabla_U} \big[\nabla_L, \nabla_U\big] \mathbf{L} \big[\nabla_L, \nabla_U\big]^\top.$$

This is a quadratic function of $\nabla_U$, and we can solve for $\nabla_U$ in closed-form by applying first order conditions,

$$\frac{\partial \big[\nabla_L, \nabla_U\big] \mathbf{L} \big[\nabla_L, \nabla_U\big]^\top}{\partial \nabla_U} = 0. \tag{4}$$

Note that the Laplacian matrix $\mathbf{L}$ can be decomposed into four parts,

$$\mathbf{L} = \left( \begin{array}{cc} \mathbf{L}_{LL} & \mathbf{L}_{LU} \\ \mathbf{L}_{LU}^\top & \mathbf{L}_{UU} \end{array} \right),$$

where $\mathbf{L}_{LL}$ is the Laplacian sub-matrix between labeled and labeled inputs, $\mathbf{L}_{LU}$ is between labeled and unlabeled, and $\mathbf{L}_{UU}$ is between unlabeled inputs. Using the decomposed Laplacian matrix, the solution of $\nabla_U$ from eq. (4) is a least squares solution:

$$\nabla_U = -(\mathbf{L}_{UU})^{-1} \mathbf{L}_{LU}^\top \nabla_L. \tag{5}$$

After solving $\nabla_U$, we use Greedy Miser to select CART trees $h_t$ to match the gradient at the prior iteration over all inputs $\nabla^{t-1} = \big[\nabla_L^{t-1}, \nabla_U^{t-1}\big]$,

$$\min_{h_t \in \mathcal{H}} \sum_{i=1}^{n+m} \Big( -\nabla_i^{t-1} - h_t(\mathbf{x}_i) \Big)^2 + \mu \sum_{\alpha=1}^{d} c_\alpha \mathcal{F}_\alpha(h_t) \tag{6}$$

4

---
**Algorithm 1** GRBB in pseudo-code.
---
1: Input: labeled data $\{\mathbf{x}_i, y_i\}$, unlabeled data $\{\bar{\mathbf{x}}_i\}$, learning rate $\eta$, iterations $T$.
2: Initialize predictions $H = 0$.
3: Construct Laplacian matrix $\mathbf{L}$ using $[\mathbf{x}, \bar{\mathbf{x}}]$.
4: **for** $t = 1$ **to** $T$. **do**
5:     Compute the gradient of labeled inputs $\nabla_L$ using eq. (3).
6:     Solve for $\nabla_U$ using eq. (5).
7:     Use Greedy Miser (1) to approximate gradients $[\nabla_L, \nabla_U]$ via $h_t$.
8:     Update $H = H + \eta h_t$.
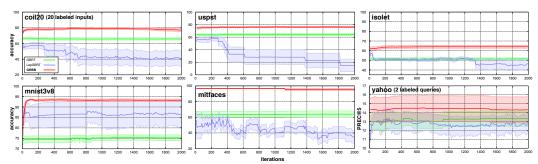9: **end for**
10: Return $H$.
---



Figure 2: GBRT, LapGBRT, and GRBB accuracy (PREC@5 for yahoo) for each tree added.

We call our approach Gradient Regularized Budgeted Boosting (GRBB), which we summarize in Algorithm 1. Note that although step 5 requires matrix inversion described in eq. (5), the inversion is only performed once. Since $-(\mathbf{L}_{UU})^{-1}\mathbf{L}_{LU}^{\top}$ is fixed once we obtain the Laplacian matrix, we only need to invert the matrix once and cache it for following iterations.

## 4.1   Reducing evaluation cost

To demonstrate that GRBB reduces the number of trees necessary, we apply GRBB to the same simulation dataset as we did LapGBRT. In this case the features have zero cost and so the second feature-cost term in eq. (6) drops out. Figure 1 (2nd row) shows the results of GRBB. By regularizing the gradients of unlabeled inputs GRBB is able to propagate the gradients of labeled inputs to the unlabeled inputs *using a single tree*. This results in near-perfect accuracy on this simple dataset, notably achieving better accuracy immediately with a single tree than LapGBRT after $1,000$ trees. The prediction confidence of GRBB only increases as more trees are added. Importantly, GRBB builds an accurate model that incorporates unlabeled inputs using a fraction of the evaluation cost of LapGBRT.

