Important Values

l_i number of data points in class i

$l = \sum_{i=1}^{c} l_i$ total amount of data points

$\mu_i = \frac{1}{l_i} \sum_{j=1}^{l_i} x_{ij}$ mean of class i

$\hat{\mu} = \frac{1}{l} \sum_{i=1}^{c} l_i * \mu_i$ mean data point

$\Delta\mathrm{L} = [(\mu_1 - \hat{\mu})|...|(\mu_c - \hat{\mu})]$ centralize the means

$L_i = [(x_{i1} - \mu_i)|...|(x_{il_i} - \mu_i)]$ centralize the data in class i

$L = [L_1|...|L_c]$ **Note:** $rank(L) = l - c$ & $rank(\Delta\mathrm{L}) = c - 1$

Main Values:

$S_B = \Delta\mathrm{L} * \Delta\mathrm{L}^T$ scatter (variance) between the classes

$S_W = L * L^T$ scatter (variance) in the calsses

Finding Projection Vectors

We need to solve $\underset{u}{\operatorname{argmax}} \dfrac{u^T S_B u}{u^T S_W u}$ where u is the projection vector (**Note:** u is an unit vector). This is equivalent to solving $\operatorname{argmax} u^T S_B u$ and $\underset{u}{\operatorname{argmin}} u^T S_W u$. Let assume, we solved $u^T S_W u$ and it is equivalent to κ (**Note:** we don't know the u). Now, we can solve $u^T S_B u$ with a given constraint $u^T S_W u = \kappa$.

Using Lagrange Multiplier:

$\mathcal{L} = u^T S_B u + \lambda(u^T S_W u - \kappa)$

$\dfrac{d\mathcal{L}}{du} = 0 = S_B u + (u^T S_B)^T - \lambda(S_W u + (u^T S_W)^T)$

$\dfrac{d\mathcal{L}}{du} = 0 = 2S_B u - \lambda(2S_W u) \Rightarrow S_B u = \lambda S_W u$, **Note:** S_B & S_W are symetric.

Note: We will find the "optimal solution" because our objective function (the thing we are tring to optimize) and constraint are convex functions which means, we are doing Convex Optimization.

Back to our first optimization problem, since we know $S_B u = \lambda S_W u$, we can re-write it as $max \dfrac{\lambda u^T S_W u}{u^T S_W u}$, since we assume $u^T S_W u = \kappa$, we can subtitute it in and get $max \dfrac{\lambda\kappa}{\kappa} = \lambda$. As we can see, the answer is independent from κ which means, we don't need to find κ to solve the problem. We can see, vector u and λ are the eigen pairs of the $S_W^{-1}S_B$. But there is a problem, S_W is not full rank, so the inverse doesn't exist however, we can do some numerical tricks.

$S_W = U\Sigma V^T$, the SVD decomposition of S_W. Using SVD, we can "find inverse" of S_W, we know that U and V^T inverse exist because they are unitary matrixes (special matrixes). To find the inverse of Σ we do, if $\Sigma(i,i) = 0$ then $\Sigma^{\dagger}(i,i) = 0$ else $\Sigma^{\dagger}(i,i) = \frac{1}{\Sigma(i,i)}$. **Note:** Σ is diagonal matrix (only the diagonal part contains non-zero values). The $\dagger$ symbol means the pseudo inverse (estimating the inverse of a given matrix). Now, we can write pseudo inverse of S_W, $S_W^{\dagger} = V\Sigma^{\dagger}U^T$.

Let we call $W = S_W^{\dagger}S_B$, the top "n" (value n is given by user) biggest eigen values corresponding eigen vectors will be the our projection vector. **Remember** we said $max\frac{u^T S_B u}{u^T S_W u} = \lambda$ and vector u and λ are the eigen pairs of the $S_W^{-1}S_B$. **Note:** We can only pick top $c-1$ eigen vector, because rank(S_B) is $c-1$ (bottleneck), because $c-1 < l-c$, l is much greater than c.