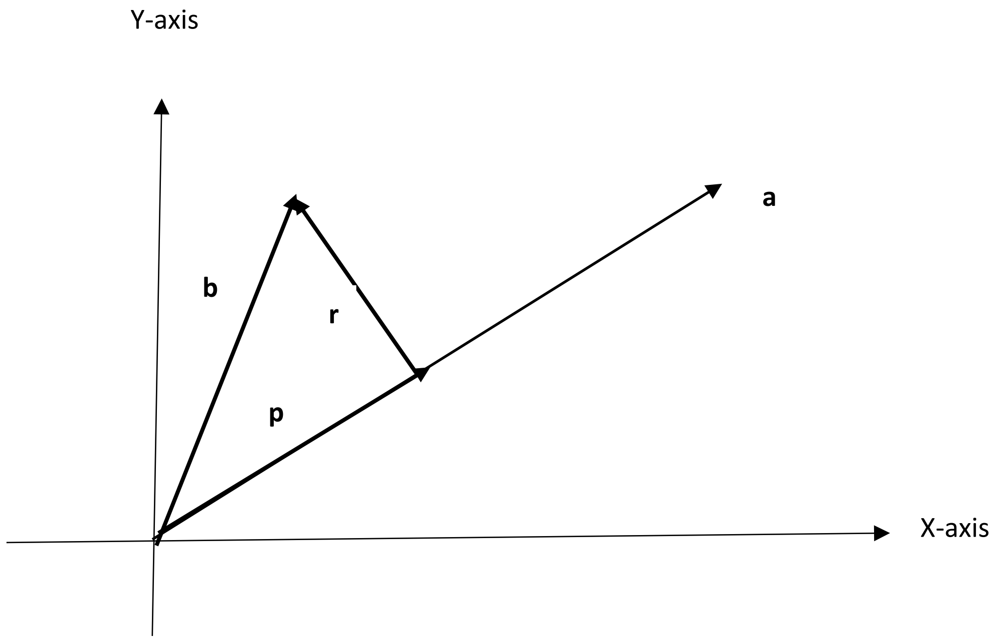
REVIEW OF LINEAR ALGEBRA: Orthogonality

Definition: Two vectors a and b are said to be orthogonal if and only if the dot product, $a^T b = 0$. This also implies the $Cos\ \theta = 0$



Let us consider two vectors $a, b \in \mathbb{R}^m$ of m-dimension in the figure above. We can orthogonally decompose the vector b in the direction of vector a. This is given as $= p + r$. The vector p is called the ***orthogonal projection*** or simply ***projection of b*** on the vector a.

The magnitude of the vector $r = b - p$ gives the orthogonal/perpendicular distance between b and a.

Now let us derive the expression for p by noting that $p = ca$ for some scalar c, as p is parallel to a.

$$p = ca$$

$$r = b - p = b - ca$$

Because p and r are orthogonal

$$p^T r = \ ca^T(b - ca) = ca^T b - c^2 a^T a = 0$$

$$c = \frac{a^T\, b}{a^T a}$$

Therefore, the projection of b on a is given as:

$$p = ca = \left(\frac{a^T\ b}{a^T\ a}\right) a$$

LINEAR ALGEBRA REVIEW: Eigenvalues and eigenvectors

To explain eigenvalues, we first explain eigenvectors. Almost all vectors change direction, when they are multiplied by A. Certain exceptional vectors x are in the same direction as Ax. Those are the "eigenvectors".

This basically means if you multiply an eigenvector by A, and the vector Ax is a number λtimes the original x.

Definition: A scalar λ is called an eigenvalue of the $n \ x \ n$ matrix A is there is a nontrivial solution x of $Ax = \lambda x$. Such an x is called an eigenvector corresponding to the eigenvalue λ. Or simply put the number λ is an eigenvalue of A if and only if $A - \lambda I$ is singular: $\det(A - \lambda I) = 0$

==

ALGORITHM: Linear Discriminant Analysis (LDA)

Given a labeled dataset D of d – dimensional points x_i along with their classes y_i. The goal of LDA is to find a vector w that maximizes the separation between the classes after projection onto w.

Let dataset D consist of n labeled points $\{x_i, y_i\}$ where $x_i \in \mathbb{R}^d$ and $y_i \in \{c_1, c_2, \dots, c_k\}$. Let D_i denote the subset of the points labeled class c_i i.e $D_i = \{x_i | y_i = c_i\}$ and $|D_i| = n_i$.

Let w be a unit vector, that is $w^T w = 1$.

