IAI, TCG CREST
Avisek Gupta (avisek003@gmail.com)

Machine Learning

12 – Feature Scaling, Distance Metrics, Penalty Norms

October 25, 2022

Data: Instances and Features

For a large number of Machine Learning problems, we assume the existence of a **data matrix** with n rows and d number of columns.

The data matrix can be written as $X \in \mathbb{R}^{n \times d}$, where the rows represent n **data instances** or **samples**, and the columns represent **features**.

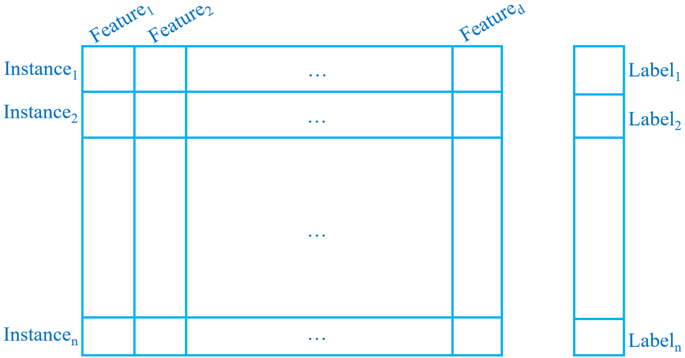
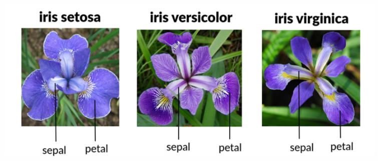


Figure: Data Matrix $X \in \mathbb{R}^{n \times d}$, with an accompanying label vector

Data: Instances and Features

The data matrix can be written as $X \in \mathbb{R}^{n \times d}$, where the rows represent n **data instances** or **samples**, and the columns represent **features**.

E.g.: the **Iris data set** -



	Petal Length	Petal Width	Sepal Length	Sepal Width		
Iris Instance$_1$	5.1	3.5	1.4	0.2	0	Iris Species$_1$
Iris Instance$_2$	4.9	3.0	1.4	0.2	0	Iris Species$_2$
Iris Instance$_3$	4.7	3.2	1.3	0.2	0	Iris Species$_3$
	…				…	

Image modified from: `https://medium.com/@Nivitus./iris-flower-classification-machine-learning-d4e337140fa4`

Data: Instances and Features

- Data Instances:
  - Having more data is generally better.
  - Training ML models for problems where the data is limited is a challenge.
- Features:
  - Collecting more features may seem beneficial, since more information is gathered about a problem. However, more features may lead to lower ML model accuracies.
  - Can some features be easily discarded?

Data: Instances and Features

- Data Instances:
  - Having more data is generally better.
  - Training ML models for problems where the data is limited is a challenge.
- Features:
  - Collecting more features may seem beneficial, since more information is gathered about a problem. However, more features may lead to lower ML model accuracies.
  - Can some features be easily discarded?
    - Constant Features (and features with very low variance)
    - Linearly dependent features

Data: Instances and Features

- Data Instances:
  - Having more data is generally better.
  - Training ML models for problems where the data is limited is a challenge.
- Features:
  - Collecting more features may seem beneficial, since more information is gathered about a problem. However, more features may lead to lower ML model accuracies.
  - Can some features be easily discarded?
    - Constant Features (and features with very low variance)
    - Linearly dependent features
  - Q1. How can we handle features with **differing range of values**?
  - Q2. Can ML methods learn **which features are useful**?

Different ranges of feature values

Let $X \in \mathbb{R}^2$ have two features, x_1 and x_2. Let $x_1 \in [0, 1]$, and $x_2 \in [0, 1000]$.

The squared Euclidean distance between any two data instances is given by:

$$||\mathbf{x}^{(i)} - \mathbf{x}^{(j)}||_2^2 = (x_1^{(i)} - x_1^{(j)})^2 + (x_2^{(i)} - x_2^{(j)})^2$$

The second term in the R.H.S. will dominate the overall measure of distance.

Different ranges of feature values

Let $X \in \mathbb{R}^2$ have two features, x_1 and x_2. Let $x_1 \in [0, 1]$, and $x_2 \in [0, 1000]$.

The squared Euclidean distance between any two data instances is given by:

$$||\mathbf{x}^{(i)} - \mathbf{x}^{(j)}||_2^2 = (x_1^{(i)} - x_1^{(j)})^2 + (x_2^{(i)} - x_2^{(j)})^2$$

The second term in the R.H.S. will dominate the overall measure of distance.

In general, features with higher ranges of values will dominate a distance measure, features with lower ranges in values will be ignored.

Different ranges of feature values

How can features be re-scaled to have similar ranges of values?

Method 1: **Min-Max Standardization**

1. For each feature x_i, find the minimum and maximum values ($x_i^{\min}$ and $x_i^{\max}$)
2. Update every feature component:

$$x_i := \frac{x_i - x_i^{\min}}{x_i^{\max} - x_i^{\min}}$$

By min-max standardization, each feature is rescaled to the range of $[0, 1]$.

