Fast KDAC

February 2, 2017

Given the objective function :

$$
\begin{aligned}
&\min_{W} && -\sum_{i,j} \gamma_{i,j}\, e^{-\frac{\operatorname{tr}\left(W^T A_{i,j} W\right)}{2\sigma^2}} \\
&s.t && W^T W = I \\
& && W \in \mathbb{R}^{d\times q} \\
& && A \in \mathbb{R}^{d\times d} \\
& && \gamma_{i,j} \in \mathbb{R}
\end{aligned} \tag{1}
$$

To optimize this cost function, the original KDAC uses an optimization technique called Dimensional Growth (DG). It rewrites the cost function into separate columns of the W matrix, and solve the problem one column at a time in a Greedy fashion.

$$
\begin{aligned}
&\min_{W} && -\sum_{i,j} \gamma_{i,j}\, e^{-\frac{w_1^T A_{i,j} w_1}{2\sigma^2}} e^{-\frac{w_2^T A_{i,j} w_2}{2\sigma^2}} \ldots e^{-\frac{w_q^T A_{i,j} w_q}{2\sigma^2}} \qquad w_i = i\,\text{th column of } W \\
&s.t && W^T W = I
\end{aligned}
$$

For example, to solve the first column w_1, it ignores the rest of the columns and simplify the problem into :

$$
\begin{aligned}
&\min_{w_1} && -\sum_{i,j} \gamma_{i,j}\, e^{-\frac{w_1^T A_{i,j} w_1}{2\sigma^2}} \\
&s.t && w_1^T w_1 = 1
\end{aligned}
$$

This problem could be solved using standard Gradient methods. Once w_1 is computed, DG then treats the exponential term as a constant $g(w_1)$ and solve for the next variable.

$$
f(w_2) = \; -\sum_{i,j} \gamma_{i,j}\, e^{-\frac{w_1^T A_{i,j} w_1}{2\sigma^2}} e^{-\frac{w_2^T A_{i,j} w_2}{2\sigma^2}}
$$

$$
f(w_2) = \; -\sum_{i,j} \gamma_{i,j}\, g(w_1) e^{-\frac{w_2^T A_{i,j} w_2}{2\sigma^2}}
$$

Without the orthgonality constraint, each stage of the optimization process could be solved using Gradient methods. However, the orthogonality constraint of $W^T W = I$ requires further complication to ensure compliance. To start, the initialization of each new column w_i must go through the Gram Schmit method to ensure its orthogonality against all previous columns. Further more, the gradient direction calculated during each iterations also must undergo Gran Schmit. By removing components from previous vectors, it ensures that each update of w_i maintains feasibility.

Dimension Growth was the original approach used to solve the optimization problem of equation (1), and it achieved its objective of demonstrating the viability of KDAC. However, as the technology approach its next developmental stage, the implementation of this technology on large scale data requires KDAC to adapt for a more implementable algorithm.

The complexity of Dimension Growth algorithm heavily increases time of code development. This issue is especially prominent when speed requirement forces the development to be done in C or on the GPU. In these cases, simpler algorithm using off the shelf techniques could significantly reduce the developmental time and therefore the cost of its implementation.

The convergence speed of Dimension Growth in KDAC is slow. As we know from optimization theory, the convergence rate for gradient methods heavily depend on the conditional value of the Hessian matrix. The conditional value is defined as the ratio between the maximum and the minimum eigenvalue of the Hessian matrix.

$$\text{condition value} = \frac{\text{eig}_{\max}(\nabla^2 f(x))}{\text{eig}_{\min}(\nabla^2 f(x))}$$

The ideal condition value is when the ratio is equal to 1 and the convergence rate slows down very quickly as we increase the condition value beyond 10. Given this fact, it would be instructive to study the Hessian matrix to potentially explain the slow convergence of gradient methods. The Hessian for each column r has the following form :

$$\nabla^2 f(w) = \sum_{i,j} \frac{\gamma_{i,j}}{\sigma^2} e^{-\frac{w_r^T A_{i,j} w_r}{2\sigma^2}} \left[A_{i,j} - \frac{1}{\sigma^2} A_{i,j} w_r w_r^T A_{i,j} \right]$$

From the Hessian matrix, we see that the condition value depends on the summation of matrix $A_{i,j}$ and $A_{i,j}w$ multiplied by some constant term, $\frac{\gamma_{i,j}}{\sigma^2} e^{-\frac{w_r^T A_{i,j} w_r}{2\sigma^2}}$. From this form, it is intuitive to find clues to size of the conditional value from the individual behaviors of $A_{i,j}$ and $A_{i,j}w$.

