Machine Learning & Voronoi Diagrams

Laila Albalkhi
Supervised by Dr. Luis Rueda

University of Windsor

Oct. 22 2023

Types of Machine Learning

- Supervised learning (predictive)
  - Given inputs and right answers
  - Mapping from inputs x to outputs y given a labeled set $D = \{(x_i, y_i)\}_{i=1}^N$
  - Classification, regression, structured prediction
- Unsupervised learning (descriptive)
  - Given only unlabeled set $D = \{x_i\}_{i=1}^N$
  - Density estimation, clustering
- Semi-supervised learning
  - Given both a labeled and unlabeled training
- Reinforcement learning
  - Learn from reward or punishment functions

Basic Concepts

- Parametric vs non-parametric models
- K-nearest neighbour classifier (KNN)
- Curse of dimensionality
- Linear regression
- Overfitting
- Model selection
  - Misclassification rate
  - Generalization error
  - K-fold cross-validation
  - No free lunch theorem

Models

- Training, Inference, Decision
- Discriminative Functions
  - Find a discriminative function $f(x)$ that maps each input x to a class label $\hat{y}$
  - Probabilities do not matter
  - E.g.: k-Nearest Neighbour
- Discriminative Models
  - Model the posterior class probabilities $p(y|x)$ directly
  - Inputs are not modeled
  - E.g.: Logistic regression, k-Nearest Neighbour
- Generative Models
  - Model joint distribution $p(x, y)$
  - Model class-conditional densities $p(x|y)$ for each individual class
  - E.g.: Naive Bayes

Other Topics Covered

- Generative models for discrete data
  - Beta-Binomial model
  - Dirichlet-Multinomial model
  - Naive Bayes
- Classifiers for continuous data
  - Logistic regression
  - Softmax regression
  - Gaussian Naive Bayes
- Point Estimation
  - Maximum Likelihood Estimation (MLE) and Maximum a Posteriori (MAP)
  - Empirical Risk Minimization (ERM)
  - Gradient-Based Optimization
  - Regularized Risk Minimization (RRM)
- Dimensionality Reduction
  - Matrix Decompositions (and SVD)
  - Latent Linear Models

Application: Naive Bayes

Given the MSINT database, how can we classify images of handwritten digits?

Figure: 150 digits grouped by their respective class labels using a Naive Bayes model with $\alpha = 2$

Image Classsification Results

- Accuracy rate of approximately 85%.
- Computationally efficient.
- Assumes conditional independence.

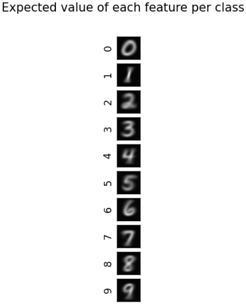

Figure: Expected values of the digits grouped by the class labels.

Voronoi Diagrams

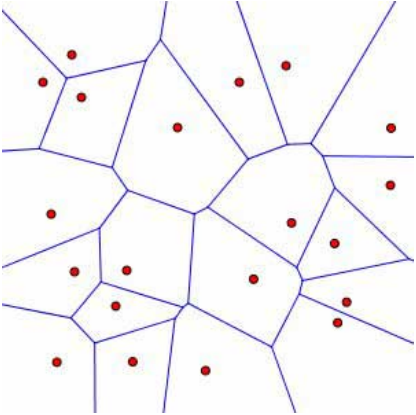

Figure: Example of a Voronoi Diagram

Definition of a Voronoi Diagram

Let P be a set of points in the plane (called *sites* or *seeds*)

- Cells/Regions
  - $V(p) = x \in \mathbb{R}^2 : |xp| < |xp| \, \forall \, q \in P$
  - The region encapsulates the space closest to each point
- Edges
  - $V(p, p') = x : |xp| = |xp'|$ and $|xp| < |xq| \, \forall \, q \neq p, p'$
- Vertices
  - $V(p, p', p'') = x : |xp| = |xp'| = |xp''|$ and $|xp| \leq |xq| \, \forall q$

Properties of Voronoi Diagrams

- Given n sites, $Vor(P)$ consists of at most $2n-5$ vertices and $3n-6$ edges
  - Using Euler's formula for a planar graph
- Largest empty circle lies at a vertex
- Duality of Delaunay Triangulations

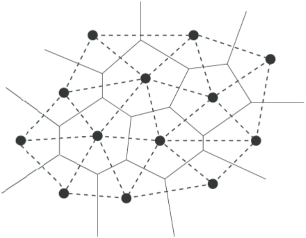

Figure: Delaunay Triangulations on a Voronoi Diagram

Computing Voronoi Diagrams

- Brute force
  - For each $p \in P$, compute the hyperplane intersection.
  - $O(n \log^2 n)$ time
- Beachline sweep: Fortune's algorithm
  - Parabolic computations for each site
  - $O(n \log n)$ time
- Voronoi Computation Demo

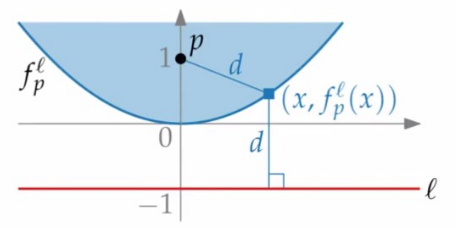

Figure: f_p^l is the parabola with focus p and directrix l.

Voronoi Diagrams using Python

- Using numpy, scipy.spatial, skimage, pyclesperanto (GPU needed)

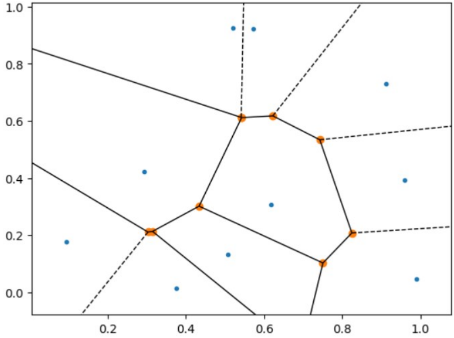

Figure: Voronoi diagrams generated with python from a set of random coordinates.

Convolutional Neural Networks

- Area of deep learning specializing in pattern recognition
- Consists of convolutional layers
  - Receives input, transforms it, then outputs to next layer
  - Performs a convolution operation
  - Dot product of weights and neutrons + bias
  - $a^1 = \sigma(Wa^0 + b)$
- Filters in the convolutional layer perform pattern recognition
- Application of filters increase with each added layer

Convolutional Neural Networks

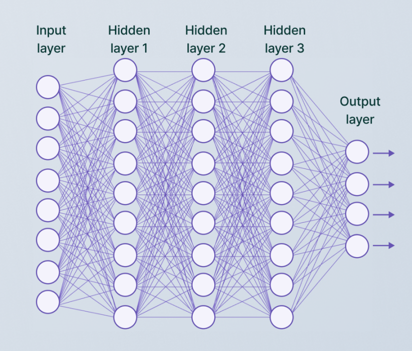

Figure: Overview of a CNN

Input

0	1	1	0	1
0	1	1	0	1
0	1	1	0	1
0	1	1	0	1
0	1	1	0	1

Filter / Kernel

1	0	1
1	1	1
0	0	1

Figure: Filter in a convolutional layer

Questions and Further Explorations

What are our research goals?

- Increasing accuracy
- Increasing efficiency and computational power
- Exploring different application avenues
- Link between CNNs and Voronoi diagrams
- Scope and timeline