We also evaluate GBRT, LapGBRT, and GRBB on 6 real datasets (again without feature costs for simplicity): *COIL20*: classifying images into 20 object categories (we binarized the labels into groups 1-10 and 11-20). *USPS*: digit recognition from images, processed according to [26]. *Isolet*: identifying spoken English letters from audio (we binarized the data into groups 1-13 and 14-26). *MNIST3v8*: recognizing digits 3 from 8, derived from the MNIST dataset. *MITfaces*: detecting faces vs. non-faces in $19 \times 19$ gray-scale images. *Yahoo*: a binary web-search ranking dataset [19]. We train all models using 20 labeled inputs (or 2 labeled queries for Yahoo). We select hyperparameters using using Bayesian optimization (BO) [33], with expected improvement as the acquisition function (GBRT: learning rate $\eta$, tree depth; LapGBRT & GRBB: $\eta$, tree depth, Laplaican regularization trade-off $\lambda$, and number of nearest neighbors in $\mathbf{L}$). We use accuracy as the performance metric (or PREC@5 for Yahoo [19]). For all datasets, GRBB outperforms GBRT and LapGBRT. Better, with the addition of gradient regularization GRBB is able to achieve near top accuracy *in under 200 trees*. These results show that GRBB is well-equipped to reduce the evaluation cost without sacrificing accuracy on small labeled training sets.

# 5 Making Reliable Predictions

Different from fully-supervised budgeted learning we have an additional free parameter: the number of labeled inputs. Even though we employ unlabeled data to help curb overfitting, a larger number of labeled inputs serves to reduce *prediction variance*. For the practitioner, given a cost budget $B$, it may be crucial to decide how large the labeled set should be so that the learned model is close, in some sense, to the model trained on *infinitely many inputs*. In this section, we introduce a simple method for computing a lower bound on the variance of the prediction probabilities of GRBB: $\sigma(\hat{H}) = \frac{1}{1+e^{-\hat{H}}}$ (where $\hat{H}$ is model learned from eq. (6)).

When learning the GRBB model, we estimate the true function $H$ by minimizing the negative log-likelihood functional $\ell(H)$ plus gradient regularization. Minimizing the negative log-likelihood is equivalent to maximum likelihood estimation (MLE). Moreover, this estimate is consistent and asymptotically normal. Let $\hat{H}$ denote this estimate of $H$. Note that $\hat{H}$ is a random variable, because it depends on the data used for estimation, while $H$ is not. The average prediction variance of over all labeled and unlabeled inputs can be expressed as

$$\frac{1}{n+m}\sum_{i=1}^{n+m}\text{Var}[\sigma(\hat{H}_i)] = \frac{1}{n+m}\sum_{i=1}^{n+m}\text{E}\Big[\big\{\sigma(\hat{H}_i) - \sigma(H_i^*)\big\}^2\Big], \tag{7}$$

where $H_i$ is the prediction of the $i$th input, $n+m$ is the total number of labeled and unlabeled inputs, and $H^*$ is the estimate of predicting function $H$ assuming we have an infinite amount of training inputs. For each input, the expectation is taken over the distribution of predicting functions $p(\hat{H})$. The equation holds because the expectation of this distribution $\text{E}[\hat{H}] = H^*$.

