Least Squares Optimization

The following is a brief review of least squares optimization and constrained optimization techniques, which are widely used to analyze and visualize data. Least squares (LS) optimization problems are those in which the objective (error) function is a quadratic function of the parameter(s) being optimized. The solutions to such problems may be computed analytically using the tools of linear algebra, have a natural and intuitive interpretation in terms of distances in Euclidean geometry, and also have a statistical interpretation (which is not covered here).

We assume the reader is familiar with basic linear algebra, including the Singular Value decomposition (as reviewed in the handout *Geometric Review of Linear Algebra*, `http://www.cns.nyu.edu/∼eero/NOTES/geomLinAlg.pdf`).

1 Regression

Least Squares regression is a form of optimization problem. Suppose you have a set of measurements, y_n (the "dependent" variable) gathered for different known parameter values, x_n (the "independent" or "explanatory" variable). Suppose we believe the measurements are proportional to the parameter values, but are corrupted by some (random) measurement errors, ϵ_n:

$$y_n = px_n + \epsilon_n$$

for some unknown slope p. The LS regression problem aims to find the value of p minimizing the sum of squared errors:

$$\min_p \sum_{n=1}^{N} (y_n - px_n)^2$$

Stated graphically, If we plot the measurements as a function of the explanatory variable values, we are seeking the slope of the line through the origin that best fits the measurements. We can rewrite the error expression in vector form by collecting the $y_n's$ and $x_n's$ in to column vectors ($\vec{y}$ and $\vec{x}$, respectively):

$$\min_p \|\vec{y} - p\vec{x}\|^2$$

or, expressing the squared vector length as an inner product:

$$\min_p (\vec{y} - p\vec{x})^T (\vec{y} - p\vec{x})$$

We'll consider three different ways of obtaining the solution. The traditional approach is to use calculus. If we set the derivative of the error expression with respect to p equal to zero and

• Authors: Eero Simoncelli, Center for Neural Science, and Courant Institute of Mathematical Sciences, and Nathaniel Daw, Center for Neural Science and Department of Psychology.

• Created: 15 February 1999. Last revised: October 7, 2017.

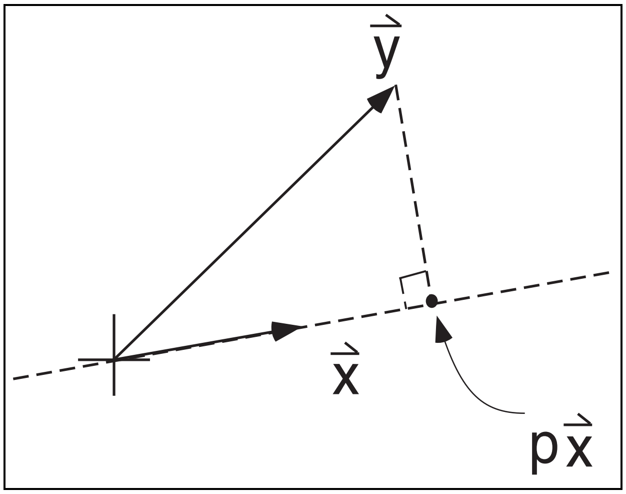
• Send corrections or comments to eero.simoncelli@nyu.edu

• The latest version of this document is available from: `http://www.cns.nyu.edu/∼eero/NOTES/`

solve for p, we obtain an optimal value of

$$p_{\text{opt}} = \frac{\vec{y}^T\vec{x}}{\vec{x}^T\vec{x}}.$$

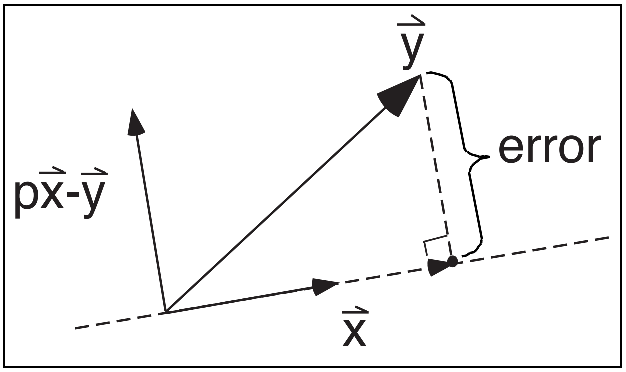
We can verify that this is a minimum (and not a maximum or saddle point) by noting that the error is a quadratic function of p, and that the coefficient of the squared term must be positive since it is equal to a sum of squared values [verify].



