Exercises in Machine Learning

Srikanth Pai, Assistant Professor, MSE, Chennai

March 18, 2026

1 Gradient Descent for Quadratic Optimization

1. Let $a, b > 0$ with $a > b$. Prove that

$$\min_{\alpha>0} \max\{|1-\alpha a|, |1-\alpha b|\} = \frac{a-b}{a+b} \tag{1}$$

achieved at $\alpha^* = \dfrac{2}{a+b}$.

2. Let $\alpha > 0$ be fixed. Prove that $f(x) = |1-\alpha x|$ is a convex function of x, and that for any finite set $\{x_1, \ldots, x_n\}$, the maximum $\max_i f(x_i)$ is attained at an extreme point.

3. **Setup:** Consider the quadratic optimization problem

$$f(w) = \frac{1}{2} w^T H w - b^T w + c \tag{2}$$

with $H \in \mathbb{R}^{d\times d}$ symmetric positive definite. The gradient descent update is

$$w^{(t+1)} = w^{(t)} - \alpha \nabla f(w^{(t)}) \tag{3}$$

Let $w^* = H^{-1}b$ and $e^{(t)} = w^{(t)} - w^*$. Prove that $e^{(t+1)} = (I - \alpha H)e^{(t)}$. For eigendecomposition $H = U\Lambda U^T$ with $\lambda_1 \geq \cdots \geq \lambda_d > 0$, define $z^{(t)} = U^T e^{(t)}$ and prove that $z_i^{(t+1)} = (1-\alpha\lambda_i) z_i^{(t)}$ for all i. Conclude that

$$\|e^{(t)}\|_2 \leq \rho^t \|e^{(0)}\|_2 \quad \text{where} \quad \rho = \max_i |1-\alpha\lambda_i|. \tag{4}$$

4. Using items 1 and 2, prove that the optimal step size $\alpha^* = \frac{2}{\lambda_1+\lambda_d}$ yields the convergence rate

$$\|e^{(k)}\|_2 \leq \left(\frac{\kappa-1}{\kappa+1}\right)^k \|e^{(0)}\|_2 \tag{5}$$

where $\kappa = \frac{\lambda_1}{\lambda_d}$ is the condition number of H.

5. For least squares with $H = \frac{1}{n}X^TX$ where $X \in \mathbb{R}^{n\times d}$, prove that achieving $\|e^{(N)}\|_2 \leq \epsilon\|e^{(0)}\|_2$ requires

$$N \geq \frac{\kappa+1}{2}\log\left(\frac{1}{\epsilon}\right) \tag{6}$$

iterations. Since each iteration costs $O(nd)$ flops, show that the total runtime is

$$T_{\text{GD}} = O\left(nd\cdot\kappa\log\left(\frac{1}{\epsilon}\right)\right). \tag{7}$$

Prove that gradient descent is faster than solving the normal equations via matrix inversion at cost $O(d^3)$ when $\kappa\log(1/\epsilon) \ll d^2$.

6. Argue that for sparse matrices with nnz nonzeros stored in CSR format, the gradient descent runtime cost is $O(\text{nnz}\cdot\kappa\log(1/\epsilon))$, since each iteration now costs $O(\text{nnz})$ instead of $O(nd)$.

7. Implement the gradient descent algorithm for least squares in Python. Generate synthetic data with $n = 1000$ samples and $d = 100$ features, and compare the convergence of gradient descent with the closed-form solution obtained via matrix inversion. Plot the convergence of $\|e^{(t)}\|_2$ over iterations. Also use CSR (Compressed Sparse Row) format to store the data matrix and compare the runtime with the dense implementation.

