This detailed notes consolidate the content covered across the provided source excerpts, focusing on optimization fundamentals, convexity, optimality conditions, and numerical algorithms.

## Detailed Notes on Optimization Topics

### I. Optimization Fundamentals and Conditions

#### A. Problem Setup and Solution Concepts

Optimization seeks to **minimize subject to** , where is the objective (loss) function and is the constraint set. \* **Decision Variables** (parameters) are the variables that are adjusted to minimize . \* A point is a **global minimizer** if for all . \* A point is a **local minimizer** if for all in a neighborhood . \* Optimization is crucial to Data Science, being the backbone alongside Linear Algebra, Calculus, and Statistics. Many Machine Learning problems, such as Model fitting (e.g., Linear Regression), involve optimization.

#### B. Optimality Conditions (Necessary and Sufficient)

These conditions characterize solutions, assuming differentiability at an interior point .

1. **First-Order Necessary Condition (Fermat’s Rule):**
   * If is a local minimizer or maximizer, the **gradient must vanish**: .
   * In 1D, this means . Note that is *not* sufficient (e.g., inflection points).
2. **Second-Order Sufficiency Conditions:** Used at a stationary point where . The classification depends on the **Hessian matrix** .
   * (Positive Definite) **strict local minimum**.
   * (Negative Definite) **strict local maximum**.
   * indefinite **saddle point**.
3. **Positive Definite and Semidefinite Matrices:**
   * A matrix is positive definite (PD) or positive semidefinite (PSD) if and only if all its **eigenvalues are positive or nonnegative**, respectively.
   * If a matrix is PSD (PD), its trace and determinant are nonnegative (positive).
   * A matrix is PD if and only if all its **principal minors are positive**.

### II. Convexity

#### A. Definitions and Characterization

Convexity is essential because it guarantees that **every local minimum is also a global minimum**, and first-order conditions become sufficient.

* **Convex Set:** A set is convex if, for all and , the line segment connecting them is contained in : .
* **Convex Function:** A function is convex if, for all and , the function value along the line segment is below the value of the chord: .
  + *Examples:* , , and .
* **First-Order Characterization:** A differentiable function is convex **iff** for all .
  + **Sufficient Condition:** For a differentiable convex function, is a necessary and **sufficient** condition for to be a global minimizer.
* **Second-Order Condition:** A twice differentiable function is convex **iff** its **Hessian matrix is positive semidefinite** everywhere: .

#### B. Quadratic Functions

A quadratic function is represented as . \* **Convexity:** A quadratic function is convex if and only if the defining symmetric matrix is positive semidefinite. \* **Minimizers:** If is positive definite, the unique global minimizer is .

#### C. Convexity in Machine Learning

Many core ML problems are convex, allowing them to be solved efficiently: \* **Least Squares Regression:** Has a convex quadratic objective, . \* **Logistic Regression:** Uses a convex log-loss function. \* **Support Vector Machines (SVMs):** Uses the convex hinge loss.

### III. Numerical Optimization Algorithms

Many optimization problems lack a closed-form solution, necessitating iterative numerical methods to generate a sequence converging to a critical point where .

#### A. General Line Search Method

Iterative methods generally follow a line search scheme: 1. Choose a descent direction (where ). 2. Choose a step-size . 3. Update .

#### B. Gradient Descent (Steepest Descent)

This method uses the direction of steepest descent: . \* **Update Rule:** . \* **Issues:** Batch GD requires the entire dataset (slow for large data) and is sensitive to the choice of learning rate .

#### C. Step Length Selection

Choosing an appropriate step length is critical. \* **Exact Line Search** minimizes with respect to (often too expensive). \* **Inexact Line Search** uses conditions like **Wolfe’s conditions**: 1. **Armijo (Sufficient Decrease):** . 2. **Curvature Condition:** . \* **Lipschitz Gradient (L-smoothness):** If , choosing a step size guarantees a descent step and facilitates convergence analysis. \* **Backtracking Line Search:** A practical implementation of inexact line search that iteratively reduces until the Armijo condition is met.

#### D. Gradient Descent Variants (for Scalability)

These methods are widely used for large datasets, especially in deep learning. \* **Stochastic Gradient Descent (SGD):** Updates parameters based on the gradient of a single randomly sampled data point per iteration, resulting in faster but noisier updates. \* **Mini-Batch Gradient Descent:** Uses a small subset (e.g., 32, 64) of data points per iteration, providing a smoother update path than SGD. \* **Adam Optimizer:** A widely used default that **combines Momentum and RMSProp** by maintaining moving averages of both the gradients and the squared gradients.

#### E. Newton’s Method

Newton’s method uses the second derivative information (Hessian) to compute the descent direction. \* **Direction :** . \* **Pure Newton’s Method:** Uses . For quadratic functions with a positive definite Hessian, it reaches the unique minimizer in **just one step**. \* **Damped Newton’s Method:** Incorporates a line search step size to ensure descent .

#### F. Quasi-Newton Methods

These methods approximate the Newton step without the high computational cost of calculating and inverting the Hessian. They replace the Hessian with a positive definite approximation . \* **Search Direction:** , where (approximation of the inverse Hessian). \* **Secant Equation:** The approximation must satisfy , or . \* **BFGS (Broyden-Fletcher-Goldfarb-Shanno):** The most popular quasi-Newton method, providing a rank two update formula for . \* **L-BFGS (Limited-memory BFGS):** A modification that stores only a few vectors () instead of the full inverse Hessian approximation, making it scalable for high-dimensional problems.

#### G. Conjugate Gradient (CG) Method

CG is an algorithm designed primarily for minimizing convex quadratic functions where is positive definite. \* **Principle:** It generates a set of search directions that are **conjugate** with respect to , meaning for . \* **Advantage:** For quadratic functions, CG finds the exact solution in at most steps. \* **Fletcher-Reeves CG:** An iteration scheme that dynamically determines the conjugate directions using the current and previous gradients () and a scaling factor . The update direction is .