Due to the complexity of the Hessian form, it may be sufficiently instructive to simply look for an approximation of the Hessian to study its conditional value behavior. Given the problem :

$$f(w) = -\sum_{i,j} \gamma_{i,j} e^{-\frac{w^T A_{i,j} w}{2\sigma^2}} \tag{2}$$

We could approximate the equation (2), by using the Taylor Expansion around 0. Since gradient methods are techniques using the 1st order approximation, we could similarly make a 1st order approximate of the original function.

$$f(w) \approx -\sum_{i,j} \gamma_{i,j} \left(1 - \frac{w^T A_{i,j} w}{2\sigma^2} \right) \tag{3}$$

At this point, we find the approximate Hessian by taking the 2nd derivative.

$$\nabla f(w) \approx \sum \frac{\gamma_{i,j}}{\sigma^2} A_{i,j} w$$

$$\nabla^2 f(w) \approx \sum \frac{\gamma_{i,j}}{\sigma^2} A_{i,j}$$

From this form, we further simplied the Hessian matrix. And the simplified Hessian suggests a dominant influence of the summation of the $A_{i,j}$ matrix with some constant value $\gamma_{i,j}/\sigma^2$.

At this point, let's take a step back and ask how the eigenvalues of $A_{i,j}$ and $A_{i,j} w_r$ influence the conditional value depending on the type of data we handle. We first note that the $A_{i,j}$ matrix is formed with the following equation.

$$A_{i,j} = (x_i - x_j)(x_i - x_j)^T$$

Given a single gaussian cluster of data :

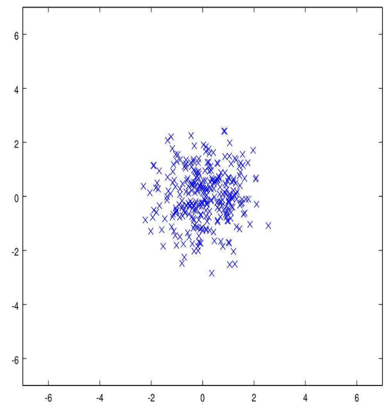

Figure 1.

If we calculate

$$A = \sum A_{i,j}$$

If we randomly generate 100 similar distributions with randomize mean, variance, sample size and dimension, the condition value stays within a small bounded range.

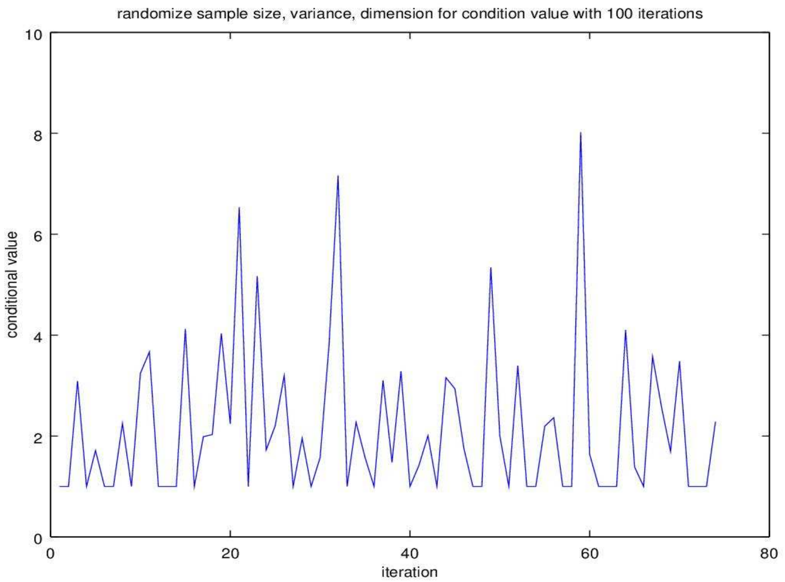

Figure 2.

