4.6 METRICS AND NEAREST-NEIGHBOR CLASSIFICATION

Li Yu, Hongda Mao & Joan Wang

The properties of a metric

- $D(a,b)$ – the distance between $a$ and $b$
  - Non-negativity: $D(a,b) \geq 0$
  - Reflexivity: $D(a,b)=0$ if and only if $a=b$
  - Symmetry: $D(a,b)=D(b,a)$
  - Triangle inequality: $D(a,b)+D(b,c) \geq D(a,c)$

Example: Minkowski Metric ($L_k$ norm)

- The distance between $a$ and $b$ in $d$ dimensions
  
  $L_k(a,b) = \sqrt[d]{\sum_{i=1}^{d} |a_i - b_i|^k}$

- What if $k=1$ ($L_1$ norm)
  - Manhattan or city block distance

- What if $k=2$ ($L_2$ norm)
  - Euclidean distance

$L_1$ Norm

- $d$-dimensional

  $L_1(a,b) = \sum_{i=1}^{d} |a_i - b_i|$

- 1 dimensional

  $L_1(a,b) = |a-b|$

  $L_1$ norm is also called Manhattan or city block distance

$L_2$ Norm

- $d$-dimensional

  $L_2(a,b) = \sqrt{\sum_{k=1}^{d} (a_k - b_k)^2}$

- 2 dimensional

  $L_2(a,b) = \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2}$

  $L_2$ norm is the Euclidean distance

  Axis rescale problem with Euclidean distance
Another example in taxonomy: Tanimoto Metric

- The distance between two sets \( S_1 \) and \( S_2 \)

\[
D_{\text{Tanimoto}}(S_1, S_2) = \frac{n_1 + n_2 - 2n_{12}}{n_1 + n_2 - n_{12}}
\]

Where
- \( n_1 \) – number of elements in \( S_1 \)
- \( n_2 \) – number of elements in \( S_2 \)
- \( n_{12} \) – number of elements in both \( S_1 \) and \( S_2 \)

Tanimoto Coefficient

- The similarity between two “fingerprints” \( S_1 \) and \( S_2 \)

\[
T = \frac{n_{12}}{n_1 + n_2 - n_{12}}
\]

Where
- \( n_1 \) – number of features in \( S_1 \)
- \( n_2 \) – number of features in \( S_2 \)
- \( n_{12} \) – number of common features

- Widely used in biology and chemistry to compare species/molecules
- “fingerprints” could be coded molecular structure [1], gas chromatograms [2], etc

Drawbacks of using a particular metric

- There may be drawbacks inherent in the uncritical use of a particular metric in nearest-neighbor classifiers.

Example:
1. Consider a 100-dimensional pattern \( x' \) representing a 10x10 pixel grayscale image of a handwritten 5.

2. Computing the Euclidean distance from \( x' \) to the pattern representing an image that is shifted horizontally but otherwise identical.

3. Computing the Euclidean distance from \( x' \) to an unshifted 8.

4. Making a comparison between the two Euclidean distances


Discussions

- Like the horizontal transformation, other transformations, such as overall rotation or scale of the image, would not be well accommodated by Euclidean distance in this manner.
- Such drawbacks are especially pronounced if we demand that our classifier be simultaneously invariant to several transformations, such as horizontal translation, vertical translation, overall scale, rotation, line thickness, and so on.
- One remedy:
  - Preprocess the images by shifting their centers to coalign, then have the same bounding box, and so forth.
  - Sensitivity to outlying pixels or to noise

Ideally, during classification we would like to first transform the patterns to be as similar to one another and then compute their similarity, for instance by the Euclidean distance. However, the computational complexity of such transformations makes this ideal unattainable.

Example

- Merely rotating a k x k image by a known amount and interpolating to a new grid is $O(k^2)$.
- We don't the proper rotation angle ahead of time and must search through several values, each value requiring a distance calculation to test whether the optimal setting has been found.
- Searching for the optimal set of parameters for several transformation for each stored prototype during classification, the computational burden is prohibitive.

Tangent Distance

- The general approach in tangent distance classifiers is to use a novel measure of distance and a linear approximation to the arbitrary transforms.
- Construction of the classifier:
  - Perform each of the transformations $F_i(x'; a_i)$ on the prototype $x'$
  - Construct a tangent vector $TV_i$ for each transformation:
    - $TV_i = F_i(x'; a_i) - x'$
  - $TV_i$ can be expressed as a $1 \times d$ vector
  - We can construct a $r \times d$ matrix $T$: Here $r$ is the number of transformations $d$ is the number of dimensions

The small red number in each image is the Euclidean distance between the tangent approximation and the image generated by the unapproximated transformations.

Tangent Distance

- Computing a test point $x$ to a particular stored prototype $x'$.
  - The tangent distance from $x'$ to $x$ is:
    $$D_{tang}(x', x) = \min_a ||(x' + Ta) - x||$$
  - "one-sided" tangent distance,
    - Only one pattern is transformed.
  - "two-sided" tangent distance,
    - Both of the two patterns are transformed. Although it can improve the accuracy, it brings a large added computational burden.