Because we do not know $H_i^*$ we must approximate it. We can do so by taking the first-order Taylor expansion of $\sigma(H_i^*)$, and expanding around $\hat{H}_i$, to obtain,

$$\sigma(H_i^*) \simeq \sigma(\hat{H}_i) + \frac{\partial\sigma(H_i)}{\partial H_i}\Big|_{\hat{H}_i}(H_i^* - \hat{H}_i). \tag{8}$$

Using this expression, we can rewrite the average variance in eq. (7) as,

$$\frac{1}{n+m}\sum_{i=1}^{n+m}\text{E}\Big[\big\{\sigma(\hat{H}_i) - \sigma(H_i^*)\big\}^2\Big] = \frac{1}{n+m}\sum_{i=1}^{n+m}\text{E}\Bigg[\bigg\{\frac{\partial\sigma(H_i)}{\partial H_i}\Big|_{\hat{H}_i}(H_i^* - \hat{H}_i)\bigg\}^2\Bigg] \tag{9}$$

Since $H_i^* = \text{E}[\hat{H}_i]$, it is the case that $\text{E}\Big[(H_i^* - \hat{H}_i)^2\Big] = \text{V}[\hat{H}]$. Note that $\frac{\partial\sigma(H_i)}{\partial H_i}\Big|_{\hat{H}_i}$ is constant with respect to the expectation so we can further simplify,

$$= \frac{1}{n+m}\sum_{i=1}^{n+m}\text{V}\Big[\frac{\partial\sigma(H_i)}{\partial H_i}\Big|_{\hat{H}_i}\hat{H}_i\Big] = \frac{1}{n+m}\sum_{i=1}^{n+m}\Big(\frac{\partial\sigma(H_i)}{\partial H_i}\Big|_{\hat{H}_i}\Big)^2\text{V}[\hat{H}_i]. \tag{10}$$

The only unknown part is the variance of the estimate $\hat{H}$. Because of the asymptotic normality of MLE, we have $\hat{H} \sim \text{N}(H^*, \text{V}[\hat{H}])$. We can compute $\text{V}[\hat{H}]$ in closed-form by noting that $\text{V}[\hat{H}] = \text{I}_n(\hat{H})^{-1}$, where $\text{I}_n(\hat{H})$ is the Fisher information [34]. Given that our objective is twice differentiable, the empirical Fisher information can be computed in closed-form:

$$I_n(\hat{H}) = \sum_{i=1}^{n} -\text{E}\Big[\frac{\partial^2\mathcal{L}(\mathbf{y}|\mathbf{x};\hat{H})}{\partial\hat{H}^2}\Big], \tag{11}$$

where $\mathcal{L}(\cdot)$ is our objective (loss $\ell(\cdot)$ + Laplacian regularization), eq. (2).

However, different from other distribution parameters, in our model, the distribution parameter $H$ is a function, and the derivative of the objective functional w.r.t. parameter function $H$ can only be evaluated at each input. Therefore, the second order derivative in eq. (11) is w.r.t. every single input,

$$\frac{\partial^2\mathcal{L}(y_i|\mathbf{x}_i;\hat{H})}{\partial\hat{H}^2} = \frac{\partial^2\mathcal{L}(y_i|\mathbf{x}_i;\hat{H})}{\partial\hat{H}_i\partial\hat{H}_j}. \tag{12}$$
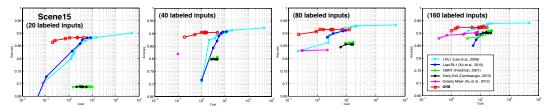
Figure 3: Cost and accuracy performance of various budgeted learning algorithms on the scene15 data set with different number of labeled inputs.
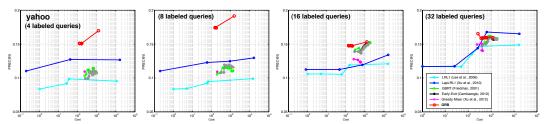


Figure 4: Cost and accuracy performance of various budgeted learning algorithms on the Yahoo data set with different number of labeled inputs.