Different ranges of feature values

Min-Max Standardization:

Update every feature component:

$$x_i := \frac{x_i - x_i^{\min}}{x_i^{\max} - x_i^{\min}}$$

0.1	1000
1.1	3000
0.6	2000

Different ranges of feature values

Min-Max Standardization:

Update every feature component:

$$x_i := \frac{x_i - x_i^{\min}}{x_i^{\max} - x_i^{\min}}$$

0.1	1000
1.1	3000
0.6	2000

0	0
1	1
0.5	0.5

Different ranges of feature values

How can features be re-scaled to have similar ranges of values?

Method 2: **Mean-Standard-Deviation Normalization**

1. For each feature x_i, find the mean and the standard deviation (μ_i and σ_i)
2. Update every feature component:

$$x_i := \frac{x_i - \mu_i}{\sigma^i}$$

After mean-standard-deviation normalization, each feature is transformed to follow a univariate standard normal distribution.

Different ranges of feature values

Mean-Standard-Deviation Normalization:

Update every feature component:

$$x_i := \frac{x_i - \mu_i}{\sigma^i}$$

0.1	1000
1.1	3000
0.6	2000

Different ranges of feature values

Mean-Standard-Deviation Normalization:

Update every feature component:

$$x_i := \frac{x_i - \mu_i}{\sigma^i}$$

0.1	1000
1.1	3000
0.6	2000

-1.22	-1.22
1.22	1.22
0	0

Different ranges of feature values

Method 1: **Min-Max Standardization**

1. For each feature x_i, find the minimum and maximum values ($x_i^{\min}$ and $x_i^{\max}$)
2. Update every feature component:

$$x_i := \frac{x_i - x_i^{\min}}{x_i^{\max} - x_i^{\min}}$$

Method 2: **Mean-Standard-Deviation Normalization**

1. For each feature x_i, find the mean and the standard deviation (μ_i and σ_i)
2. Update every feature component:

$$x_i := \frac{x_i - \mu_i}{\sigma^i}$$

Different ranges of feature values

Method 1: **Min-Max Standardization**

1. For each feature x_i, find the minimum and maximum values ($x_i^{\min}$ and $x_i^{\max}$)
2. Update every feature component:

$$x_i := \frac{x_i - x_i^{\min}}{x_i^{\max} - x_i^{\min}}$$

Method 2: **Mean-Standard-Deviation Normalization**

1. For each feature x_i, find the mean and the standard deviation (μ_i and σ_i)
2. Update every feature component:

$$x_i := \frac{x_i - \mu_i}{\sigma^i}$$

Which approach is better?

Measures of Dissimilarity: Metric

A metric $d : X \times X \to \mathbb{R}$ is a function that satisfies the following for all $\mathbf{x}, \mathbf{y}, \mathbf{z} \in X$,

1. Non-negativity: $d(\mathbf{x}, \mathbf{y}) \geq 0$, with $d(\mathbf{x}, \mathbf{y}) = 0$ iff $\mathbf{x} = \mathbf{y}$
2. Symmetry: $d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x})$.
3. Triangle Inequality: $d(\mathbf{x}, \mathbf{z}) \geq d(\mathbf{x}, \mathbf{y}) + d(\mathbf{y}, \mathbf{z})$.

Measures of Dissimilarity: Metric

A metric $d : X \times X \to \mathbb{R}$ is a function that satisfies the following for all $\mathbf{x}, \mathbf{y}, \mathbf{z} \in X$,

1. Non-negativity: $d(\mathbf{x}, \mathbf{y}) \geq 0$, with $d(\mathbf{x}, \mathbf{y}) = 0$ iff $\mathbf{x} = \mathbf{y}$
2. Symmetry: $d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x})$.
3. Triangle Inequality: $d(\mathbf{x}, \mathbf{z}) \geq d(\mathbf{x}, \mathbf{y}) + d(\mathbf{y}, \mathbf{z})$.

Some examples of metrics:

- Euclidean distance: $||\mathbf{x} - \mathbf{y}||_2 = \{\sum_{i=1}^{d} (x_i - y_i)^2\}^{1/2}$
- Hamming distance: $||\mathbf{x} - \mathbf{y}||_1 = \sum_{i=1}^{d} |x_i - y_i|$
- Minkowski p-norm: $||\mathbf{x} - \mathbf{y}||_p = \{\sum_{i=1}^{d} |x_i - y_i|^p\}^{1/p}$

Measures of Dissimilarity: Metric

Some examples of metrics:

- Euclidean distance: $||\mathbf{x} - \mathbf{y}||_2 = \{\sum_{i=1}^{d} (x_i - y_i)^2\}^{1/2}$
- Hamming distance: $||\mathbf{x} - \mathbf{y}||_1 = \sum_{i=1}^{d} |x_i - y_i|$
- Minkowski p-norm: $||\mathbf{x} - \mathbf{y}||_p = \{\sum_{i=1}^{d} |x_i - y_i|^p\}^{1/p}$