8. **Application: 20 Newsgroups Sparse Text Dataset.** The 20 Newsgroups dataset (available at `http://qwone.com/~jason/20Newsgroups/`) contains $n = 18,846$ documents and $d = 130,107$ TF-IDF features. The feature matrix $X \in \mathbb{R}^{n\times d}$ is 99.8 percent sparse. Form the quadratic problem $\min_w \frac{1}{2n}\|Xw - y\|_2^2$ with binary target $y \in \{-1, +1\}$ for category prediction. Compute $H = \frac{1}{n}X^TX$ and $b = \frac{1}{n}X^Ty$ using sparse matrix operations. Run gradient descent for $N = \lceil\frac{\kappa+1}{2}\log(10^6)\rceil$ iterations with optimal step size. Record the wall-clock time. Estimate the storage and runtime required for explicit matrix inversion and verify that sparse gradient descent is orders of magnitude faster.

2 Decision Trees and Random Forests

The problem of learning from data admits two opposing demands: individual predictors must be accurate (strength), yet the ensemble must be diverse (low correlation). This tension, first articulated rigorously by Breiman (2001), is the foundation of modern random forests. We develop this theory from first principles.

2.1 Information-Theoretic Foundations of Splitting

1. Let $p \in [0, 1]$. Define entropy as

$$H(p) = -p\log_2(p) - (1-p)\log_2(1-p) \tag{8}$$

with $0\log(0) = 0$. Prove that H is strictly concave on $[0, 1]$, maximized at $p = 0.5$ with $H(0.5) = 1$, and $H(0) = H(1) = 0$.

2. A dataset S has class proportions $p_0 = \frac{\#\text{label } 0}{|S|}$ and $p_1 = 1 - p_0$. A split creates disjoint S_L, S_R with $S_L \cup S_R = S$, satisfying

$$p_0 = \frac{|S_L|}{|S|} p_{0,L} + \frac{|S_R|}{|S|} p_{0,R}. \tag{9}$$

Define information gain as

$$\text{IG} = H(p_0) - \frac{|S_L|}{|S|} H(p_{0,L}) - \frac{|S_R|}{|S|} H(p_{0,R}). \tag{10}$$

Prove that $\text{IG} \geq 0$ for all splits using the concavity of H from item 1.

3. A tree of depth d has at most 2^d leaves. Prove this by induction on d.

4. Implement `best_split(X, y)` where $X \in \mathbb{R}^{n \times d}$, $y \in \{0, 1\}^n$. For each feature j and threshold t, partition into S_L, S_R, compute information gain, and return the feature, threshold, and maximum gain.

5. Implement `grow_tree(X, y, max_depth)` that recursively calls `best_split`, creates children, stores split feature and threshold. Leaves predict majority class. Verify that depths 1, 3, 10 have increasing leaves and decreasing training error.

2.2 Variance Reduction via Ensemble Averaging

Trees are weak learners; their power comes from aggregation. Let $T_b(x) \in \mathbb{R}$ denote the prediction of tree b at input x. The random forest prediction is the average $\text{RF}(x) = \frac{1}{B} \sum_{b=1}^{B} T_b(x)$. The following lemmas establish how correlation and ensemble size interact to reduce prediction variance.

6. Let $Y_1, \ldots, Y_B$ be independent random variables with mean μ and variance σ^2. Define

$$\bar{Y} = \frac{1}{B} \sum_{b=1}^{B} Y_b. \tag{11}$$

Prove that $\mathbb{E}[\bar{Y}] = \mu$ and $\text{Var}[\bar{Y}] = \frac{\sigma^2}{B}$.

7. Let $Y_1, \ldots, Y_B$ have mean μ, variance σ^2, and pairwise correlation $\text{Corr}(Y_b, Y_{b'}) = \rho$ for all $b \neq b'$. Define

$$\bar{Y} = \frac{1}{B} \sum_{b=1}^{B} Y_b. \tag{12}$$

Prove that

$$\text{Var}[\bar{Y}] = \frac{\sigma^2}{B} + \frac{B-1}{B} \rho \sigma^2. \tag{13}$$

8. For a random forest, assume each tree b produces a prediction $T_b(x)$ with common variance $\text{Var}[T_b(x)] = \sigma^2$ and pairwise correlation $\text{Corr}(T_b(x), T_{b'}(x)) = \rho$ for all $b \neq b'$. The random forest prediction is $\text{RF}(x) = \frac{1}{B} \sum_{b=1}^{B} T_b(x)$.