Note that the log-likelihood functional plus manifold regularization objective $\mathcal{L}(\cdot)$ has two parts. The log-likelihood functional has a non-zero second order derivative only when the derivative is taken w.r.t. one *labeled* input prediction $H_i$ twice. The Laplacian regularization term has a clean second order derivative, which is the Laplacian matrix $\mathbf{L}$. If we define a diagonal matrix $\Delta_{jj} = \sigma(\hat{H}_j)^2(1 - \sigma(\hat{H}_j))$ we can combine both parts in closed form and express the Fisher information as

$$\mathbf{I}_n(\hat{H}) = \Delta + \lambda \mathbf{L}. \tag{13}$$

Applying the computed Fisher information to eq. (10), we derive a closed-form expression for computing the average variance of link predictions given labeled and unlabeled training inputs. Since at every iteration during gradient boosting, we generate trees to *approximate* the negative gradient, our estimate of $\hat{H}$ is less efficient than MLE, and the variance we compute is a lower bound.

## 6  Results

In this section, we evaluate GRBB on two real-world budgeted learning datasets with limited labeled training data. We compare its performance with several state-of-the-art budgeted learning algorithms.

### 6.1  Scene Recognition

We evaluate GRBB on the scene15 dataset [30]. The dataset contains 4485 images from 15 scenes (e.g., forest, coast, street, kitchen), and the task is to classify the scene in each image. We follow the procedure used in [35; 10], randomly sampling 100 images from each class, resulting in 1500 training images, and leaving the rest 2985 images for testing. We also follow their procedure to generate 2760 features with a variety of costs (for more detail, we refer readers to [35; 10]). Finally, we binarize the data set by grouping labels 1-8 as group 0, and 9-15 as group 1.

We randomly select $10, 20, 40, 80$ labeled inputs and leave the rest training inputs as unlabeled. All results are averaged over 5 runs. Hyper-parameters of GRBB (depth of trees, number of neighbors in Laplacian matrix) are set by cross-validation. To generate the cost/accuracy performance curve, we evaluate 11 different $\mu$ $(4^{-5}, 4^{-4}, 4^{-3}, \cdots, 0, 4^0, 4^1, \ldots, 4^4)$.

**Comparison.** The first baseline we compare against is logistic regression with weighted $l_1$ regularization [36] (*LRL1*), where the weight is the feature extraction cost. We generate the cost/accuracy curve by varying the regularization trade-off parameter. Since this simple baseline does not consider

any unlabeled inputs, we evaluate an extension. Similar to the first baseline, [37] uses $l_1$ regularization, but it also has an manifold regularization term to incorporate unlabeled information. We modify their algorithm and replace the $l_1$ regularization with weighted $l_1$ regularization, again weighting by the feature extraction cost. We refer to this algorithm as *LapLRL1*. To introduce nonlinearity, we evaluate regular *GBRT* [18] and two extensions *Early-exit* [22] and *Greedy Miser* [10]. We generate the performance curve of GBRT by evaluating every 10 trees. For Early-exit, we follow the procedure in [10], introducing an exit every 10 trees by removing test inputs whose prediction value is greater than a threshold after applying the link function $\sigma(\cdot)$. Finally, we vary the accuracy/cost trade-off parameter in Greedy Miser to generate its performance curve. For all these three algorithms, we use cross-validation to set the depth of trees, and set other hyper-parameters to the same value as GRBB (learning rate, total number of trees).

Figure 3 shows the performance of different budgeted learning algorithms described above. GRBB clearly out-performs all other algorithms when the cost is low. Specifically, when the number of labeled inputs is small $(10, 20, 40)$, linear algorithms (LRL1, LapLRL1) perform better as they are less likely to over-fit, and LapLRL1 has slightly better accuracy than LRL1 because it uses unlabeled data. Gradient boosted algorithms (GBRT, Early-exit and Greedy Miser) all suffer from over-fitting. On the other hand, even when the labeled inputs are limited, GRBB maintains a very high accuracy at a very low cost, indicating its gradient regularization significantly helps in reducing the test-time cost by selectively picking simultaneously inexpensive and predictive features that are otherwise difficult to discover from a small number of labeled inputs. When the number of labeled training inputs is large (80), nonlinear algorithms start to gain in accuracy, particularly Greedy Miser. However, even with increased labeled data, GRBB still out-performs Greedy miser as unlabeled data provide additional information.