The projection of any d-dimensional point x_i onto the vector w is given by:

$$x`_i = \left(\frac{w^T\, x_i}{w^T\, w}\right) w$$

since $w^T w = 1$,

$x`_i = \left(\frac{w^T\, x_i}{w^T\, w}\right) w = \ (w^T x_i) w = \ a_i w$ where $a_i = \ w^T x_i$.

where a_i specifies the offset or coordinate of x_i along the vector w.

Therefore, the set of n scalars $\{a_1, a_2, \dots, a_n\}$ represents the mapping from $\mathbb{R}^d$ to $\mathbb{R}$, that is, from the original d-dimensional space to a 1-dimensional space (along w).

Assuming we have a dataset of two points, each point coordinate a_i is associated with it the original class label y_i, and thus we can compute, for each of the two classes, the mean of the projected points as follows:

$$m_1 = \frac{1}{n_i} \sum_{x_i = D_1} a_i$$

$$= \frac{1}{n_i} \sum_{x_i = D_1} w^T\, x_i$$

$$= \ w^T \left(\frac{1}{n_i} \sum_{x_i = D_1} x_i \right)$$

$$= \ w^T \mu_1$$

where μ_1 is the mean of all points in D_1.

The similar computations yields $m_2 = \ w^T \mu_2$

This basically means that the mean of the projected points is the same as the projection of the mean.

In order to maximize the separation between the classes, we need two things to happen:

i. maximize the difference between the projected means , $|m_1 - m_2|$.
ii. the variance of the projected points for each class should also not be too large.

LDA maximizes the separation by ensuring that the scatter s_i^2 for the projected points within each class is small, scatter is defined as:

$$s_i^2 = \sum_{x_j = D_i} \left(a_j - m_i\right)^2$$

Scatter is the total squared deviation from the mean, as opposed to the variance, which is the average deviation from mean. This can be rewritten as follows:

$$s_i^2 = n_i \sigma_i^2$$

where $n_i = |D_i|$ is the size, and σ_i^2 is the variance, for class c_i.

Two LDA criteria, maximizing the distance between projected means and minimizing the sum of the projected scatter.

Getting these two criteria and incorporate them into the single maximization criterion, Fisher LDA objective function:

$$max_w J(w) = \frac{(m_1 + m_2)^2}{s_1^2 + s_2^2}$$

Again, the goal of LDA is to find the vector w that maximizes $J(w)$, that is, the direction that maximizes the separation between the two means m_1 and m_2, and minimizes the total scatters $s_1^2 + s_2^2$ of the two classes.

The vector w is also called the optimal linear discriminant (LD).

The optimization objective function above is in the projected space.

We can now rewrite the term $(m_1 + m_2)^2$ of the objective function as follows:

$$(m_1 - m_2)^2 = \left(w^T\,(\mu_1 - \mu_2)\right)^2$$
$$= w^T((\mu_1 - \mu_2)(\mu_1 - \mu_2)^T)w$$
$$= w^T B w$$

where $B = (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T$ is a $d\ x\ d$ rank –one matrix called the ***between class scatter matrix***

Next we can compute the projected scatter for class c_1 as follows:

$$s_1^2 = \sum_{x_i \in D_1} (a_i - m_1)^2$$

$$= \sum_{x_i \in D_1} (w^T x_i - w^T \mu_1)^2$$

$$= \sum_{x_i \in D_1} (w^T (x_i - \mu_1))^2$$

$$= w^T \left(\sum_{x_i \in D_1} (x_i - \mu_1)(x_i - \mu_1)^T \right) w$$

$$= w^T S_1 w$$

where S_1 is the scatter matrix for D_1. Likewise, we can obtain the similar results for class c_2, $s_2^2 = w^T S_2 w$.

Closer look at the scatter matrix, they look like the covariance matrix, but instead of recording the average deviation from the mean, it records the total deviation, that is, $S_i = n_i \Sigma_i$.

Now, we can add the two scatter matrices of our classes to rewrite the denominator of our objective function,

$$s_1^2 + s_2^2 = w^T S_1 w + w^T S_2 w$$

$$= w^T (S_1 + S_2) w = w^T S w$$

where $S = (S_1 + S_2)$ denotes the within –class scatter matrix for the pooled data.

Because both S_1 and S_2 are dxd symmetric positive semidefinite matrices, S has the same properties.

From what we have derived, we can safely rewrite the objective function as follows:

$$max_w J(w) = \frac{(m_1 + m_2)^2}{s_1^2 + s_2^2} = \frac{w^T B w}{w^T S w}$$

To solve for the best direction w, we need to differentiate the objective function with respect to w and set the result to zero.

