CSC421 Artificial Intelligence

Supervised Learning II

George Tzanetakis

University of Victoria

2021

Table of Contents I

1 Baseline and Nearest Neighbor Classifiers

2 Generative Probabilistic Approaches

Dummy Classifiers for baseline

Dummy classifiers typically only utilize information about the number of classes and their prior distribution in the training data without taking into account any of the features. They are useful as a baseline to show if the metrics achieved by a particular algorithm for a particular problem are better than the ones obtained with a dummy classifier.

- Random Self-explanatory
- ZeroR Everything is predicted as the most popular class in the training set.
- RandomPrior Random class labels drawn based on prior distribution of labels.

Nearest Neighbor

Main Idea

Be lazy during training doing nothing other than storing the labeled feature vectors. Only perform modeling locally when a prediction is needed. Simple and intuitive method: Find K nearest neighbors of a d-dimensional point. The predicted class label is the majority of the class labels of these neighbors. Who needs a model !

Related Information

- Related to the **Rote Classifier** which only classifies examples that exactly match one of the training samples
- Eager vs Lazy Learners
- Online vs Batch Learning
- Need for a similarity measure

K-NN Voting

Definition

$$Majority \quad Voting: y' = \arg\max_{v} \sum_{(\mathbf{x}, y_i) \in D_x} I(v = y_i) \tag{1}$$

Definition

$$DistanceWeighted \quad Voting: y' = \arg\max_{v} \sum_{(\mathbf{x}, y_i) \in D_x} w_i \times I(v = y_i) \tag{2}$$

$$w_i = 1/d(\mathbf{x'}, \mathbf{x_i})^2 \tag{3}$$

where v is a class label, y_i is the class label for one of the nearest neighbors, and $I()$ is an indicator function that returns the value 1 if its artument is true and 0 otherwise.

Similarity (Proximity) Measures

Easy to transform similarities to dissimilarities and vice versa. **Euclidean Distance** between points in space:

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{k=1}^{n} (x_k - y_k)^2} \tag{4}$$

Generalizes to **Minkowski** distance metric:

$$d(\mathbf{x}, \mathbf{y}) = \sqrt[1/r]{\sum_{k=1}^{n} (|x_k - y_k|)^r} \tag{5}$$

$L1$ Manhattan Distance, $L2$ Euclidean, and L_{inf} are common variants. Metrics have to obey: positivity, symmetry, triangle inequality.

Cosine Similarity

Measure of similarity between two vectors measuring the angle between them. Frequently used in document retrieval where only the relative percentages of each term frequency matter rather than their absolute values.

$$cos(\theta) = \frac{A * B}{||A||||B||} \tag{6}$$

Mahalanobis Distance

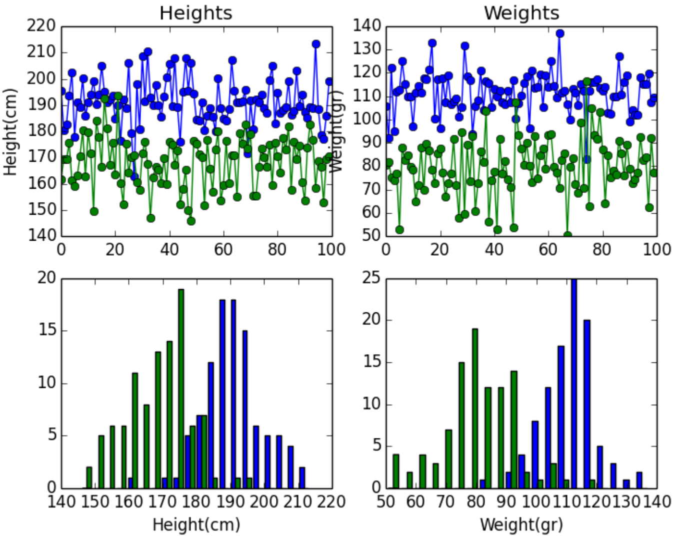
Normalize the distance metric to correlations and different dynamic ranges (prompted by the problem of identifying similarities of skulls based on measurements in 1927 - analysis of race mixture in Bengal):