This plot is generated with the following code :

```
cv = []
for k = 1:100
        n = 20; %floor(40*rand());
        d = floor(5*rand());
        A = zeros(d,d);
        p = floor(5*rand())*randn(n,d);
```

```
        for m = 1:n
                for n = 1:n
                        v = p(m,:) - p(n,:);
                        A = A + v'*v;
                end
        end

        [U,S,V] = svd(A);
        D = diag(S) + 1;

        %max(D)/min(D)
        cv = [cv max(D)/min(D)];

end

plot(cv)
xlabel('iteration')
ylabel('conditional value')
title('randomize sample size, variance, dimension for condition value with 100
iterations')
```

From the plot of the conditional value, we can conclude that the conditional value stays bounded regardless of the sample size, variance, mean and dimensionality. This is for the case of a single cohesive cluster. However, what would happen if there are multiple clusters? As we vary the number of clusters as well as their distance apart, a clear pattern imerges. The following plot shows how the conditional value of 2 and 3 gaussian distributions. As we move them further apart from each other, the condition value explodes very quickly.

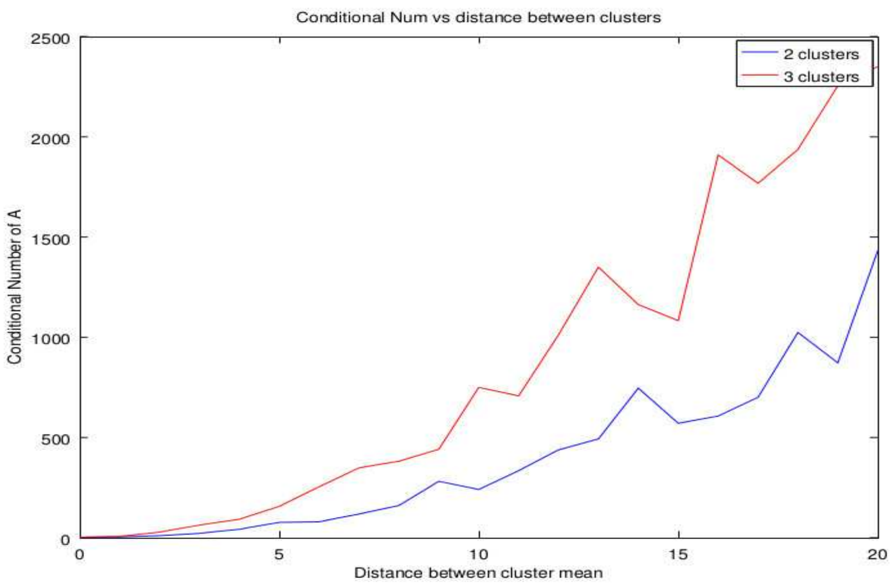

Figure 3.

In conjunction to the plot of a single cluster, this plot suggests that the conditional value is not bounded when the data type are separate into different clusters. As a matter of fact, the more clusters are involved the faster the conditional value grows. The purpose of showing ill-conditionality of this problem provide a reasoning to avoid Gradient method as an optimization approach. This allows us to lead into using a different optimization all together.

1 Fast KDAC

FKDAC is an alternative approach to KDAC that estimates the optimal result without using gradient methods. Similar to DG, we start by looking at a single column of W at a time.

$$\begin{array}{ll} \min & -\sum_{i,j} \gamma_{i,j} e^{-\frac{w^T A_{i,j} w}{2\sigma^2}} \\ W & \\ s.t & w^T w = 1 \\ & W \in \mathbb{R}^{d \times 1} \\ & A \in \mathbb{R}^{d \times d} \\ & \gamma_{i,j} \in \mathbb{R} \end{array} \tag{4}$$

A standard approach to find the optimal solution is to set the derivative of the Lagrangian to zero and solve for w. According to Bertsekas, the first order necessary condition of the Lagrange Multipliers states that :

Proposition 1. *Let x^* be a local minimum of f s.t h(x)=0, and assume that the constraint gradient $\nabla h_1(x^*), \nabla h_2(x^*), \ldots$ are linearly independent. Then there exists a unique vector $\lambda^* = (\lambda_1^*, \lambda_2^*, \ldots,$), called the Lagrange multiplier such that :*

$$\nabla f(x^*) + \sum_{i=1}^{m} \lambda_i^* \nabla h_i(x^*) = 0$$

Notice that the only assumption of the Lagrange multiplier theorem is the linear independence of constraint functions. In our case, since the constraint is :

$$h(w) = w^T w - 1 = 0$$

$$\nabla h(w) = 2 w$$

In the single vector case, the constraint gradient produces only a single vector, and therefore the independence is automatically satisfied. Given that we meet the assumption, we can apply the theorem on our problem and yield :

$$\mathcal{L} = -\sum_{i,j} \gamma_{i,j} e^{-\frac{w^T A_{i,j} w}{2\sigma^2}} + \frac{\lambda}{2} (w^T w - 1)$$

$$\frac{\partial \mathcal{L}}{\partial w} = \left[\sum_{i,j} \frac{\gamma_{i,j}}{\sigma^2} e^{-\frac{w^T A_{i,j} w}{2\sigma^2}} A_{i,j} w \right] + \lambda w = 0$$