A second method of obtaining the solution comes from considering the geometry of the problem in the N-dimensional space of the data vector. We seek a scale factor, p, such that the scaled vector $p\vec{x}$ is as close as possible to $\vec{y}$. From basic linear algebra, we know that the closest scaled vector should be the projection of $\vec{y}$ onto the line in the direction of $\vec{x}$ (as seen in the figure). Defining the unit vector $\hat{x} = \vec{x}/||\vec{x}||$, we can express this as:

$$p_{\text{opt}}\vec{x} = (\vec{y}^T\hat{x})\hat{x} = \frac{\vec{y}^T\vec{x}}{||\vec{x}||^2}\vec{x}$$

which yields the same solution that we obtained using calculus [verify].



A third method of obtaining the solution comes from the so-called **orthogonality principle**. We seek a scaled version $\vec{x}$ that comes closest to $\vec{y}$. Considering the geometry, we can easily see that the optimal choice is vector that lies along the line defined by $\vec{x}$, for which the error vector $\vec{y} - p_{\text{opt}}\vec{x}$ is perpendicular to $\vec{x}$. We can express this directly using linear algebra as:

$$\vec{x}^T(\vec{y} - p_{\text{opt}}\vec{x}) = 0.$$

Solving for p_{opt} gives the same result as above.

Generalization: Multiple explanatory variables

Often we want to fit data with more than one explanatory variable. For example, suppose we believe our data are proportional to a set of known x_n's plus a constant (i.e., we want to fit the data with a line that does not go through the origin). Or we believe the data are best fit by a third-order polynomial (i.e., a sum of powers of the x_n's, with exponents ranging from 0 to 3). These situations may also be handled using LS regression as long as (a) the thing we are fitting to the data is a weighted sum of *known* explanatory variables, and (b) the error is expressed as a sum of squared errors.

Suppose, as in the previous section, we have an N-dimensional data vector, $\vec{y}$. Suppose there are M explanatory variables, and the mth variable is defined by a vector, $\vec{x}_m$, whose elements are the values meant to explain each corresponding element of the data vector. We are looking for weights, p_m, so that the weighted sum of the explanatory variables approximates the data. That is, $\sum_m p_m \vec{x}_m$ should be close to $\vec{y}$. We can express the squared error as:

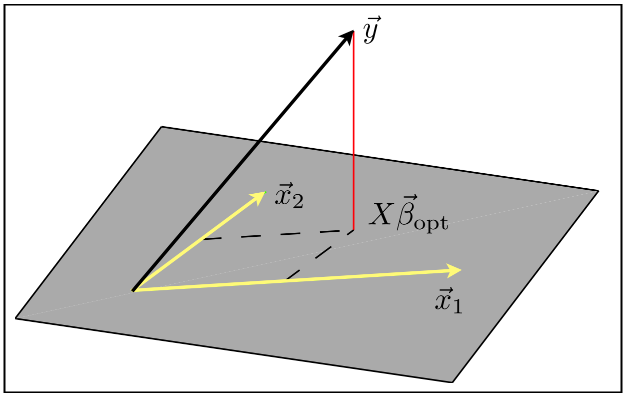
$$\min_{\{p_m\}} \|\vec{y} - \sum_m p_m \vec{x}_m\|^2$$

If we form a matrix X whose columns contain the explanatory vectors, we can write this error more compactly as

$$\min_{\vec{p}} \|\vec{y} - X\vec{p}\|^2$$

For example, if we wanted to include an additive constant (an intercept) in the simple least squares problem shown in the previous section, X would contain a column with the original explanatory variables (x_n) and another column containing all ones.

