Feature Selection

Week	TUES. Week 2
Assignment Due	
Assignment Done	☐
Due Date	
Notes Done	☑

Presentation

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/f1886e41-1690-41ea-98db-73997aa8d8e2/L1-FeatureSelection.pdf

Class Notes

Feature Selection

Given a set of features, some features are more important than others.

- We typically want system to make predictions.
    - In trying to figure out best features, we can do an **exhaustive (naive) approach**:
        - Can train a system for each possible combinations of features, finding out which do the best and use these features (**feature set**)
        - This can gives a globally optimal solution.
        - Doing this, however, is NOT feasible, especially, when the number of features is high.
            - $\sum_{i=0}^{D} \binom{D}{i} = 2^D$ possible systems (if features are binary)

- Ex: 2 binary features would have 4 possible values that they could undertake that we'd need to create to train our system on all possible combinations.

- Another idea is to do a **greedy approach**:
    - We can start with NO features and train a system with this.
        - Will likely do very poorly.
    - For each iteration t, for each remaining feature j
        - Run learning algorithm for features $F \cup \{j\}$
        - Ex: Would look like this:
            - $F = \{\}, f_1, f_2, f_3$
            - Train system with each feature used individually:
                - $\{f_1\}, \{f_2\}, \{f_3\}$
                - Say that f_2 improved performance.
            - Continue training system with the remaining features:
                - $F = \{f_2\}$
                - $\{f_2, f_1\}, \{f_2, f_3\}$
                - Say that $\{f_2, f_1\}$ has better performance.
            - Continue training:
                - $F = \{f_2, f_1\}$
                - At this point, we can stop and determine if adding another feature worsens performance or not.
                    - It may be the case that we stop earlier in our search since performance may worsen here.
    - This would involve training/evaluating $D + (D-1) + (D-2) + 1 = \sum_{i=1}^{D} i = \frac{D(D+1)}{2}$ systems total, ending up with D^2.
- We could think about whether we could **rank the features (separability feature selection)**.

        - Instead of having to add each feature at a different iteration, can we just add in the highest ranked one?
            - Here, the question is: **How do we rank our features?**
            - If we don't have class features, we could use something like covariance or standard deviation to rank the features.
                - Ultimately, we want features that are unique so using some combo of deviations within the feature and similarity with other features to be reduced.
            - If we have supervised information, class labels, then we can maybe rank the feature by how well they do in separating the data by class.
                - We'll focus on this!!!

Entropy-Based Feature Selection

Given a set of features, some features are more important than others:

- **<u>Entropy</u>** is randomness in a system:
    - Say if we have k different possibilities with a probability of occurring, then we can compute the entropy of the set, based on the following formula.

$$H\big(P(v_1),\dots,P(v_K)\big) = \sum_{i=1}^{K}\left(-P(v_i)\log_K P(v_i)\right)$$

- <u>In calculating the entropy of tossing fair coin:</u>
    - The value of 1 means that this is a purely random system.
    - A deterministic system would have a probability of 0.

        - $v_1 = heads, v_2 = tails,$
        - $P(v_1) = 0.5, P(v_2) = 0.5$
        - $H\left(\frac{1}{2},\frac{1}{2}\right) = -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2} = 1$

- Entropy gives a measure of randomness in a system.

For our purposes, the outcomes are the class target labels:

- We start by taking the subsets generated by a feature, calculating the entropy of this subset, and calculate the ***weighted average entropy***.
- The feature that provides the lowest weighted average entropy for its subsets should be selected.

Weighted Average Entropy

Let H_i be the entropy of subset i.

Let $|C_i|$ be the number of observations in subset i.

Based on this, we can define the average entropy as:

$\mathbb{E} = \sum_{i=1}^{S} \frac{|C_i|}{N} H_i$

Entropy in general is calculated as:

$H(P(v_1), ..., P(v_k)) = \sum_{i=1}^{K} (-P(v_i) \log_K P(v_i))$

Example

$$X = \begin{bmatrix} 1 & 1 \\ 1 & 3 \\ 2 & 2 \\ 1 & 2 \\ 3 & 2 \\ 2 & 2 \end{bmatrix}, Y = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Feature 1 has 3 unique values, creating 3 subsets of data.