$$D_M(x) = \sqrt{(x-\mu)^T \Sigma^{-1} (x-\mu)} \tag{7}$$

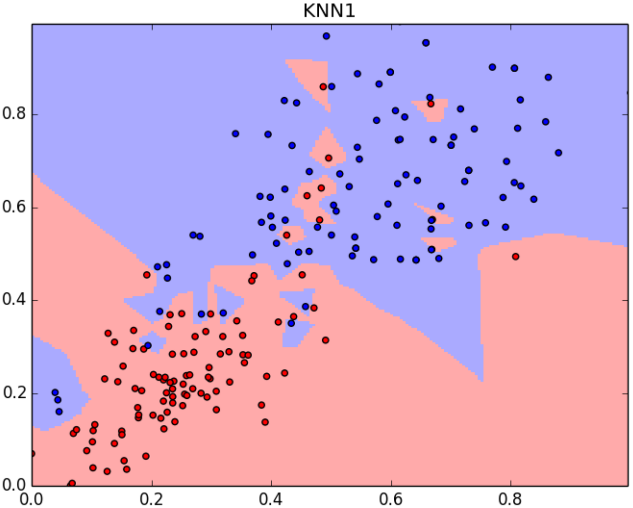
Characteristics of K-NN Classifiers

- Although training is easy (no model building), classifying a test example can be quite expensive as the similarity to all the points in the training data needs to be calculated.
- Tends to be succeptible to noise
- Arbitrarily shaped decision boundaries that depend on choice of k
- Curse of dimensionality - when the # of dimensions is > 10 then distances become meaningless as all points have approximately the same distance to the query.
- Excellent asymptotic consistency results - as the number of instances approaches infinity the KNN algorithm is guaranteed to have an error no more than twice the theoretically optimum *Bayes Error Rate*.

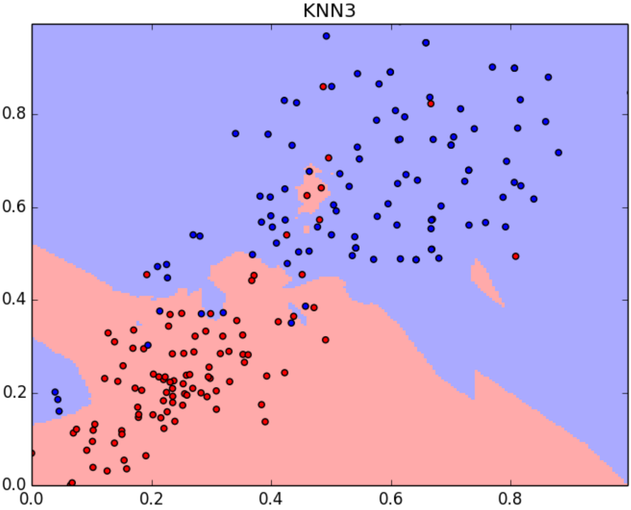
The hoops dataset



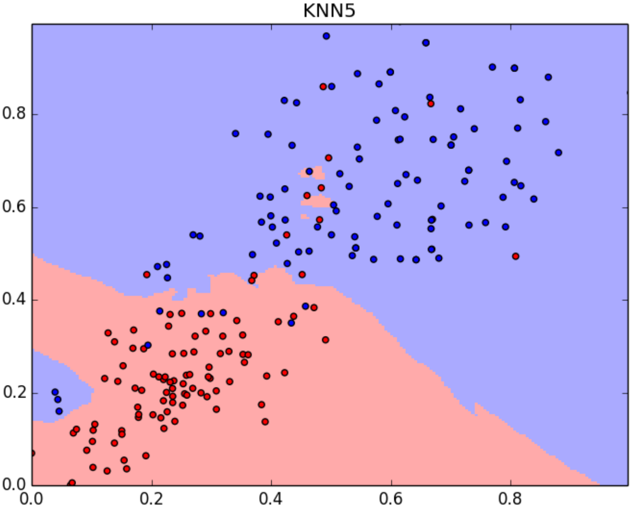
1-NN Decision Boundary



3-NN Decision Boundary



5-NN Decision Boundary



Plot Generation

Steps for generating plot

- Plot training set points with colors indicating class labels
- Train 3-NN classifier
- Predict the classes of a dense grid of points (pixels)
- Color based on the predicted class to visualize decision boundary