(a) Prove that

$$\mathrm{Var}[\mathrm{RF}(x)] = \frac{\sigma^2}{B} + \frac{B-1}{B}\rho\sigma^2. \tag{14}$$

(b) Prove that for fixed B, the variance $\mathrm{Var}[\mathrm{RF}(x)]$ decreases strictly as ρ decreases from 1 to 0.

(c) Let $\mathcal{F}_b, \mathcal{F}_{b'} \subset \{1, \ldots, d\}$ be the set of features used by trees b and b', each of size m, drawn independently uniformly at random. Prove that

$$\mathbb{E}[|\mathcal{F}_b \cap \mathcal{F}_{b'}|] = \frac{m^2}{d}. \tag{15}$$

Assume the tree prediction decomposes as

$$T_b(x) = \sum_{j \in \mathcal{F}_b} g_j(x_j) \tag{16}$$

where $g_j(x_j)$ are deterministic functions and the random variables $g_j(x_j)$ are independent with common variance τ^2. Prove that

$$\mathrm{Cov}(T_b(x), T_{b'}(x)) = \tau^2 |\mathcal{F}_b \cap \mathcal{F}_{b'}|, \tag{17}$$

hence

$$\mathbb{E}[\mathrm{Cov}(T_b(x), T_{b'}(x))] = \tau^2 \frac{m^2}{d}. \tag{18}$$

Conclude that expected correlation decreases as m decreases.

9. Let $T_b(x)$ denote the prediction of tree b at input x. For a fully grown tree (leaf size 1), $T_b(x)$ equals a single training label, hence

$$\mathrm{Var}(T_b(x)) = \sigma_\varepsilon^2.$$

Assume $\mathrm{Corr}(T_b(x), T_{b'}(x)) = \rho < 1$ for $b \neq b'$. Defining

$$\mathrm{RF}(x) = \frac{1}{B} \sum_{b=1}^{B} T_b(x),$$

an application of the formula for variance earlier, yields

$$\mathrm{Var}(\mathrm{RF}(x)) = \frac{\sigma_\varepsilon^2}{B} + \frac{B-1}{B}\rho\sigma_\varepsilon^2 < \sigma_\varepsilon^2.$$

10. Implement `random_forest_train(X, y, B, max_depth)` that trains B trees on bootstrap samples. Implement `random_forest_predict(trees, X)` that computes $\mathrm{RF}(x) = \frac{1}{B}\sum_{b=1}^{B} T_b(x)$. Measure empirical variance $\widehat{\mathrm{Var}}[\mathrm{RF}]$ for $B \in \{1, 5, 10, 50\}$ on held-out test set and verify it is approximately proportional to $\frac{1}{B}$.

2.3 Bootstrap Sampling and Out-of-Bag Estimation

Breiman's key insight was that bootstrap resampling provides a "free" validation set. Every sample left out of a tree's training set can be used to estimate generalization error, strength, and correlation without sacrificing training data.

Let $D = \{(x_i, y_i)\}_{i=1}^n$ be a fixed training dataset with $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$. For each tree $b \in \{1, \ldots, B\}$, a bootstrap sample S_b is drawn by sampling n indices uniformly at random with replacement from $\{1, \ldots, n\}$. The tree T_b is trained on the subset $\{(x_i, y_i) : i \in S_b\}$ of the dataset. Let $T_b : \mathcal{X} \to \mathcal{Y}$ denote the resulting tree predictor.