- From here, we're going to look at the class values of each subset.
  - Subset 1:
    - Feature observations: Class label 1, Class label 1, Class label 0
    - The probability of being class label 0 is $\frac{1}{3}$
    - The probability of being class label 1 is $\frac{2}{3}$
    - The entropy of this subset = $H_1 = -\frac{1}{3}\log_2(\frac{1}{3}) - \frac{2}{3}\log_2(\frac{2}{3})$
  - Subset 2:
    - Feature observations: Class label 1, Class label 0
    - The probability of being class label 0 is $\frac{1}{2}$
    - The probability of being class label 1 is $\frac{1}{2}$
    - The entropy of this subset resembles that of a fair coin flip: $H_2 = 1$
  - Subset 3:
    - Feature observations: Class label 0
    - This subset has no entropy (no randomness) → $H_3 = 0$
- Based on the entropy calculated, we can finally calculate the weighted average entropy:
  - Weighted average entropy = $\frac{3}{6}H_1 + \frac{1}{2}H_2 + \frac{1}{6}H_3$
    - First subset has 3 observations, so we weight it with $\frac{3}{6}$. We continue this with the rest of the subsets.

Feature 2 has 3 unique values as well, also creating 3 subsets of data.

- Subset 1 and 3 have no entropy.
- Subset 2, however, is the subset with the largest number of observations and has entropy within it.

Feature 1

$$\mathbb{E}_1 = \frac{3}{6}\left(-\frac{2}{3}\log_2\frac{2}{3} + -\frac{1}{3}\log_2\frac{1}{3}\right) + \frac{2}{6}\left(-\frac{1}{2}\log_2\frac{1}{2} + -\frac{1}{2}\log_2\frac{1}{2}\right) + \frac{1}{6}(0\log(0) - 1\log(1)) = 0.7925$$

Feature 2

$$\mathbb{E}_2 = \frac{1}{6}\left(-\frac{1}{1}\log_2 1 + -0\log_2 0\right) + \frac{4}{6}\left(-\frac{1}{4}\log_2\left(\frac{1}{4}\right) + -\frac{3}{4}\log_2\left(\frac{3}{4}\right)\right) + \frac{1}{6}\left(-\frac{1}{1}\log\left(\frac{1}{1}\right) - \frac{0}{1}\log\left(\frac{0}{1}\right)\right) = 0.5409$$

$$\boxed{X = \begin{bmatrix} 1 & 1 \\ 1 & 3 \\ 2 & 2 \\ 1 & 2 \\ 3 & 2 \\ 2 & 2 \end{bmatrix}, Y = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}}$$

We can see the feature 2 gives us less entropy, so we'd rank it higher in terms of features that we're considering.

- The second feature would be "more discriminatory," as it helps to discriminate the class of an observation better by having a lower entropy.

Entropy for Multiple Classes

These equations would hold even is there were more than 2 classes.

$$H\big(P(v_1), \ldots, P(v_K)\big) = \sum_{i=1}^{K}(-P(v_i)\log_K P(v_i))$$

$$\mathbb{E} = \sum_{i=1}^{S}\frac{|C_i|}{N}H_i$$

Here, we now consider the value K for the different classes, instead of 2, to ensure that our entropy values are between 0 and 1.

Entropy with Continuous-Valued Features

In a lot of datasets, we may have features that are not finite, rather have a continuous set of values that they can take:

- <u>There are two approaches to handling this:</u>
    - Break up the range of possible values Z internals and assign values to enumerated "bins", effectively converting continuous features into categorical-

ordinal feature (categorizing them)

- Work directly in the continuous space by assuming the data follows some ***distribution*** (ex: Normal, Gaussian)
  - Either by assumption or observation of the data

Normal-Gaussian Distribution

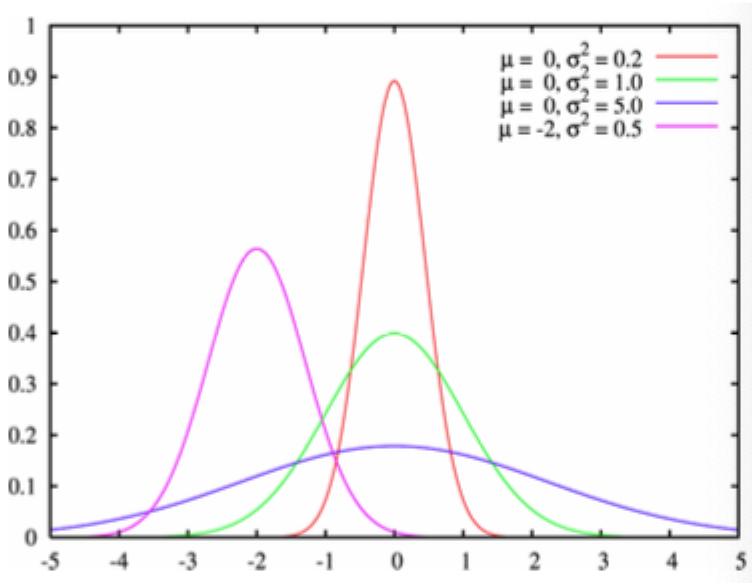
Throughout the course, we will often use the normal/Gaussian ***probability distribution function*** (PDF) to approximate the probability of generating continuous value x given some parameters (μ, σ).