Plot Generation

Steps for generating plot

- Plot training set points with colors indicating class labels
- Train 3-NN classifier
- Predict the classes of a dense grid of points (pixels)
- Color based on the predicted class to visualize decision boundary

Plot Generation

Steps for generating plot

- Plot training set points with colors indicating class labels
- Train 3-NN classifier
- Predict the classes of a dense grid of points (pixels)
- Color based on the predicted class to visualize decision boundary

Plot Generation

Steps for generating plot

- Plot training set points with colors indicating class labels
- Train 3-NN classifier
- Predict the classes of a dense grid of points (pixels)
- Color based on the predicted class to visualize decision boundary

Plot Generation

Steps for generating plot

- Plot training set points with colors indicating class labels
- Train 3-NN classifier
- Predict the classes of a dense grid of points (pixels)
- Color based on the predicted class to visualize decision boundary

Table of Contents I

1 Baseline and Nearest Neighbor Classifiers

2 Generative Probabilistic Approaches

Generative Approaches

Main Idea

Transform the problem of classification to the multiple subproblems. Each subproblem consists of estimating the parameters of a model capable of generating samples similar to the ones associated with a particular class. In order to transform the problem and perform the model estimation we first have to review some concepts from probability and statistics.

Probabilities

Probability summarizes (with a value between 0 and 1) the uncertainty we have about the world. Probabilities (between 0 and 1) correspond to intermediate degrees of belief in the truth of a sentence. Important: the sentence itself is either true or false. Degree of belief is different than degree of truth and depends on our current state of knowledge.

Consider drawing a card and asking what is the probability it is the card is the ace of spades. Before looking at the card the probability is $\frac{1}{52}$ but after it will either be 0 or 1. Therefore, all probability statements should indicate the evidence used to obtain the probability estimate. The probability before evidence is obtained is termed the prior (or unconditional) and after evidence the posterior (or conditional).

Probability Notation

Random Variables represents a "part" of the world whose "status" is initially unknown. Each random variable has a domain that it can take on. For example, the **RV** *Weather* can have the values: *sun*, *rain*, *cloud*, *snow*. Probabilities are assigned over values in the domain. The notation $\mathbf{P}(Weather)$ denotes a vector of values for the probabilities of each individual state of the weather:

$$\begin{aligned}
P(Weather = sunny) &= 0.65\\
P(Weather = rain) &= 0.25\\
P(Weather = cloudy) &= 0.07\\
P(Weather = snow) &= 0.03\\
\mathbf{P}(Weather) &= (0.65, 0.25, 0.07, 0.03)
\end{aligned}$$

Joint Probability Distribution

Joint Probability Distribution

Complete set of RVs used to describe the problem can be represented as the joint probability distribution. For example the joint distribution $P(Weather, Raincoat, Season)$ can be represented as a $2x2x4$ table.

Continuous RVs

Continuous Random Variables

For continuous variables it is not possible to write out the distribution as a table as it would have infinite many values. Instead, the probability that a random variable takes on some value x is represented as a parameterized function of x. For example we could have $P(X = x) = U[5, 20](x)$ express the belief that the temperature tomorrow will be uniformly distributed between 5 and 20 degrees Celcius. Notice that P(X = 18) should not be interpreted as the probability that the temperature is exactly 18 degrees which is zero. Instead, the temperate in a small region around 18 degrees is equal to the value in the limit. Sometimes lower case $p(x)$ is used to differentiate continuous (density) from discrete distributions.