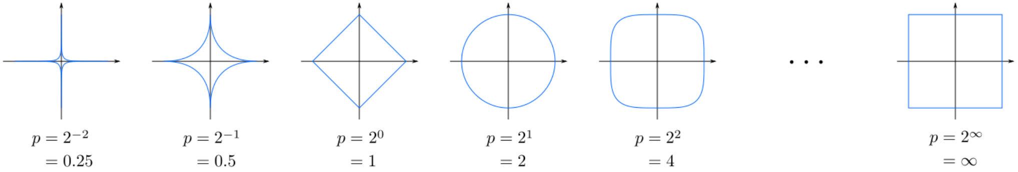


Image Source: `https://en.wikipedia.org/wiki/Minkowski_distance#/media/File:2D_unit_balls.svg`

Measures of Dissimilarity

Some examples of metrics:

- Euclidean distance: $||\mathbf{x}-\mathbf{y}||_2 = \{\sum_{i=1}^{d}(x_i - y_i)^2\}^{1/2}$
- Hamming distance: $||\mathbf{x}-\mathbf{y}||_1 = \sum_{i=1}^{d}|x_i - y_i|$
- Minkowski p-norm: $||\mathbf{x}-\mathbf{y}||_p = \{\sum_{i=1}^{d}|x_i - y_i|^p\}^{1/p}$

Some examples of general measures of similarity / dissimilarity (not metrics):

- Cosine similarity: $S(\mathbf{x},\mathbf{y}) = \dfrac{\mathbf{x}^T\mathbf{y}}{||\mathbf{x}||\,||\mathbf{y}||}$
- KL-divergence: $KL(P||Q) = -\sum_{X} P(\mathbf{x}) \ln \dfrac{Q(\mathbf{x})}{P(\mathbf{x})}$
- Hellinger distance: $H^2(P,Q) = \frac{1}{2}\int_X (\sqrt{p(\mathbf{x})} - \sqrt{q(\mathbf{x})})^2 \lambda(d\mathbf{x})$

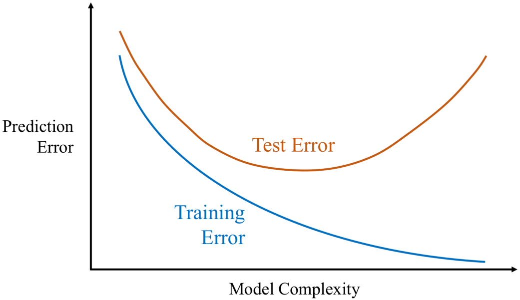
Uses of Measures of Similarities / Dissimilarities

- Differentiate between different data instances

Uses of Measures of Similarities / Dissimilarities

- Differentiate between different data instances
- Use a metric induced norm as a penalty function

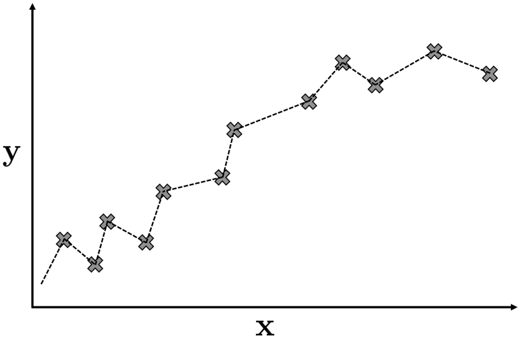
Model Complexity - Accuracy tradeoff



As the model complexity increases, it tends to overfit the data.

Objective - To train a high complexity model, but decrease its tendency to overfit.

Observation - High weights for an overfit model



If we look at the weights:

$w = [2594.67, -18843.27, 73281.03, -165354.85, 217150.0475519, ...]$

The presence of large magnitude weights are indicative of an overfit model.

Penalties in Regression

Ridge Regression: Uses an ℓ_2-norm to not let the model parameters attain large magnitudes.

$$\min_{\mathbf{w}} \sum_{i=1}^{n} (y^{(i)} - \sum_{j=1}^{d} w_j x_j^{(i)} - w_0)^2 + \lambda ||\mathbf{w}||_2^2$$

Penalties in Regression

Ridge Regression: Uses an ℓ_2-norm to not let the model parameters attain large magnitudes.

$$\min_{\mathbf{w}} \sum_{i=1}^{n} (y^{(i)} - \sum_{j=1}^{d} w_j x_j^{(i)} - w_0)^2 + \lambda ||\mathbf{w}||_2^2$$

Lasso Regression: Uses an ℓ_1-norm to drop weights that are close to zero.

$$\min_{\mathbf{w}} \sum_{i=1}^{n} (y^{(i)} - \sum_{j=1}^{d} w_j x_j^{(i)} - w_0)^2 + \lambda ||\mathbf{w}||_1$$

Elastic Net: Penalizes both the ℓ_2 and ℓ_1 norms.

$$\min_{\mathbf{w}} \sum_{i=1}^{n} (y^{(i)} - \sum_{j=1}^{d} w_j x_j^{(i)} - w_0)^2 + \lambda_1 ||\mathbf{w}||_2^2 + \lambda_2 ||\mathbf{w}||_1$$