Finding the minimum distance

- The Euclidean distance:
  $$D^2(x' + Ta, x) = ||(x' + Ta) - x||^2.$$  
- Computing the gradient with respect to the vector of parameters $a$, The projections onto the tangent vectors— as
  $$\nabla_a D^2(x' + Ta, x) = 2T(x' + Ta - x).$$

According to the gradient Descent method, we can start with an arbitrary $a$ and take a step in the direction of the negative gradient, updating our parameter vector as:
$$a(t + 1) = a(t) - \eta \nabla_a D^2(x' + Ta(t) + x' - x).$$
Gradient descent methods [3][4]

Gradient descent is an optimization algorithm. To find a local minimum of a function using gradient descent, one takes steps proportional to the negative of the gradient of the function at the current point.


4.7 FUZZY CLASSIFICATION

What is fuzzy classification

Using informal knowledge about problem domain for classification

• Example:
  - Adult salmon is oblong and light in color
  - Sea bass is stouter and dark

• Goal:
  - Convert objectively measurable parameters to “category membership” function
  - Then use this function for classification

Categories V.S. Classes

- Categories here do not refer to final classes
- Categories refer to ranges of feature values
- e.g. lightness is divided into five “categories”
  - Dark
  - Medium-dark
  - Medium
  - Medium-light
  - Light

Conjunction Rule

- With multiple “category memberships”, we need a conjunction rule to produce a single discriminate function for classification
- Many possible ways of merging e.g. for two membership functions $\mu_x$ and $\mu_y$,

$$\mu_x(x) \cdot \mu_y(y)$$


- Three membership functions: soil, water, vegetation
- Then summed up to form the discriminant function

Category membership functions V.S. probabilities

- Category membership functions do not represent probabilities
- E.g. half teaspoon of sugar placed in tea
  - Implying sweetness is 0.5
  - Not probability of sweetness is 50%

Limitations of fuzzy methods

- Cumbersome to use in
  - High dimensions
  - Complex problems
- Amount of information designer can bring is limited
- Lack normalization thus poorly suited to changing cost matrices
- Training data not utilized (but there are attempts [5])
- Main contribution: Converting knowledge in linguistic form to discriminant functions

4.9 Approximations by Series Expansions

Drawbacks of Nonparametric Methods

- All of the samples must be stored
- The designer have extensive knowledge of the problem
- Example:

Modified Parzen-window procedure

- Basic idea: approximate the window function by a finite series expansion that is acceptably accurate in the region of interest.
  \[ \psi \left( \frac{x - x_i}{h_k} \right) = \sum_{j=1}^{n} a_j \psi_j(x) \psi_j(x_i) \]
- Split the dependence upon x and xi

From Eq. 11 we have

Modified Parzen-window procedure

\[ \sum_{i=1}^{n} \psi \left( \frac{x - x_i}{h_k} \right) = \sum_{j=1}^{n} a_j \psi_j(x) \sum_{i=1}^{n} \psi_j(x_i) \]

Then from Eq. 11 we have

\[ p_{n}(x) = \frac{1}{n \sum V_i} \frac{1}{h_k} \psi \left( \frac{x - x_i}{h_k} \right) = \sum_{j=1}^{n} b_j \psi_j(x) \]

\[ b_j = \frac{a_j}{n \sum \psi_j(x)} \]
Taylor series

- There are many types of series expansions that can be used.
- Taylor series is a representation of a function as an infinite sum of terms calculated from the values of its derivatives at a single point.

\[ f(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!} (x-a)^n \]

\[ e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \ldots \]

Evaluation of Error

- We have \[ \sqrt{n} p_n(x) = e^{x^2} - \sum_{m=0}^{\infty} \frac{(-1)^m x^{2m}}{m!} \]

The quality of the approximation is controlled by the remainder term

\[ R_n(x) = e^{x^2} - \sum_{m=0}^{n} \frac{(-1)^m x^{2m}}{m!} \leq \frac{e^{x^2}}{m!} \cdot \left( \frac{|x|^2}{1} \right)^m \cdot \left( \frac{4}{e} \right)^m \]

Stirling’s approximation

- Roughly, this means that these quantities approximate each other for all sufficiently large integers \( n \).
Limitations

- In a polynomial expansion we might find the terms associated with an \( x_i \) far from \( x \) contributing most (rather than least) to the expansion.

\[
\sqrt{\pi} \left( \frac{x-x_i}{h} \right)^2 \approx 1 - \left( \frac{x-x_i}{h} \right)^2 = 1 + \frac{2}{h^2} xx_i - \frac{1}{h^2} x^2 - \frac{1}{h^2} x_i^2
\]

- The error becomes small only when \( m > e(r/h)^2 \). It needs for many terms if the window size \( h \) is small relative to the distance \( r \) from \( x \) to the most distant sample. Attractive when the large window.

\[
\frac{1}{\sqrt{\pi h^2 2m}} \left( \frac{e}{m} \right)^{1/2} > m > e(r/h)^2
\]