## 6.2 Yahoo Learning to Rank

We also evaluate GRBB on Yahoo Learning to Rank data set [23]. The data set consists of query-document instance pairs. We follow [38; 12] to subsample and binarize the data set. The resulting labels are $\{-1, +1\}$, where $-1$ means the document is irrelevant to the query and $+1$ means relevant. The dataset is very imbalanced, with the majority of inputs being irrelevant. The total binarized data set contains $20258, 20258, 26256$ training, validation and testing documents. As each query corresponds to a number of documents we randomly sample $4, 8, 16, 32$ queries, along with their corresponding documents, as labeled inputs, leaving the remaining documents unlabeled. We average over 5 runs.

We perform a similar evaluation to that of the scene15 data set. There are three differences: First, we only train 200 trees of GRBB and other gradient boosted algorithms as the Yahoo data set is much larger. Second, to be fairer to the Early-exit baseline, because the dataset is imbalanced, we convert it to exit umpromising lower-ranked documents rather than more confident ones. Third, we use Precision@5 (PREC@5) as the evaluation metric, which counts the number of relevant documents in the top five retrieved documents for each query. Figure 4 shows the performance of the same algorithms we evaluate on scene15, on the Yahoo
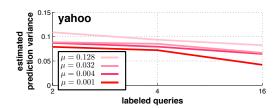


Figure 5: Prediction variance on the Yahoo! LTR dataset with different number of labeled training inputs and different allowed budgets.

data set. A similar trend can be observed, as we increase the number of labeled queries, gradient boosted algorithms generally out-perform linear algorithms at lower costs. GRBB has a clear advantage over other algorithms for fewer labeled inputs. This again is due to its unique capacity to use unlabeled inputs to prevent over-fitting.

We use the method described in section 5 to compute a lower bound on the prediction variance for different numbers of labeled queries $(2, 4, 16)$ and different values of the feature-cost trade-off parameter $\mu$. We average over 10 runs, using ensembles with $1,000$ trees. We note that as $\mu$ decreases, more features may be extracted (as effectively the budget is larger) and the prediction variance bound decreases. As we would expect, as more queries are trained on the prediction variance bound also

decreases. In practice, with a known feature cost budget (corresponding to a specific value of $\mu$) we can use these curves to decide if more labeled inputs should be requested (*e.g.*, from an expert).

## 7    Related Work

Using boosting in budgeted learning has proved very successful [4; 39; 6; 8; 10]. Viola and Jones [4] were perhaps the first to recognize that boosting was well suited for selecting features. Reyzin [39] developed modifications of Adaboost for selecting features with costs. Busa et al. [6] use a Markov decision process to adaptively select boosted learners. Grubb and Bagnell [8] amd Xu et al. [10] greedily construct cost-sensitive boosted trees. None of the prior work in budgeted learning, to our knowledge, considers learning with a small amount of labeled data and a large amounts of unlabeled data.

At the same time boosting has been used for semi-supervised learning [40; 41; 42; 43]. Saffari et al. [41] use expectation regularization on the margin of the boosting loss in AdaBoost to incorporate unlabeled inputs. Kumar et al. [43] iteratively sample unlabeled data for training by assigning confidences to nearby inputs. These inputs, alongside the training data, are used to train an AdaBoost-inspired ensemble. Grabner et al. [42] modify AdaBoost for online semi-supervised boosting. Most similar to our method, Chen et al. [40] add a local smoothness regularization term to AdaBoost. Different from all prior methods, our proposed gradient regularization explicitly learns a compact semi-supervised model with gradient boosting that has small evaluation cost.

## 8    Discussion

To our knowledge, we propose the first algorithm for budgeted learning that leverages unlabeled data. We learn cheap, compact ensembles via a novel idea that regularizes the gradients of the unlabeled inputs. We show how one can get an informative lower bound on the prediction variance that can tell practitioners if they should invest in labeling more inputs, whatever their application. Our empirical results are highly encouraging as GRBB successfully inherits the merits of cost-sensitive feature selection from gradient boosting, even when the labeled training set is small, while simultaneously shrinking the evaluation cost.