Conditional Probability

Definition

$P(a/b)$ where a and b are propositions. The probability of a given that all that we know is b. The conditional probability can be defined in terms of unconditional probabilities as follows:

$$P(a/b) = \frac{P(a, b)}{P(b)} \tag{8}$$

Conditional Probability Notation

Notation

$$P(X = x_1, Y = y_1) = P(X = x_1 / Y = y_1)P(Y = y1) \quad (9)$$

$$P(X = x_1, Y = y_2) = P(X = x_1 / Y = y_2)P(Y = y2) \quad (10)$$

$$\ldots \quad (11)$$

can be combined with the notation denoting a set of equations:

$$\mathbf{P}(X, Y) = \mathbf{P}(X/Y)\mathbf{P}(Y) \quad (12)$$

Bayes Rule

Definition

$$\mathbf{P}(Y|X) = \frac{\mathbf{P}(X/Y)\mathbf{P}(Y)}{\mathbf{P}(X)} \tag{13}$$

Example

Suppose L is a rv corresponding to people with lung cancer in a population and S is a rv corresponding to the smokers. We have the following data: $P(L) = 0.001$, $P(S/L) = 0.9$, $P(S/\hat{L}) = 0.21$. P(L/S) corresponds to the percent of smokers who have lung cancer and can be calculated using the Bayes theorem:

$$P(L/S) = \frac{P(S/L)P(L)}{P(S)} = \frac{0.0009}{0.9 + 0.21} = 0.0043 \tag{14}$$

What's the big deal ?

Bayes theorem allows us to "choose" in a particular problem the conditional probabilities that are easier to calculate. For example it is easier to obtain the probability that someone who has lung cancer is a smoker than the probability that a smoker has lung cancer.

Bayes Classification

$$P(Y/\mathbf{X}) = \frac{P(\mathbf{X}/Y)P(Y)}{P(\mathbf{X})} \tag{15}$$

where Y is the class label and $\mathbf{X}$ is the feature vector. Notice that this is a set of equations, one for each class label in Y. Therefore there will be L posterior probabilities one for each class. To classify a test instance a *Bayesian* classifier computes these posterior probabilities and selects the class label corresponding to the maximum posterior. Main challenge becomes how to estimate $P(\mathbf{X}/Y)$ from the labeled training samples. For each class the corresponding training samples are used to estimate the parameters of the corresponding pdfs.

Bayes Classification

$$P(Y/\mathbf{X}) = \frac{P(\mathbf{X}/Y)P(Y)}{P(\mathbf{X})} \tag{15}$$

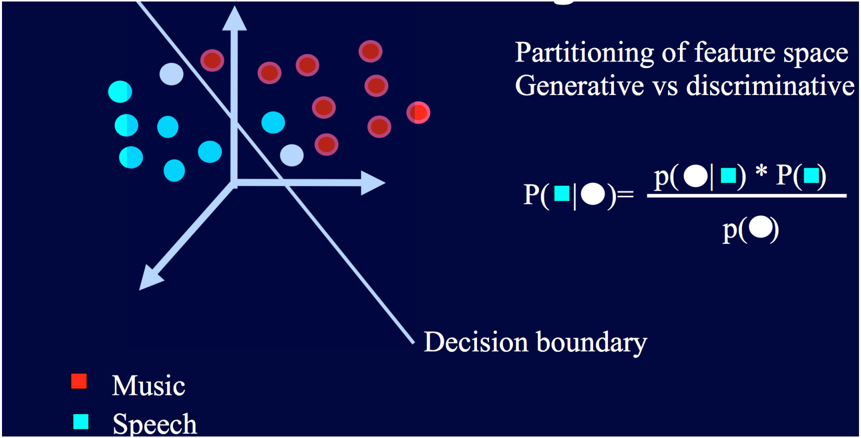
where Y is the class label and $\mathbf{X}$ is the feature vector. Notice that this is a set of equations, one for each class label in Y. Therefore there will be L posterior probabilities one for each class. To classify a test instance a *Bayesian* classifier computes these posterior probabilities and selects the class label corresponding to the maximum posterior. Main challenge becomes how to estimate $P(\mathbf{X}/Y)$ from the labeled training samples. For each class the corresponding training samples are used to estimate the parameters of the corresponding pdfs.

Bayes Classification

$$P(Y/\mathbf{X}) = \frac{P(\mathbf{X}/Y)P(Y)}{P(\mathbf{X})} \tag{15}$$

where Y is the class label and $\mathbf{X}$ is the feature vector. Notice that this is a set of equations, one for each class label in Y. Therefore there will be L posterior probabilities one for each class. To classify a test instance a *Bayesian* classifier computes these posterior probabilities and selects the class label corresponding to the maximum posterior. Main challenge becomes how to estimate $P(\mathbf{X}/Y)$ from the labeled training samples. For each class the corresponding training samples are used to estimate the parameters of the corresponding pdfs.