Let's for a moment assume that we magically know the optimal solution, w^*, then the problem could be rewritten as :

$$\left[\left[\sum_{i,j} \frac{\gamma_{i,j}}{\sigma^2} e^{-\frac{w^{*T} A_{i,j} w^*}{2\sigma^2}} A_{i,j} \right] + \lambda I \right] w^* = 0$$

If we let :

$$\Phi = \left[\sum_{i,j} \frac{\gamma_{i,j}}{\sigma^2} e^{-\frac{w^{*T} A_{i,j} w^*}{2\sigma^2}} A_{i,j} \right]$$

We can rewrite the equation :

$$[\Phi + \lambda I] w^* = 0$$

In other words, the optimal solution w^* is in the null space of the matrix $\Gamma = [\Phi + \lambda I]$. Or, from another perspective, we could rewrite the equation such that :

$$\Phi w^* = -\lambda w^* \tag{5}$$

From the equation above, w^* is an eigenvector of Φ. From this perspective, it would be extremely easy to find w^* if we know Φ. But, of course, Φ includes the variable w^*. If we already know w^*, there would be no point of finding it again.

To work around the requirement of w^* in Φ, we could approximate the cost function with a simpler cost function.

Proposition 2. *Assuming Lipschitz condition, the problem* $w^* = \underset{w}{\text{argmin}} \quad -\sum_{i,j} \gamma_{i,j}\, e^{-\frac{w^T A_{i,j} w}{2\sigma^2}}$ *can be approximated with the eigenvectors of the marix :* $\sum_{i,j} \frac{\gamma_{i,j}}{\sigma^2} A_{i,j}$ *.*

Proof. □

We start with the Lagrangian again :

$$\mathcal{L} = -\sum_{i,j} \gamma_{i,j}\, e^{-\frac{w^T A_{i,j} w}{2\sigma^2}} + \frac{\lambda}{2}\,(w^T w - 1)$$

Using the Taylor expansion around 0, we approximate $e^{-\frac{w^T A_{i,j} w}{2\sigma^2}}$ up to the first order.

$$f(w) \approx \bar{f}\,(w) = -\sum_{i,j} \gamma_{i,j} \left(1 - \frac{w^T A_{i,j}\, w}{2\,\sigma^2}\right) + \frac{\lambda}{2}\,(w^T w - 1)$$

Given the approximation function of $\bar{f}\,(w)$, we again find the derivate and set it to zero.

$$\nabla \bar{f}\,(w) = \left[\sum_{i,j} \frac{\gamma_{i,j}}{\sigma^2} A_{i,j}\right] w + \lambda\, w = 0$$

$$\nabla \bar{f}\,(w) = \left[\sum_{i,j} \frac{\gamma_{i,j}}{\sigma^2} A_{i,j} + \lambda\, I\right] w = 0$$

$$\left[\sum_{i,j} \frac{\gamma_{i,j}}{\sigma^2} A_{i,j}\right] w = -\lambda\, w$$

$$\Phi\, w = -\lambda\, w$$

Once we have found the initial w_k, we could plug it back into the original gradient equation to find a better approximation of Φ_{k+1}. Using the new Φ_{k+1}, we can once again approximate and find w_{k+1}. This process could be repeated until some satisfactory condition is met.

$$\text{eig}\left[\sum_{i,j} \frac{\gamma_{i,j}}{\sigma^2}\, e^{-\frac{w_k^T A_{i,j} w_k}{2\sigma^2}} A_{i,j}\right] = w_{k+1}$$

From this point, we could approximate the optimal w^* by simply finding the eigenvector of Φ. To extrapolate this idea to $W \in \mathbb{R}^{d \times q}$ instead of $W \in \mathbb{R}^{d \times 1}$, we could simply pick q eigenvectors. Since they are orthogonal with a magnitude of 1, these solutions are automatically within the constraint space.