## References

[1]  Z. Zheng, H. Zha, T. Zhang, O. Chapelle, K. Chen, and G. Sun. A general boosting method and its application to learning ranking functions for web search. In *NIPS*, pages 1697–1704. Cambridge, MA, 2008.

[2]  Q. Ye, L. Qin, M. Forgues, P. He, J.W. Kim, A.C. Peng, R. Simon, Y. Li, A.I. Robles, Y. Chen, et al. Predicting hepatitis b virus–positive metastatic hepatocellular carcinomas using gene expression profiling and supervised machine learning. *Nature medicine*, 9(4):416–423, 2003.

[3]  L. Bottou, J. Peters, J. Quiñonero-Candela, D.X. Charles, D.M. Chickering, E. Portugaly, D. Ray, P. Simard, and D. Snelson. Counterfactual reasoning and learning systems: the example of computational advertising. *The Journal of Machine Learning Research*, 14(1):3207–3260, 2013.

[4]  P. Viola and M.J. Jones. Robust real-time face detection. *IJCV*, 57(2):137–154, 2004.

[5]  T. Gao and D. Koller. Active classification based on value of classifier. In *NIPS*, pages 1062–1070. 2011.

[6]  R. Busa-Fekete, D. Benbouzid, B. Kégl, et al. Fast classification using sparse decision dags. In *ICML*, 2012.

[7]  H. He, H. Daumé III, and J. Eisner. Cost-sensitive dynamic feature selection. In *ICML Workshop on Inferning: Interactions between Inference and Learning, Edinburgh*, 2012.

[8]  A. Grubb and J. A. Bagnell. Speedboost: Anytime prediction with uniform near-optimality. In *AISTATS*, 2012.

[9]  S. Karayev, T. Baumgartner, M. Fritz, and T. Darrell. Timely object recognition. In *Advances in Neural Information Processing Systems 25*, pages 899–907, 2012.

[10]  Z. Xu, K. Weinberger, and O. Chapelle. The greedy miser: Learning under test-time budgets. In *ICML*, pages 1175–1182, 2012.

[11] Kirill Trapeznikov, Venkatesh Saligrama, and David Castañón. Multi-stage classifier design. *Machine learning*, 92(2-3):479–502, 2013.

[12] Z. Xu, M.J. Kusner, G. Huang, and K.Q. Weinberger. Anytime representation learning. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 1076–1084, 2013.

[13] Joseph Wang, Kirill Trapeznikov, and Venkatesh Saligrama. An lp for sequential learning under budgets. In *AISTATS*, pages 987–995, 2014.

[14] S. Karayev, M. Fritz, and T. Darrell. Anytime recognition of objects and scenes. In *CVPR*, 2014.

[15] Luming Wang and Per-Olof Persson. A high-order discontinuous galerkin method with unstructured space–time meshes for two-dimensional compressible flows on domains with large deformations. *Computers & Fluids*, 118:53–68, 2015.

[16] Sam Kanner, Luming Wang, and Per-Olof Persson. Implicit large-eddy simulation of 2d counter-rotating vertical-axis wind turbines. In *34th Wind Energy Symposium*, page 1731, 2016.

[17] Luming Wang. *Discontinuous Galerkin Methods on Moving Domains with Large Deformations*. PhD thesis, UC Berkeley, 2015.

[18] J.H. Friedman. Greedy function approximation: a gradient boosting machine. *The Annals of Statistics*, pages 1189–1232, 2001.

[19] M. Chen, K.Q. Weinberger, O. Chapelle, D. Kedem, and Z. Xu. Classifier cascade for minimizing feature evaluation cost. In *AISTATS*, pages 218–226, 2012.

[20] Jeffrey Lee Hellrung Jr, Luming Wang, Eftychios Sifakis, and Joseph M Teran. A second order virtual node method for elliptic problems with interfaces and irregular domains in three dimensions. *Journal of Computational Physics*, 231(4):2015–2048, 2012.