Bayes Classification



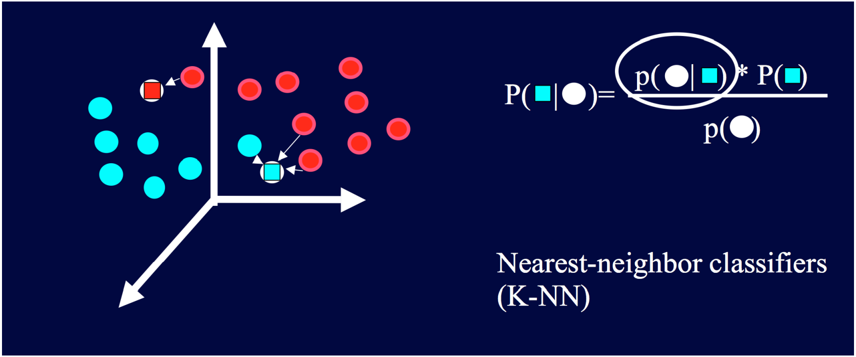
Partitioning of feature space
Generative vs discriminative

$$P(\blacksquare|\bullet)=\frac{p(\bullet|\blacksquare) * P(\blacksquare)}{p(\bullet)}$$

Decision boundary

🟥 Music
🟦 Speech

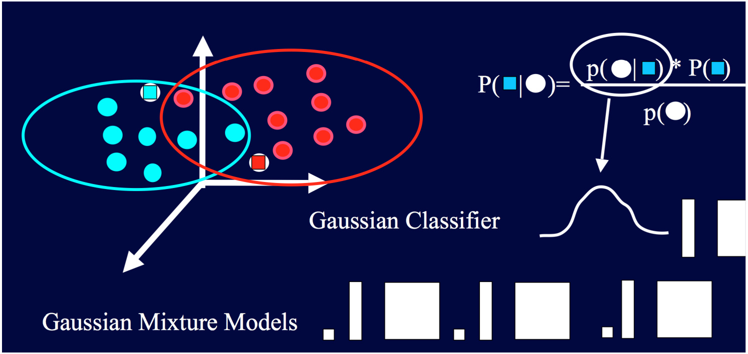
Bayes Classification - KNN



$$P(\blacksquare \mid \bullet) = \frac{\boxed{p(\bullet \mid \blacksquare)} * P(\blacksquare)}{p(\bullet)}$$

Nearest-neighbor classifiers (K-NN)

Bayes Classification - GNB GMM



Bayes Error Rate

Definition

Assume that we know the true probability distribution that governs P(**X**—Y). The Bayes error rate is the area in which the class distributions overlap. For a given feature space, the Bayes Error Rate is a lower bound on the error rate that can be achieved by any pattern classifier acting on that space. In general it can only be known directly if all the class priors and class-conditional likelihoods are known. The reason is that some of the error is inherent due to overlapping class densities but some additional error can creep in because of deficiencies in the classifier and limitations of the training data.

Naive Bayes Classifier

Definition

A Naive Bayes classifier estimates the class-conditional probability bu assuming that the attributes are conditionally independent, given the class label y. More formally:

$$P(\mathbf{X}|Y = y) = \prod_{i=1}^{d} P(X_i|Y = y) \tag{16}$$

Example

If we know that someone is a professional basketball player then the probability they have a certain height does not depend on their weight (this is the naive assumption). However the height of person does depend on their weight if we don't know whether they are basketball players or not.