2 Solving problem with multiple w columns :

We now expand our understanding to solve problems where W is a $d \times q$ matrix instead of a single vector. We now deal with the problem :

$$\begin{array}{l} \min \\ W \end{array} \quad -\sum_{i,j} \gamma_{i,j}\, e^{-\frac{\mathrm{Tr}(W^T A_{i,j} W)}{2\sigma^2}}$$

$$s.t \qquad W^T W = I$$

Writing out the Larangian :

$$\mathcal{L} = -\sum_{i,j} \gamma_{i,j} e^{-\frac{\mathrm{Tr}(W^T A_{i,j} W)}{2\sigma^2}} + \frac{1}{2} \mathrm{Tr}(\Lambda\,(W^T W - I))$$

We also similarly use the Taylor approximation on the exponential term up to the first term and yield :

$$\mathcal{L} \approx -\sum_{i,j} \gamma_{i,j} \left(1 - \frac{\mathrm{Tr}(W^T A_{i,j} W)}{2\,\sigma^2}\right) + \frac{1}{2} \mathrm{Tr}(\Lambda\,(W^T W - I))$$

Following the similar procedure by finding the derivative of $\mathcal{L}$ and setting it to zero.

$$\nabla \mathcal{L} \approx \sum_{i,j} \frac{\gamma_{i,j}}{\sigma^2} A_{i,j} W + W\,\Lambda = 0$$

We arrange the problem into standard eigenvalue/eigenvector problem.

$$\left[\sum_{i,j} \frac{\gamma_{i,j}}{\sigma^2} A_{i,j}\right] W = -W\,\Lambda$$

From this, we see that if $W \in \mathbb{R}^{d \times q}$, the W is q eigenvectors of the matrix :

$$\Phi_0 = \sum_{i,j} \frac{\gamma_{i,j}}{\sigma^2} A_{i,j}$$

And Λ is the eigenvalue matrix.

$$\Lambda = \mathrm{eig}\left[\sum_{i,j} \frac{\gamma_{i,j}}{\sigma^2} A_{i,j}\right]$$

Using the W_0 discovered from this initialization point, we could similarly form an updated Φ_{n+1} using the original cost derivative without any approximation.

$$\Phi_{n+1} = \left[\sum_{i,j} \frac{\gamma_{i,j}}{\sigma^2} e^{-\frac{\mathrm{Tr}(W_n^T A_{i,j} W_n)}{2\sigma^2}} A_{i,j}\right]$$

At this point, it is reasonable to wonder that given so many singular values, which singular value would be the most reasonable to pick? One reasonable approach is to use the greedy algorithm to find q eigenvectors that produces the lowest cost. The following section explores the theoretic motivation for choosing the optimal singular values.

3 Single Column Frank Wolfe Method

Let q be the reduced the dimension and d as the original dimension. We have seen from the previous section that the stationary poins could be found by picking q eigenvectors from the Φ marix. However, since q could be significantly smaller than d, using the appropriate eigenvectors could effect the outcome. Without any prior knowledge, the problems requires the optimal solution to choose from potenial large outcomes of $\left(\begin{array}{c} d \\ q \end{array}\right)$. To simpliy the selection process, this section provides a theoretical argument on how these vectors could be chosen intelligently. To demonstrate the idea, we start by simplifying the problem into a single column w vector. This is without much loss of generality since the idea could be generalized into higher dimensions. Again, assuming that the Lipschitz condition holds, we start by borrowing the idea of Frank Wolfe method by taking the 1st order Taylor Expansion around some w_k.

$$f(w) = -\sum_{i,j} \gamma_{i,j} e^{-\frac{w_k^T A_{i,j} w_k}{2\sigma^2}} + 2\left[\sum_{i,j} \gamma_{i,j} e^{-\frac{w_k^T A_{i,j} w_k}{2\sigma^2}} A_{i,j} w_k\right]^T (w_{k+1} - w_k) \tag{6}$$

The Frank Wolfe method assume that the constraint space is convex. For our case, since our constraint space is not convex, we must assume at least that the constraint space is convex within a certain radius.

$$w \in S \qquad s.t \qquad S \text{ is convex} \qquad \forall \quad \|w - w_k\| \leq \varepsilon$$