[21] Luming Wang and Per-Olof Persson. A discontinuous galerkin method for the navier-stokes equations on deforming domains using unstructured moving space-time meshes. In *21st AIAA Computational Fluid Dynamics Conference*, page 2833, 2013.

[22] B Barla Cambazoglu, Hugo Zaragoza, Olivier Chapelle, Jiang Chen, Ciya Liao, Zhaohui Zheng, and Jon Degenhardt. Early exit optimizations for additive machine learned ranking systems. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 411–420. ACM, 2010.

[23] O. Chapelle and Y. Chang. Yahoo! learning to rank challenge overview. In *JMLR: Workshop and Conference Proceedings*, volume 14, pages 1–24, 2011.

[24] O. Chapelle, P. Shivaswamy, S. Vadrevu, K. Weinberger, Y. Zhang, and B. Tseng. Boosted multi-task learning. *Machine learning*, 85(1):149–173, 2011.

[25] Xiaojin Zhu. Semi-supervised learning literature survey. *Computer Science, University of Wisconsin-Madison*, 2:3, 2006.

[26] Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *The Journal of Machine Learning Research*, 7:2399–2434, 2006.

[27] Kazuki Yoshiyama and Akito Sakurai. Manifold-regularized minimax probability machine. In *Partially Supervised Learning*, pages 42–51. Springer, 2012.

[28] Xiaojin Zhu and Zoubin Ghahramani. Learning from labeled and unlabeled data with label propagation. Technical report, CMU-CALD-02-107, Carnegie Mellon University, 2002.

[29] Z. Xu, M. J. Kusner, K. Q. Weinberger, and M. Chen. Cost-sensitive tree of classifiers. In *ICML*, 2013.

[30] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, pages 2169–2178, 2006.

[31] L. Breiman. *Classification and regression trees*. Chapman & Hall/CRC, 1984.

[32] Fan RK Chung. *Spectral graph theory*, volume 92. AMS Bookstore, 1997.

[33] J. Snoek, H. Larochelle, and R. P. Adams. Practical bayesian optimization of machine learning algorithms. In *NIPS 2012*, pages 2960–2968. 2012.

[34] Lucien Le Cam. Asymptotic methods in statistical decision theory. *New York*, 1986.

[35] L.J. Li, H. Su, E.P. Xing, and L. Fei-Fei. Object bank: A high-level image representation for scene classification and semantic feature sparsification. *NIPS*, 2010.

[36] S. Lee, H. Lee, P. Abbeel, and A. Y. Ng. Efficient l1 regularized logistic regression. In *Proceedings of the National Conference on Artificial Intelligence*, volume 21, page 401. Menlo Park, CA; Cambridge, MA;

London; AAAI Press; MIT Press; 1999, 2006.

[37] Zenglin Xu, Irwin King, MR-T Lyu, and Rong Jin. Discriminative semi-supervised feature selection via manifold regularization. *Neural Networks, IEEE Transactions on*, 21(7):1033–1047, 2010.

[38] M. Chen, Z. Xu, K. Q. Weinberger, and O. Chapelle. Classifier cascade for minimizing feature evaluation cost. In *AISTATS*, 2012.

[39] L. Reyzin. Boosting on a budget: Sampling for feature-efficient prediction. In *ICML*, pages 529–536, 2011.

[40] K. Chen and S. Wang. Regularized boost for semi-supervised learning. In *NIPS*, pages 281–288, 2007.

[41] Amir Saffari, Helmut Grabner, and Horst Bischof. Serboost: Semi-supervised boosting with expectation regularization. In *Computer Vision–ECCV 2008*, pages 588–601. Springer, 2008.

[42] Helmut Grabner, Christian Leistner, and Horst Bischof. Semi-supervised on-line boosting for robust tracking. In *Computer Vision–ECCV 2008*, pages 234–247. Springer, 2008.

[43] P Kumar Mallapragada, Rong Jin, Anil K Jain, and Yi Liu. Semiboost: Boosting for semi-supervised learning. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(11):2000–2014, 2009.