The vector $X\vec{p}$ is a weighted sum of the explanatory variables, which means it lies somewhere in the subspace spanned by the columns of X. This gives us a geometric interpretation of the regression problem: we seek the vector in that subspace that is as close as possible to the data vector $\vec{y}$. This is illustrated to the right for the case of two explanatory variables (2 columns of X).



As before there are three ways to obtain the solution: using (vector) calculus, using the geometry of projection, or using the orthogonality principle. The orthogonality method is the simplest to understand. The error vector should be perpendicular to the subspace spanned by the explanatory vectors, which is equivalent to saying it must be perpendicular to *each* of the explanatory vectors. This may be expressed directly in terms of the matrix X:

$$X^T \cdot (\vec{y} - X\vec{p}) = \vec{0}$$

Solving for $\vec{p}$ gives:

$$\vec{p}_{\text{opt}} = (X^T X)^{-1} X^T \vec{y}$$

Note that we've assumed that the square matrix $X^T X$ is invertible.

This solution is a bit hard to understand in general, but some intuition comes from considering the case where the columns of the explanatory matrix X are orthogonal to each other. In this case, the matrix $X^T X$ will be diagonal, and the mth diagonal element will be the squared norm of the corresponding explanatory vector, $||\vec{x}_m||^2$. The inverse of this matrix will also be diagonal, with diagonal elements $1/||\vec{x}_m||^2$. The product of this inverse matrix with X^T is thus a matrix whose rows contain the original explanatory variables, each divided by its own squared norm. And finally, each element of the solution, $\vec{p}_{\text{opt}}$, is the inner product of the data vector with the corresponding explanatory variable, divided by its squared norm. Note that this is exactly the same as the solution we obtained for the single-variable problem described above: each $\vec{x}_m$ is rescaled to explain the part of $\vec{y}$ that lies along its own direction, and the solution for each explanatory variable is not affected by the others.

In the more general situation that the columns of X are not orthogonal, the solution is best understood by rewriting the explanatory matrix using the singular value decomposition (SVD), $X = USV^T$, (where U and V are orthogonal, and S is diagonal). The optimization problem is now written as

$$\min_{\vec{p}} \|\vec{y} - USV^T\vec{p}\|^2$$

We can express the error vector in a more useful coordinate system by multipying it by the matrix U^T (note that this matrix is orthogonal and won't change the vector length, and thus will not change the value of the error function):

$$\|\vec{y} - USV^T\vec{p}\|^2 = \|U^T(\vec{y} - USV^T\vec{p})\|^2 = \|U^T\vec{y} - SV^T\vec{p}\|^2$$

where we've used the fact that U^T is the inverse of U (since U is orthogonal).

Now we can define a modified version of the data vector, $\vec{y}^* = U^T\vec{y}$, and a modified version of the parameter vector $\vec{p}^* = V^T\vec{p}$. Since this new parameter vector is related to the original by an orthogonal transformation, we can rewrite our error function and solve the modified problem:

$$\min_{\vec{p}^*} \|\vec{y}^* - S\vec{p}^*\|^2$$

Why is this easier? The matrix S is diagonal, and has M columns. So the mth element of the vector $S\vec{p}^*$ is of the form $S_{mm}p^*_m$, for the first M elements. The remaining $N - M$ elements are zero. The total error is the sum of squared differences between the elements of $\vec{y}$ and the elements of $S\vec{p}^*$, which we can write out as

$$\begin{aligned} E(\vec{p}^*) &= \|\vec{y}^* - S\vec{p}^*\|^2 \\ &= \sum_{m=1}^{M}(y^*_m - S_{mm}p^*_m)^2 + \sum_{m=M+1}^{N}(y^*_m)^2 \end{aligned}$$

Each term of the first sum can be set to zero (its minimum value) by choosing $p^*_m = y_m/S_{mm}$. But the terms in the second sum are unaffected by the choice of $\vec{p}^*$, and thus cannot be eliminated. That is, the sum of the squared values of the last $N - M$ elements of $\vec{y}^*$ is equal to the minimal value of the error.