- This value, which is ***proportional*** to the actual probability, is computed as:

$$P(x|\mu,\sigma) \propto p(x|\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}}^{e^{-\frac{(x-\mu)^2}{2\sigma^2}}}$$

$$P(x|\mu,\sigma) \propto p(x|\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

  - The probability of observing value x given this model for the distribution.
- The narrower the deviation (the steeper/higher the distribution), the larger the value of sigma is.
- Our intuition should state that if we use Gaussians to describe each feature of each class, that a good feature is one whose per-class Gaussians are well separated:



- Should we use the feature whose means are furthest apart?
  - Maybe, but what about the deviation?

- Possible intuition: Separate data by class, figure out parameters for this feature for a class
  - A good feature will possibly have its means far from one another for each class. They are effectively well-separated.
    - However, there can be overlap in a distribution being steeper and overlapping with a more gradual Gaussian.

- **What we're interested in is the probability of a class given an observation** → still will be useful in computing our entropy since this is the probability of a feature belonging to a class
  - Ultimately, we're interested in $P(y = k|x)$
  - To compute this, we'll use **Bayes' rule**:
    - $P(a|b) = \frac{P(a)P(b|a)}{P(b)}$
      - $P(b|a)$ - generative likelihood
      - $P(b)$ - evidence
      - $P(a)$ - prior

        - Our density function gives us something proportional to the probability of a feature belonging to a certain class.
- Bayes' Rule gives us a mechanism to reverse the relationship between knowing
- Using Bayes rule, we can say this is equal to:
    - $P(y=k|x) = \frac{P(y=k)P(x|\mu_k,\sigma_k)}{P(x)}$
- We can compute $P(x)$ by using the **Law of Total Probability**:
    - $P(x) = \sum_{i=1}^{K} P(y=k)P(x|y=k)$
        - Sum the probability of all possible values of Y multiplied by the probability of seeing the observation x given that the class is k.
- So finally, our calculation ends up being:
    - $P(y=k|x) = \frac{P(y=k)P(x|\mu_k,\sigma_k)}{\sum_{i=1}^{K} P(y=i)P(x|\mu_i,\sigma_i)}$

Steps to Calculate Entropy Using Norm PDF

Basic steps are as follows:

1. Compute the priors $P(y=k)$ across the entire dataset.
2. Separate the dataset into subsets according to *class*.
3. For the feature you're computing the entropy for (we'll say feature j), compute the mean and standard deviation of each subset. For class/subset k, we now have the parameters μ_k, σ_k
4. For each observation, compute the probability of belonging to each class according to:
    a. $P(y=k|x_j) = \frac{P(y=k)P(x_j|\mu_k,\sigma_k)}{\sum_{i=1}^{K} P(y=i)P(x_j|\mu_i,\sigma_i)}$
5. For each observation, we'll compute the probability for each class for our overall entropy computation as the average entropy over all observations:
    a. $\mathbb{E} = \frac{1}{N}\sum_{i=1}^{N} H(P(y=1)|x_j), P(y=2|x_j), ..., P(y=K|x_j))$

Example

$$X = \begin{bmatrix} 1.0 & 1.0 \\ 1.0 & 3.0 \\ 2.0 & 2.0 \\ 1.0 & 2.0 \\ 3.0 & 2.0 \\ 2.0 & 2.0 \end{bmatrix}, Y = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Step 1 - Calculate the Priors

Given the data observed, what is the probability of each class? This is **calculating the priors**, which we do FIRST.