Looking at the right side of equation 6, the first term is identical to the current cost at w_k. If we assume that higher order terms are insignificant, we can achieve a lower cost as long as we pick w_{k+1} such that the 2nd term is a negative value. Let's look at the 2nd term more closely.

$$2\left[\sum_{i,j} \gamma_{i,j}\, e^{-\frac{w_k^T A_{i,j} w_k}{2\sigma^2}} A_{i,j}\, w_k\right]^T (w_{k+1} - w_k) \leq 0$$

Note that $A_{i,j}$ are symmetric. We can rewrite this expression :

$$w_k^T\left[\sum_{i,j} \gamma_{i,j}\, e^{-\frac{w_k^T A_{i,j} w_k}{2\sigma^2}} A_{i,j}\right] w_{k+1} \leq w_k^T\left[\sum_{i,j} \gamma_{i,j}\, e^{-\frac{w_k^T A_{i,j} w_k}{2\sigma^2}} A_{i,j}\right] w_k \tag{7}$$

Looking at the eqution above, variously conclusions could be drawn. First, by realizing that the quation :

$$v^T = w_k^T\left[\sum_{i,j} \gamma_{i,j}\, e^{-\frac{w_k^T A_{i,j} w_k}{2\sigma^2}} A_{i,j}\right]$$

is a vector, we could rewrite equation (7) as :

$$v^T w_{k+1} \leq v^T w_k$$

To draw the first conclusion, we note that $v^T w_{k+1}$ is a product of two vectors. To ensure that $v^T w_{k+1}$ is always less than or equal to $v^T w_k$, we simply pick w_{k+1} that minimizes the term $v^T w_{k+1}$.

$$\min_{w_{k+1}} \; v^T w_{k+1}$$

We know from geometry that the product of two vectors is minimized when they are opposite directions of each other. Therefore w_{k+1} is minimized when $w_{k+1} = -v$. From this approach, we can iteratively pick w_{k+1} by setting $w_{k+1} = -v$ to achieve a descending direction. However, this approach encounters the problem of discovering the step length accompanying the descending direction. Since the constraint space is not convex, the step length cannot be discovered by methods of interpolation.

Although solutions to these problems are not immediately obvious, the Frank Wolfe derivation does provide another useful insight. Using Frank Wolfe's method, the problem transforms into a different optimization problem.

$$\min_{w_{k+1}} \; w_k^T\left[\sum_{i,j} \gamma_{i,j}\, e^{-\frac{w_k^T A_{i,j} w_k}{2\sigma^2}} A_{i,j}\right] w_{k+1}$$

We note that although we cannot easily choose w_{k+1} due to the step length, we do know how the term is upper bounded if we re-examine the inequality carefully :

$$w_k^T\left[\sum_{i,j} \gamma_{i,j}\, e^{-\frac{w_k^T A_{i,j} w_k}{2\sigma^2}} A_{i,j}\right] w_{k+1} \leq w_k^T\left[\sum_{i,j} \gamma_{i,j}\, e^{-\frac{w_k^T A_{i,j} w_k}{2\sigma^2}} A_{i,j}\right] w_k \tag{8}$$

By choosing the w_k that minimizes the upper bound, we are simultaneously minimizing the Frank Wolfe method. In other words, when w_k is minimized, $w_{k+1} = w_k$. From this observation, we can again rewrite the objective function.

$$\min_{w_{k+1}} \; f(w) = \min_{w_{k+1}} \; w_{k+1}^T\left[\sum_{i,j} \gamma_{i,j}\, e^{-\frac{w_k^T A_{i,j} w_k}{2\sigma^2}} A_{i,j}\right] w_{k+1}$$

Linking this equation to FKDAC, we notice that they are in exact same form. Except from Frank Wolfe interpretation, not only are we finding any eigenvectors, we are specifically finding the eigenvectors that minimizes quadratic term.

$$\min_{w_{k+1}} f(w) = \min_{w_{k+1}} w_{k+1}^T \Phi_k w_{k+1}$$

This observation tells us that the optimal vectors to solve the FKDAC problem is the least dominant eigenvectors.