We can write the solution in matrix form as

$$\vec{p}^*_{\text{opt}} = S^{\#}\vec{y}^*$$

where $S^{\#}$ is a diagonal matrix whose mth diagonal element is $1/S_{mm}$. Note that $S^{\#}$ has to have the same shape as S^T for the equation to make sense. Finally, we must transform our solution back to the original parameter space:

$$\vec{p}_{\text{opt}} = V\vec{p}^*_{\text{opt}} = VS^{\#}\vec{y}^* = VS^{\#}U^T\vec{y}$$

You should be able to verify that this is equivalent to the solution we obtained using the orthogonality principle – $(X^T X)^{-1} X^T \vec{y}$ – by substuting the SVD into the expression.

Generalization: Weighting

Sometimes, the data come with additional information about which points are more reliable. For example, different data points may correspond to averages of different numbers of experimental trials. The regression formulation is easily augmented to include weighting of the data points. Form an $N \times N$ diagonal matrix W with the appropriate error weights in the diagonal entries. Then the problem becomes:

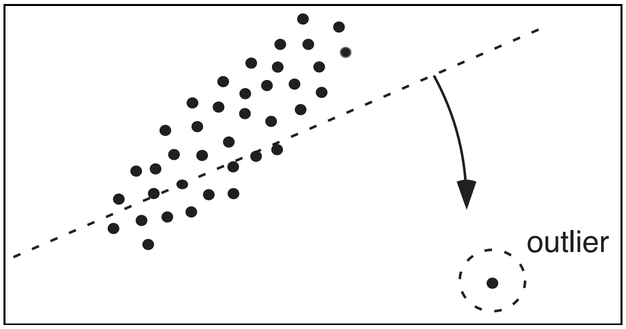
$$\min_{\vec{p}} \|W(\vec{y} - X\vec{p})\|^2$$

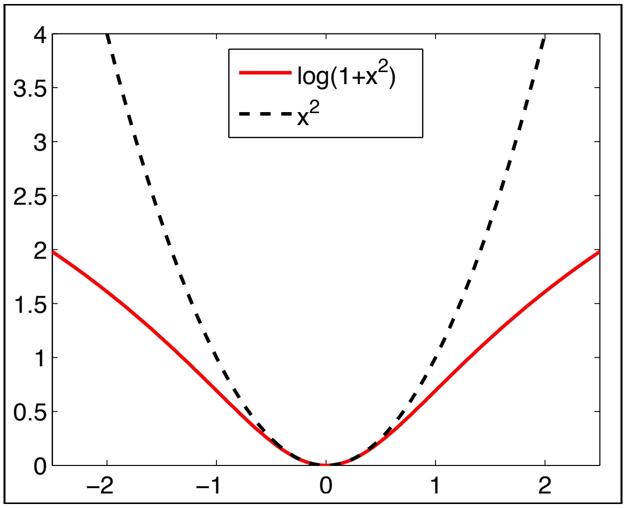
and, using the same methods as described above, the solution is

$$\vec{p}_{\text{opt}} = (X^T W^T W X)^{-1} X^T W^T W \ \vec{y}$$

Generalization: Robustness

A common problem with LS regression is non-robustness to outliers. In particular, if you have one extremely bad data point, it will have a strong influence on the solution. A simple remedy is to iteratively discard the worst-fitting data point, and re-compute the LS fit to the remaining data. This can be done iteratively, until the error stabilizes.





Alternatively one can consider the use of a so-called "robust error metric" $d(\cdot)$ in place of the squared error:

$$\min_{\vec{p}} \sum_n d(y_n - X_n\vec{p}).$$

For example, a common choice is the "Lorentzian" function:

$$d(e_n) = log(1 + (e_n/\sigma)^2),$$

plotted at the right along with the squared error function. The two functions are similar for small errors, but the robust metric assigns a smaller penalty to large errors.