Naive Bayes Classifier

The naive assumption about conditional independence of the attributes allows us to model each attribute separately. For categorical attributes estimating the conditional probabilities is straightforward counting based on the training data. For the continuous attributes it is typically assumed that the probability density function follows a certain parametric form and the task is to estimate the corresponding parameters. A common choice is the Gaussian distribution characterized by two parameters, its mean μ, and variance σ^2. For each class y_il, the class-conditional probability for attribute X_i is:

$$P(X_i = x_i | Y = y_j) = \frac{1}{\sqrt{2\pi\sigma_{ij}}} \exp^{-\frac{(x_i - \mu_{ij})^2}{2\sigma_{ij}^2}} \tag{17}$$

Likelihood

University of Victoria
Computer Science

Definition

The likelihood of a set of data for a particular statistical model assumes that the data is i.i.d (independent and identically distributed). That means that:

$$P(\mathbf{x}|M) = \prod_i P(x_i|M) = \prod_i P(x_i|\theta), \tag{18}$$

where θ is a vector of parameters characterizing the model M and x_i are the data points. If we consider the equation above as a function of the θ for the data point x_i we obtain:

$$\begin{align} L(\theta|x_1, \ldots, x_n) &= \textstyle\prod_i P(x_i|\theta) \tag{19} \\ \ln(\theta|x_1, \ldots, x_n) &= \ln \textstyle\prod_i P(x_i|\theta) = \sum_i \ln P(x_i|\theta) \tag{20} \end{align}$$

Maximum Likelihood Estimation of Parameters

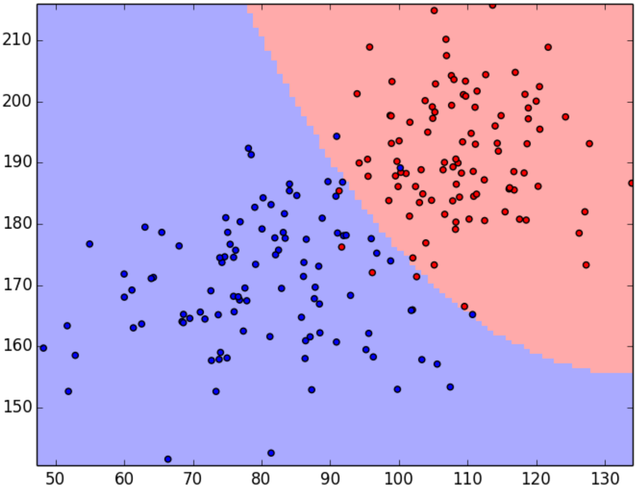
Definition

A parametric statistical model is a collection of probability distribution functions each of which is uniquely characterized by a finite dimensional vector of parameters (θ). For example Gaussian distributions are characterized by their mean and covariance matrix.

Definition

Given a dataset and set of models find the set of parameters that maximizes the likelihood of observing the data given the model. In well-behaved cases this estimation can be performed analytically rather than numerically using optimization methods. For example the ML estimate of the mean and covariance matrix are the sample mean and sample covariance.

The hoops dataset (Naive Bayes Gaussian)



Characteristics of Naive Bayes Classifiers

- Robust to isolated noise points especially when there is a lot of data
- Robust to irrelevant attributes as $P(X_i/Y)$ becomes almost uniformly distributed and therefore has little impact to the posterior probability.
- Correlated attributes can degrade the performance
- Very fast to train and predict

EM-Algorithm

Definition

The Expectation Maximization (EM) algorithm is a technique that finds maximum likelihood estimates in parametric models with incomplete data. It has many applications in machine learning such as unsupervised clustering, gaussian mixture modeling, handling missing values, learning hidden (latent) variables in Bayesian Networks, semi-supervised learning and others.

EM-algorithm

Definition

The EM-algorithm uses a set of interlocking equations in which the solution to the parameters requires the values of the latent variables and vice versa. It is an iterative (and not guaranteed to be optimal) procedure that finds the maximum likelihood estimates of the parameter vector by repeating two steps:

- E-step Assuming a known model characterized by parameters θ estimate the expected values of the missing data.
- M-step Treating the estimated values of the missing data as actual values re-estimate using ML the parameter vector θ.
- Repeate the E-step and M-step until there is little change in the parameter vector θ.

Digression - Sufficient Statistics

Definition

A statistic is a single measure calculated on a set of samples. number

Definition

A sufficient statistic for a particular model and associated unknown parameter is a measure that provide all the information needed for computing the value of that parameter.

Example

For a normal distribution $N(\mu, \sigma^2)$ with $\theta = (\mu, \sigma^2)$ the sum of samples and sum of squares $(\sum x_i, \sum x_i^2)$ are sufficient statistics.