SMAI Class-2

August 1, 2022

Feature Spaces

Problem: Classify two fruits

What feature can we extract from fruits (apples & orange) to differentiate them?

Features that we can take

- color: color itself is complex so we might measure *redness*
- weight

Each feature can be thought of as a vector and the samples will be points on the feature space (here 2D space).

Points = $[c_i w_i]^T$.

We essentially forget about raw data (after constructing the feature space).

Features we can extract

- Quantitative (Numeric)
- Discrete
- Continous
- Mixed (**Example** color: discrete, weight: continous)
- Ordinal (Example: Low, Medium, High)
  - There is an order between them but we don't know it quantitatively
- Categorical / Nominal (Example: Green, Red, Blue)
  - No order between them.

Nearest Neighbour (NN) Classifier

Summary

- Have to go over all training samples, to compute distances (during testing).
- Distance metric (and weights for each label) matters.
- Error rate (for kNN) is atmost twice that of the ideal classifier.
- Time complexity high during testing
- Ability to do Continous Learning

READ: Voronoi Tessalation to define boundary (between classes). These boundaries are called Decision Boundaries. We call the boundary as hyperplane as well. In 3D feature space we'll have 3D-hyperplane.

kNN

- Take majority class from k NNs.
- k is a user-selectable parameter (hyperparameter).

What if there is a tie?

- **Weighted kNN**: Weigh according to nearness. The sample that is close can be given more weightage etc.
- Random break the tie

Another Approach: Sample Pruning

Remove samples that do not affect the decision boundary. Prevents computing a ton of distances from the test point.

Nearest Mean Classifier

Further Approach: **Keep sample means** (*Nearest Mean Classifier*). The boundary has also reduced in complexity.

Fast/Approximate NN

Quick search for NN with possible errors.

READ: - KD Tree - Cut down on branches that are not going to be the NN etc. - Ball Tree - Circle near points together etc.

Approximate in the sense that the NN might not be the closest. But it is quicker and gives closest NN in a bound (like top 5 closest NN).

Distance Metrics & Iso-surfaces

Euclidean Distance

- In matrix/vector form: $d(P,Q)^2 = (P-Q)^T(P-Q)$
- In feature form (we know sum of differences)

Iso-surfaces: Points equidistant from a sample.

For euclidean distance, iso-surfaces are concentric circles.

Manhattan Distance

Iso-surfaces will look like diamonds.

Minkowski Distance (generalization of the first two)

Compute $d(P, Q)^r$.

REFER: Lu et al "The Minkowski Approach".

Mahalanobis Distance

$d(P, Q)^2 = (P - Q)^T S^{-1}(P - Q)$.

Isosurface tends to be ellipses.

Hamming Distance (for binary features)

Check how many features are equal (and take sum).

Cosine Distance (dot product between the vectors)

1 - dot product value will give distance.

Data Preprocessing

- Feature Normalization