# Fast KDAC

Given the objective function :

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

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.

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

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

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

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 :

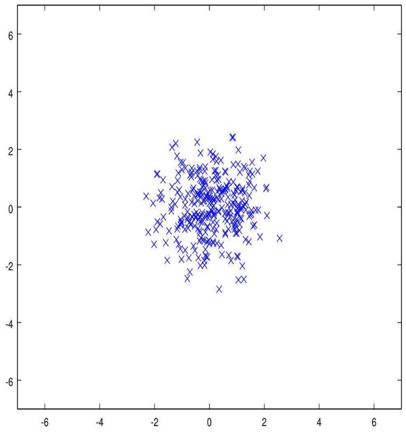

**Figure 1.**

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.

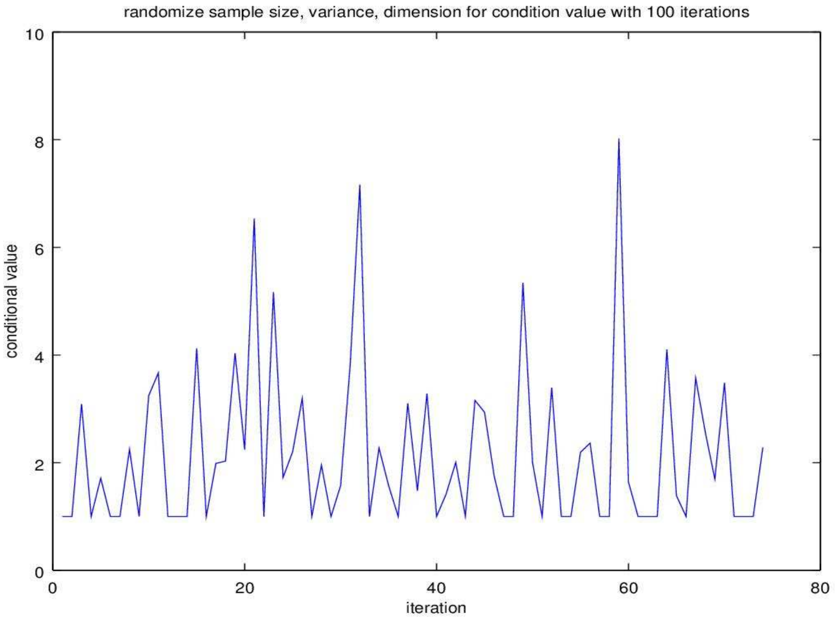

**Figure 2.**

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.

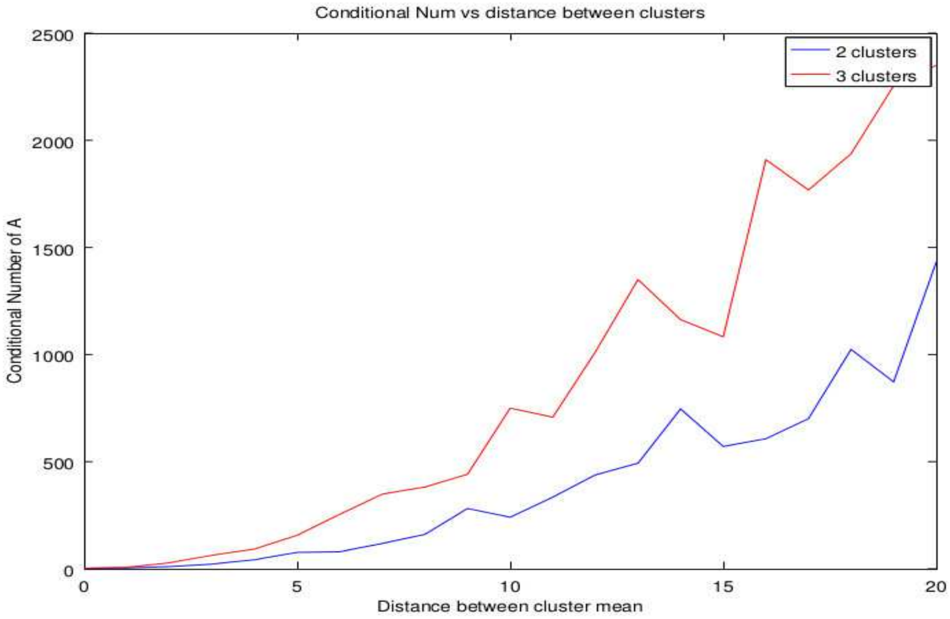

**Figure 3.**

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.

## 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.

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

A standard approach to find the optimal solution is to set the derivative of the Lagrangian to zero and solve for $w$. This could be written as :

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

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

Let's for a moment assume that we magically know the optimal solution, $w^*$, then the problem could also 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 $\Phi$. From this perspective, it would be extremely easy to find $w^*$ if we know $\Phi$. But, of course, $\Phi$ 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 $\Phi$, we could redefine an approximated version of the original cost function. 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.

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

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

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

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

From this point, we could approximate the optimal $w^*$ by simply finding the eigenvector of $\Phi$. 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. Once we have found the initial $w_k$, we could plug it back into the original gradient equation to find a better approximation of $\Phi_{k+1}$. Using the new $\Phi_{k+1}$, we can once again approximate and find $w_{k+1}$.

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

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 to choose the singular value.

## Frank Wolfe Method

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.

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

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 2 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 :

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

The first conclusion, we could draw is that if $v^T w_{k+1}$ is minimized when $w_{k+1} = -v$. From this, approach, we can iteratively pick $w_{k+1}$ by setting $w_{k+1} = -v$. This approach allows us to iterate towards convergence, however, it would be faster to simply approximate the solution at convergence. The second conclusion requires us to realize that at convergence, $w_k = w_{k+1}$. Therefore, we could approximate the solution at convergence by minimizing the left hand side of (7) assuming convergence.

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

From this perspective, we realize that the FKDAC approximation is simultaneously approximating the convergence point of the Frank Wolfe Method. Except in Frank Wolfe, it provides an objective of minimizing $f(w)$. This insight tells us that the best eigenvectors to represent $w^*$ should correspond to the smallest eigenvalue of the matrix :

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

## 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.

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

Given that we have a non-convex constraint space, we must make a more strict assumption that :

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

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.

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

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

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.