Use of such a function will, in general, mean that we can no longer get an analytic solution to the problem. In most cases, we'll have to use an iterative numerical algorithm (e.g., gradient descent) to search the parameter space for a minimum, and we may get stuck in a local minimum.

2 Total Least Squares (Orthogonal) Regression

In classical least-squares regression, as described in section 1, errors are defined as the squared distance between the data (dependent variable) values and a weighted combination of the independent variables. Sometimes, each measurement is a vector of values, and the goal is to fit a line (or other surface) to a "cloud" of such data points. That is, we are seeking structure within the measured data, rather than an optimal mapping from independent variables to the measured data. In this case, there is no clear distinction between "dependent" and "independent" variables, and it makes more sense to measure errors as the squared *perpendicular* distance to the line.

Suppose one wants to fit N-dimensional data with a subspace of dimensionality $N-1$. That is: in 2D we are looking to fit the data with a line, in 3D a plane, and in higher dimensions an $N-1$ dimensional *hyperplane*. We can express the sum of squared distances of the data to this $N-1$ dimensional space as the sum of squared projections onto a unit vector $\hat{u}$ that is perpendicular to the space. Thus, the optimization problem may thus be expressed as:

$$\min_{\vec{u}} \|M\vec{u}\|^2, \qquad \text{s.t.} \quad \|\vec{u}\|^2 = 1,$$

where M is a matrix containing the data vectors in its rows.

Performing a Singular Value Decomposition (SVD) on the matrix M allows us to find the solution more easily. In particular, let $M = USV^T$, with U and V orthogonal, and S diagonal with positive decreasing elements. Then

$$\begin{aligned} \|M\vec{u}\|^2 &= \vec{u}^T M^T M \vec{u} \\ &= \vec{u}^T V S^T U^T U S V^T \vec{u} \\ &= \vec{u}^T V S^T S V^T \vec{u}, \end{aligned}$$

Since V is an orthogonal matrix, we can modify the minimization problem by substituting the vector $\vec{v} = V^T\vec{u}$, which has the same length as $\vec{u}$:

$$\min_{\vec{v}} \vec{v}^T S^T S \vec{v}, \qquad \text{s.t.} \quad \|\vec{v}\| = 1.$$

The matrix S^TS is square and diagonal, with diagonal entries s_n^2. Because of this, the expression being minimized is a weighted sum of the components of $\vec{v}$ which must be greater than the square of the smallest (last) singular value, s_N:

$$\begin{aligned} \vec{v}^T S^T S \vec{v} &= \sum_n s_n^2 v_n^2 \\ &\geq \sum_n s_N^2 v_n^2 \\ &= s_N^2 \sum_n v_n^2 \\ &= s_N^2 \|\vec{v}\|^2 \\ &= s_N^2. \end{aligned}$$

where we have used the constraint that $\vec{v}$ is a unit vector in the last step. Furthermore, the expression becomes an equality when $\vec{v}_{\text{opt}} = \hat{e}_N = [0\ 0\ \cdots\ 0\ 1]^T$, the unit vector associated with the Nth axis [verify].

We can transform this solution back to the original coordinate system to get a solution for $\vec{u}$:

$$\begin{aligned} \vec{u}_{\text{opt}} &= V\vec{v}_{\text{opt}} \\ &= V\hat{e}_N \\ &= \vec{v}_N, \end{aligned}$$

which is the Nth column of the matrix V. In summary, the minimum value of the expression occurs when we set $\vec{v}$ equal to the column of V associated with the minimal singular value.

Suppose we wanted to fit the data with a line/plane/hyperplane of dimension $N-2$? We could first find the direction along which the data vary least, project the data into the remaining $(N-1)$-dimensional space, and then repeat the process. But because V is an orthogonal matrix, the secondary solution will be the second-to-last column of V (i.e., the column associated with the second-smallest singular value). In general, the columns of V provide a basis for the data space, in which the axes are ordered according to the sum of squares along each of their directions. We can solve for a vector subspace of any desired dimensionality that best fits the data (see next section).