- $P(y = 1) = \frac{3}{6}$
- $P(y = 0) = \frac{3}{6}$

Step 2 - Separate Data by Class and Calculate Statistics

From here, we separate our data by their classes and then calculate the μ and σ for each subset created by a certain feature. Here we arbitrarily chose the first feature.

https://www.notion-draw.art/slug

Step 3 - Calculate Probabilities of Belonging in Each Class

To calculate the probability of the first observation $x = [1.0, 1.0]$ appearing, we can perform the following calculations, while remembering the following relationships:

$$P(y=0)=\frac{3}{6}$$
$$P(y=1)=\frac{3}{6}$$
$$\mu_1^{(0)}=2, \sigma_1^{(0)}=1$$
$$\mu_1^{(1)}=1.333, \sigma_1^{(1)}=0.5774$$
$$P(x|\mu,\sigma)\propto p(x|\mu,\sigma)=\frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

- $p(x_1=1|y=0)=\frac{1}{1\sqrt{2\pi}}e^{-\frac{(1-2)^2}{2(1)^2}}=0.2420$ says: what is the probability of being in class 0, generating the feature x_1 equalling 1
- $p(x_1=1|y=1)=\frac{1}{0.5774\sqrt{2\pi}}e^{-\frac{(1-1.333)^2}{2(0.5774)^2}}=0.5849$
- $P(y=0|x_1=1)\propto P(y=0)p(x_1=1|y=0)=0.1210$
- $P(y=1|x_1=1)\propto P(y=1)p(x_1=1|y=1)=0.2925$
- $P(y=0|x_1=1)=\frac{0.1210}{0.1210+0.2925}=0.2926$ divided the numerator by the sum of the numerators here
- $P(y=1|x_1=1)=\frac{0.2926}{0.1210+0.2925}=0.7074$
- So, this observation should appear in the second subset (since the second class' posterior was larger).

Compute Average Entropy

Here we see that the weighted average entropy is relatively high, being ≈ 0.9183

- Doing this over all observations we would get the following subsets:
  - $X=\begin{bmatrix}2.0 & 2.0\\ 3.0 & 2.0\\ 2.0 & 2.0\end{bmatrix}, Y=\begin{bmatrix}1\\0\\0\end{bmatrix}, H=-\frac{1}{3}\log_2\frac{1}{3}-\frac{2}{3}\log_2\frac{2}{3}$ These observations resulted in a higher probability of class 0
  - $X=\begin{bmatrix}1.0 & 1.0\\ 1.0 & 3.0\\ 1.0 & 2.0\end{bmatrix}, Y=\begin{bmatrix}1\\1\\0\end{bmatrix} H=-\frac{1}{3}\log_2\frac{1}{3}-\frac{2}{3}\log_2\frac{2}{3}$ These observations results in a higher probability of class 1
- So, the weighed average entropy is $\mathbb{E}_1=\frac{3}{6}\left(-\frac{1}{3}\log_2\frac{1}{3}-\frac{2}{3}\log_2\frac{2}{3}\right)+\frac{3}{6}\left(-\frac{1}{3}\log_2\frac{1}{3}-\frac{2}{3}\log_2\frac{2}{3}\right)\approx 0.9183$

- We can see that in calculating these probabilities that they didn't lead to the best entropy, telling us that feature 1 doesn't give us good class separation.

<u>NOTE:</u> We can also perform these calculations for the second feature!

- However, we run into issues when calculating its standard deviation since it ends up being 0.
    - To address this, we can add in a numeric stability constant, such as 0.0001 to avoid the σ being 0.
- In setting class features with the $\sigma = 0.1$ and going through all the samples, we would get the following:

$$X = \begin{bmatrix} 2.0 & 2.0 \\ 1.0 & 2.0 \\ 3.0 & 2.0 \\ 2.0 & 2.0 \end{bmatrix}, Y = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, H = -\frac{1}{4} log_2 \frac{1}{4} - \frac{3}{4} \log_2 \frac{3}{4}$$

$$X = \begin{bmatrix} 1.0 & 1.0 \\ 1.0 & 3.0 \end{bmatrix}, Y = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, H = 0$$

- The weighted average class entropy would then be ≈ 0.5409.

$$\mathbb{E}_2 = \frac{4}{6}\left(-\frac{1}{4} log_2 \frac{1}{4} - \frac{3}{4} \log_2 \frac{3}{4}\right) \approx 0.5409$$

Note on Homework #1

Since we're kind of ahead in covering material from next week, we can get started on the first homework assignment to an extent.