Learning Objectives

At the end of this module, students will be able to:
- Design classifiers using training data and labels
- Write systems of linear equations in matrix-vector form
- Apply the definition of linear independence
- Apply the concept of rank
- Determine when a system of linear equations has no solution, a unique solution, or a nonunique solution
- Determine whether a function is a valid norm
- Find the least-squares solution to a system of linear equations
- Compute gradients of linear and quadratic functions of a vector
- Apply the definition of positive definite and positive semidefinite matrices
- Apply the orthogonality principle to solve a least-squares problem
- Project vectors onto a subspace spanned by a set of linearly independent vectors
- Design classifiers using orthonormal bases
- Find orthonormal bases using the Gram-Schmidt procedure
- Solve the Tikhonov-regularized least-squares problem
- Apply cross-validation to assess classifier performance

Significance of Unit

Supervised learning is a problem of training our machine learning algorithm using data where we know the right answers. We present a whole bunch of data to our algorithm, so it can learn the patterns in the data because it knows, for example, when it looks at a picture, it knows the identity of that person. Once it learns those patterns, we can apply new data, that the algorithm hasn't yet seen, and it can make a decision, a prediction, or a classification. Supervised learning assumes you have data where you know the truth. When you train a classifier or a model using supervised learning, you end up getting a system of linear equations that needs to be solved. Much of the linear algebra that we'll look at, involves solving systems of linear equations, which will introduce ideas such as rank, subspace, linear independence, bases, positive definite property for matrices and so on. Now, when you have a system of linear equations, there are three possibilities that you might image: 1–there's no solution to that system of linear equations 2–there's a single, exact solution 3–there are an infinite number of solutions. The cases that we're interested in are 1 and 3. The middle case, where there's a single solution, rarely happens. Often times, we're asking our algorithm to do something very difficult in which there may be no exact solution, so we're going to try to find a solution that's good. For example, minimizes the squared error is a criterion that we're going to use. So, we're going to look at finding solutions that are approximate, that give us a good result. The other case, where we have an infinite number of solutions, occurs often and this is a problem because some of those solutions end up giving us very poor performance when we generalize to new data, so we want to pick (of the many possible solutions) a good one. So, we will introduce criteria that will bias us in that direction, using a technique called regularization. One of the things we're going to do is visualize the geometry of these equations and concepts. Linear algebra can be super geometrical, and there is a lot of powerful insight you can get by looking at the geometry, so do your best at drawing these pictures. Finally, we'll look at something for evaluating performance, which his called cross validation, in which you use some of your data to train the algorithm, and then use the rest of the data to test the performance, which allows us to get a good prediction of how our results will generalize to new data.

Key Topics

1. Write classifier learning problem as a system of linear equations in matrix-vector form
2. Exact solutions of systems of linear equations

  2.1. Linear independence of vectors
  2.2. Rank of a matrix
    2.2.1. Outer product representation for low-rank matrices
  2.3. Conditions for no solution, a unique solution, or non-unique solutions

3. Approximate solutions of linear equations

  3.1. Norms
  3.2. Least squares formulation
  3.3. Positive definite and semi-positive definite matrices
  3.4. Gradients of linear and quadratic functions of a vector
  3.5. Solving least-squares problems using gradients
  3.6. Solving least-squares problems using the orthogonality principle

4. Subspaces and bases

  4.1. Orthogonality and orthonormality
  4.2. Classification in orthobases
  4.3. Projecting vectors onto a subspace

5. Cross validation for classification performance evaluation
6. Tikhonov regularization

  6.1. Rationale
  6.2. Solution
  6.3. Cross validation for choosing the regularization parameter

Learning Activities

- Instructional Units 2.1, 2.2
- Activity 4
- Instructional Units 2.3-2.5
- Activity 5
- Assignment 2
- Instructional Units 2.6-2.8
- Activity 6
- Instructional Unit 2.9
- Activity 7
- Assignment 3
- Instructional Unit 2.10
- Activity 8
- Instructional Unit 2.11
- Activity 9
- Assignment 4
- Unit 2 Overview Quiz

Recommended Reading

- LE 2.3 Inner Product and Vector Norms
- LE 2.5 Linear Independence: Bases
- LE 2.6 The Rank of a Matrix
- LE 3.2 Symmetric Positive Definite Matrices
- LE 3.6 The Least Squares Problem
- LE 4.1 Orthogonal Vectors and Matrices