4 Frank Wolfe with multiple columns

After understanding the basic concept through the usage of a single column example, we can now expand the same idea to a more general multiple column case. Given the first order Taylor expansion.

$$f(W) = f(W_k) + \nabla f(W_k)^T (W - W_k) \tag{9}$$

Applying this equation to our problem, we get :

$$f(W) = f(w) = -\sum_{i,j} \gamma_{i,j} e^{-\frac{\mathrm{Tr}(W_k^T A_{i,j} W_k)}{2\sigma^2}} + \left[\sum_{i,j} \gamma_{i,j} e^{-\frac{\mathrm{Tr}(W_k^T A_{i,j} W_k)}{2\sigma^2}} \mathrm{Vec}(A_{i,j} W_k)\right]^T \mathrm{Vec}(W - W_k)$$

Note that the notation Vec(.) is the vectorization of a matrix. Similar to the single column case, we only need to concentrate on making the 2nd term negative.

$$\left[\sum_{i,j} \gamma_{i,j} e^{-\frac{\mathrm{Tr}(W_k^T A_{i,j} W_k)}{2\sigma^2}} \mathrm{Vec}(A_{i,j} W_k)\right]^T \mathrm{Vec}(W - W_k) \le 0$$

Let :

$$\beta_{i,j} = \gamma_{i,j} e^{-\frac{\mathrm{Tr}(W_k^T A_{i,j} W_k)}{2\sigma^2}}$$

$$\left[\sum_{i,j} \beta_{i,j} (I \otimes A_{i,j}) \mathrm{Vec}(W_k)\right]^T [\mathrm{Vec}(W) - \mathrm{Vec}(W_k)] \le 0$$

We now rearrange the terms and get :

$$\left[\sum_{i,j} \beta_{i,j} (I \otimes A_{i,j}) \mathrm{Vec}(W_k)\right]^T \mathrm{Vec}(W) \le \left[\sum_{i,j} \beta_{i,j} (I \otimes A_{i,j}) \mathrm{Vec}(W_k)\right]^T \mathrm{Vec}(W_k)$$

$$\mathrm{Vec}(W_k)^T \left[\sum_{i,j} \beta_{i,j} (I \otimes A_{i,j})\right] \mathrm{Vec}(W) \le \mathrm{Vec}(W_k)^T \left[\sum_{i,j} \beta_{i,j} (I \otimes A_{i,j})\right] \mathrm{Vec}(W_k)$$

Using the same logic to lower the upper bound, the W_k that minimizes the upper bound consists of the q least dominant eigenvectors of the matrix Φ.

5 Optimality condition

The Frank Wolfe method corresponds directly with the optimality condition with a convex constrained space. Given a problem of :

$$\min_{x \in X} \quad f(x)$$

We know that given x^* as a local minimum in a convex constrained space of X, the following condition must be satisfied.

$$\nabla f(x^*)^T (x - x^*) \ge 0 \qquad \forall x \in X$$

Given that we have a non-convex constraint space, we must make more strict assumptions. Suppose x^* is a local minimum within a ball defined as $\mathcal{B}(x, \varepsilon) := \{x : \|x - x^*\| < \varepsilon\}$. Assume $\mathcal{B}(x, \varepsilon) \cap X$ is convex and f is convex within $\mathcal{B}(x, \varepsilon) \cap X$, then we state the following as the optimality condition.

$$\nabla f(x^*)^T (x - x^*) \geq 0 \qquad \forall x \in X \quad \text{and} \quad \|x - x^*\| \leq \varepsilon$$

Given a function of :

$$f(w) = -\sum \gamma_{i,j} e^{-\frac{w^T A_{i,j} w}{2\sigma^2}}$$

We create an approximate of the function :

$$f(w) \approx -\sum \gamma_{i,j} \left(1 - \frac{1}{2\sigma^2} w^T A_{i,j} w\right)$$

From the approximation, we find the gradient :

$$\nabla f(w) \approx \sum \frac{\gamma_{i,j}}{\sigma^2} A_{i,j} w$$

Using the approximated gradient, we could now check for the optimality condition.

$$w^{*T} \left[\sum \gamma_{i,j} A_{i,j}\right]^T (w - w^*) \geq 0$$

$$w^{*^T} \left[\sum \gamma_{i,j} A_{i,j}\right]^T w \geq w^* \left[\sum \gamma_{i,j} A_{i,j}\right]^T w^*$$

From the equation above, we see that in order for w^* to be a local minimum, w^* must be chosen such that the left hand side is always larger than the right hand side for any w. This is only possible when w^* is the least dominant eigenvector of $\sum \gamma_{i,j} A_{i,j}$ matrix. From this perspective, we conclude that picking the least dominant eigenvector is a reasonable approximation for the local minimum of the original cost function.