For each sample $i \in \{1, \ldots, n\}$, define the out-of-bag index set

$$B_i := \{b \in \{1, \ldots, B\} : i \notin S_b\}.$$

The OOB prediction for sample i is

$$\hat{y}_i^{\text{OOB}} := \frac{1}{|B_i|} \sum_{b \in B_i} T_b(x_i).$$

The OOB error is

$$\text{Err}_{\text{OOB}} := \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}[\hat{y}_i^{\text{OOB}} \neq y_i].$$

Assume the training data D is drawn i.i.d. from an unknown distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$. The test error is

$$\text{Err}_{\text{test}} := \mathbb{E}_{(X,Y)\sim\mathcal{D}}[\mathbf{1}[T(X) \neq Y]]$$

where T is a tree trained on a random sample from $\mathcal{D}$.

12. Prove that for a fixed sample i, the probability that i is not in bootstrap sample S_b is

$$\mathbb{P}(i \notin S_b) = \left(1 - \frac{1}{n}\right)^n. \tag{19}$$

    Show that $(1 - 1/n)^n \to e^{-1}$ as $n \to \infty$.

13. For each tree b, define the indicator $Z_{b,i} := \mathbf{1}[i \notin S_b]$. The variables $\{Z_{b,i}\}_{b=1}^B$ are i.i.d. Bernoulli with parameter $p_n := (1 - 1/n)^n$. Since $|B_i| = \sum_{b=1}^B Z_{b,i}$, prove by the law of large numbers that

$$\frac{|B_i|}{B} \xrightarrow{p} p_n \quad \text{as } B \to \infty. \tag{20}$$

    Conclude that $\mathbb{E}[|B_i|]/B \to e^{-1}$ as $n \to \infty$.

14. Since $i \notin S_b$, the tree T_b is trained on data excluding sample i. Therefore, $T_b(x_i)$ is independent of y_i conditional on $Z_{b,i} = 1$. For all $b \in B_i$, prove that

$$\mathbb{E}[\mathbf{1}[\hat{y}_i^{\text{OOB}} \neq y_i]] = \mathbb{E}[\mathbf{1}[T(X) \neq Y]] \tag{21}$$

    where $(X, Y) \sim \mathcal{D}$ is an independent test sample. Averaging over all i, prove that

$$\mathbb{E}[\text{Err}_{\text{OOB}}] = \text{Err}_{\text{test}}. \tag{22}$$

15. Implement `random_forest_oob_error(X, y, B, max_depth)` that computes the OOB error. Train a random forest on Iris with $B = 50$ trees and depth 5 using 120 training samples. Compute the OOB error and compare it to the error on a held-out test set of 30 samples. Verify that they are approximately equal.

2.4 Bias-Variance Decomposition and Depth Selection

This section analyzes the bias-variance decomposition of tree ensembles as a function of depth. The analysis illustrates why direct method comparison requires understanding the error structure of each approach.

1. A logistic regression model defines decision boundary $\{x : w^T x = 0\}$. A tree partitions into axis-aligned rectangles. Prove that the region

$$R = \{x : X_1 > t_1 \text{ and } X_2 > t_2\} \tag{23}$$

   cannot be the zero set of any linear function $w^T x$. Conclude that trees represent feature interactions that linear models cannot without explicit feature engineering.

2. Implement logistic regression with gradient descent on Iris (120 train, 30 test). Report test error. Train single tree (depth 5) and RF ($B = 10$, depth 5). Create table with all three test errors. Verify tree methods outperform logistic regression.

3. A RF with B trees of depth d costs $O(Bnd \log n)$ flops. Logistic regression costs $O(ndk)$ where k is iterations. For Iris with $n = 120, d = 4$, compute $d \log n \approx 16$ and measure k (typically ≥ 100). Implement both and compare wall-clock training time.

4. Implement `optimal_depth(X, y, B, depths)` that trains RF at each depth, computes OOB error, returns depth minimizing OOB error. Apply to Iris with $B = 10$ and depths $\{1, 2, 3, 4, 5, 7, 10\}$. Plot OOB error versus depth.