The total least squares problem may also be formulated as a pure (unconstrained) optimization problem using a form known as the **Rayleigh Quotient**:

$$\min_{\vec{u}} \frac{||M\vec{u}||^2}{||\vec{u}||^2}.$$

The length of the vector $\vec{u}$ doesn't change the value of the fraction, but by convention, one typically solves for a unit vector. As above, this fraction takes on values in the range $[s_N^2, s_1^2]$, and is equal to the minimum value when $\vec{u}$ is set equal to the last column of the matrix V.

Relationship to Eigenvector Analysis

The Total Least Squares and Principal Components problems are often stated in terms of **eigenvectors**. The eigenvectors of a square matrix, A, are a set of vectors that the matrix re-scales:

$$A\vec{v} = \lambda\vec{v}.$$

The scalar λ is known as the **eigenvalue** associated with $\vec{v}$. Any symmetric real matrix can be factorized as:

$$A = V\Lambda V^T,$$

where V is a matrix whose columns are a set of orthonormal eigenvectors of A, and Λ is a diagonal matrix containing the associated eigenvalues. This looks similar in form to the SVD, but it is not as general: A must be square and symmetric, and the first and last orthogonal matrices are transposes of each other.

The problems we've been considering can be restated in terms of eigenvectors by noting a simple relationship between the SVD of M and the eigenvector decomposition of M^TM. The total least squares problems all involve minimizing expressions

$$\|M\vec{v}\|^2 = \vec{v}^T M^T M\vec{v}$$

Substituting the SVD ($M = USV^T$) gives:

$$\vec{v}^T VS^TU^TUSV^T\vec{v} = \vec{v}(VS^TSV^T\vec{v})$$

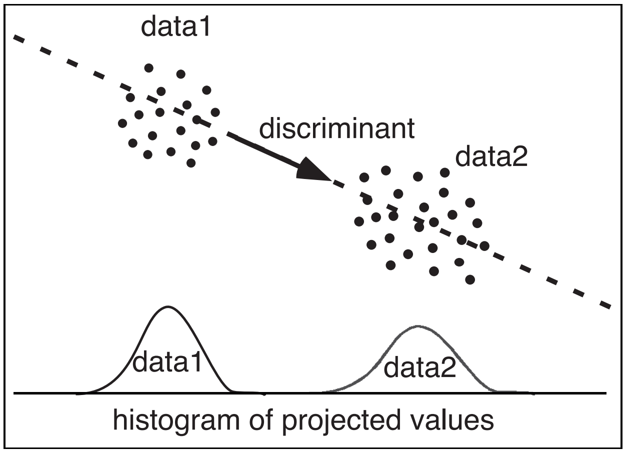
Consider the parenthesized expression. When $\vec{v} = \vec{v}_n$, the nth column of V, this becomes

$$M^TM\;\vec{v}_n = (VS^TSV^T)\;\vec{v}_n = Vs_n^2\vec{e}_n = s_n^2\vec{v}_n,$$

where $\vec{e}_n$ is the nth standard basis vector. That is, the $\vec{v}_n$ are eigenvectors of $(M^T M)$, with associated eigenvalues $\lambda_n = s_n^2$. Thus, we can either solve total least squares problems by seeking the eigenvectors and eigenvalues of the symmetric matrix $M^T M$, or through the SVD of the data matrix M.

3 Fisher's Linear Discriminant

Suppose we have two sets of data gathered under different conditions, and we want to identify the characteristics that differentiate these two sets. More generally, we want to find a *classifier* function, that takes a point in the data space and computes a binary value indicating the set to which that point most likely belongs. The most basic form of classifier is a *linear* classifier, that operates by projecting the data onto a line and then making the binary classification decision by comparing the projected value to a *threshold*. This problem may be expressed as a LS optimization problem (the formulation is due to Fisher (1936)).