DIFFERENTIATING THE OBJECTIVE FUNCTION ($\boldsymbol{max_w J(w)}$)

Using the division rule of differentiation:

$$\frac{d}{dw}\left(\frac{f(w)}{g(w)}\right) = \frac{f`(w)g(w) - g`(w)f(w)}{g(w)^2}$$

If we let $(w) = w^T Bw$, then $f`(w) = 2Bw$

While if $g(w) = w^T Sw$, then $g`(w) = 2Sw$

Since $J(w) = h(w) = \frac{f(w)}{g(w)}$, then its derivative can be written as follows:

$$h`(w) = \frac{2Bw(w^T Sw) - 2Sw(w^T Bw)}{(w^T Sw)^2} = 0$$

which yields

$$Bw(w^T Sw) = Sw(w^T Bw)$$

$$Bw = Sw\left(\frac{w^T Bw}{w^T Sw}\right)$$

$$Bw = J(w)Sw$$

$$Bw = \lambda Sw$$

where $\lambda = J(w)$. The derivative above represent a generalized eigenvalue problem where λ is a generalized eigenvalue of B and S; the eigenvalue λ satisfies the equation $\det(B - \lambda S) = 0$.

Because the goal is to maximize the objective function, $J(w) = \lambda$ should be chosen to be the largest generalized eigenvalue, and w to be the corresponding eigenvector.

If S is nonsingular, that is, if S^{-1} exist, then:

$$Bw = \lambda Sw$$

$$(S^{-1}B)w = \lambda w$$

the regular eigenvalue-eigenvector equation. Which basically means if S^{-1} exists, the $\lambda = J(w)$ is an eigenvalue, and w is an eigenvector of the matrix $(S^{-1}B)$.

To maximize $J(w)$ we have to look for the largest eigenvalue λ, and the corresponding dominant eigenvector w specifies the best linear discriminant vector.

==

ALGORITHM: Linear Discriminant Analysis (LDA)

LinearDiscriminant$(D = \{(x_i + y_i)\}_{i=1}^{n})$:

$D_i \leftarrow \{x_i | y_j = c_i, j = 1, \dots, n\}, i = 1,2$ //class-specific subsets

$\mu_i \leftarrow mean(D_i), i = 1,2$ //class means

$B \leftarrow (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T$ //between-class scatter matrix

$Z_i \leftarrow D_i - 1_{n_i}\mu_i^T, i = 1,2$ //center class matrix

$S_i \leftarrow Z_i^T Z_i, i = 1,2$ // class scatter matrix

$S = S_1 + S_2$ // within class scatter matrix

$\lambda_1, w \leftarrow eigen(S^{-1} B)$ //compute dominant eigenvector

---

<u>SPECIAL CASE: THE TWO CLASS SCENARIO</u>

In a case where we have two classes, if S is singular, we can compute the value of w directly without getting to compute the eigenvalues and eigenvectors.

Recall that $B = (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T$ is a dxd rank-one matrix, and thus Bw must point in the same direction as $(\mu_1 - \mu_2)$.

Let us see how it plays out:

$$Bw = \left((\mu_1 - \mu_2)(\mu_1 - \mu_2)^T\right)w$$

$$= (\mu_1 - \mu_2)((\mu_1 - \mu_2)^T w)$$

$$= b(\mu_1 - \mu_2)$$

where $b = (\mu_1 - \mu_2)^T w$ is just a scaler multiplier.

$$Bw = \lambda S w$$

$$b(\mu_1 - \mu_2) = \lambda S w$$

$$w = \frac{b}{\lambda} S^{-1}(\mu_1 - \mu_2)$$

Because $\frac{b}{\lambda}$ is just a scalar, we can solve for the best linear discriminant as:

$$w = S^{-1}(\mu_1 - \mu_2)$$

Once the direction of w has been found we can normalize it to be a unit vector. Thus, instead of solving for the eigenvalue/eigenvector, in the two class case, we immediately can obtain the direction w using $w = S^{-1}(\mu_1 - \mu_2)$.

References

Geron, A., 2017. *Hands-On Machine Learning with Scikit-learn & Tensorflow.* Sepastopol: O`reilly.

James, G., Witten, D., Hastie, T. & Tibshirani, R., 2017. *An Introduction to Statistical Learning.* New York: Springer.

Marsland, S., 2015. *Machine Learning: An Algorithmic Perspective.* s.l.:CRC press.

Raschka, S. & Mirjalili, V., 2017. *Python Machine Learning.* Birmingham: Packt.

Tan, P., Steinbach, M., Karpatne, A. & Kumar, V., 2019. *Introduction to Data Mining.* 2nd ed. New York: Pearson Education.

Zaki, M. J. & Wagner, M., 2014. *Data Mining and Analysis: Fundamental Concepts and Algorithms.* New York: Cambridge University Press.