We seek a vector $\vec{u}$ such that the projection of the data sets maximizes the discriminability of the two sets. Intuitively, we'd like to maximize the distance between the two data sets. But a moment's thought should convince you that the distance should be considered relative to the variability within the two data sets. Thus, an appropriate expression to maximize is the ratio of the squared distance between the means of the classes and the sum of the within-class squared distances:

$$\max_{\vec{u}} \frac{[\vec{u}^T(\bar{a}-\bar{b})]^2}{\frac{1}{M}\sum_m [\vec{u}^T\vec{a}'_m]^2 + \frac{1}{N}\sum_n [\vec{u}^T\vec{b}'_n]^2}$$

where $\{\vec{a}_m, 1 \leq m \leq M\}$ and $\{\vec{b}_n, 1 \leq n \leq N\}$ are the two data sets, $\bar{a}, \bar{b}$ represent the averages (centroids) of each data set, and $\vec{a}'_m = \vec{a}_m - \bar{a}$ and $\vec{b}'_n = \vec{b}_n - \bar{b}$.

Rewriting in matrix form gives:

$$\max_{\vec{u}} \frac{\vec{u}^T[(\bar{a}-\bar{b})(\bar{a}-\bar{b})^T]\vec{u}}{\vec{u}^T[\frac{A^TA}{M} + \frac{B^TB}{N}]\vec{u}}$$

where A and B are matrices containing the $\vec{a}'_m$ and $\vec{b}'_n$ as their rows. This is now a quotient of quadratic forms, and we transform to a standard Rayleigh Quotient by finding the eigenvector matrix associated with the denominator[1]. In particular, since the denominator matrix

[1] It can also be solved directly as a generalized eigenvector problem.

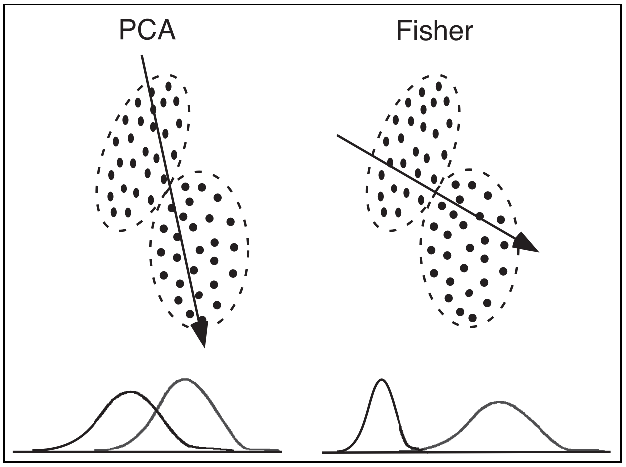
is symmetric, it may be factorized as follows

$$[\frac{A^TA}{M} + \frac{B^TB}{N}] = VD^2V^T$$

where V is orthogonal and contains the eigenvectors of the matrix on the left hand side, and D is diagonal and contains the square roots of the associated eigenvalues. Assuming the eigenvalues are nonzero, we define a new vector relate to $\vec{u}$ by an invertible transformation: $\vec{v} = DV^T\vec{u}$. Then the optimization problem becomes:

$$\max_{\vec{v}} \frac{\vec{v}^T[D^{-1}V^T(\bar{a} - \bar{b})(\bar{a} - \bar{b})^T VD^{-1}]\vec{v}}{\vec{v}^T\vec{v}}$$

The optimal solution for $\vec{v}$ is simply the eigenvector of the numerator matrix with the largest associated eigenvalue.[2] This may then be transformed back to obtain a solution for the optimal $\vec{u}$.



To emphasize the power of this approach, consider the example shown to the right. On the left are the two data sets, along with the first Principal Component of the full data set. Below this are the histograms for the two data sets, as projected onto this first component. On the right are the same two data sets, plotted with Fisher's Linear Discriminant. The bottom right plot makes it clear this provides a much better separation of the two data sets (i.e., the two distributions in the bottom right plot have far less overlap than in the bottom left plot).

[2] In fact, the rank of the numerator matrix is 1 and a solution can be seen by inspection to be $\vec{v} = D^{-1}V^T(\bar{a} - \bar{b})$.