M = 20 inducing points

N = 20,000 data points

$O(N^3)$ = 8,000,000,000,000

$O(NM^2) + O(M^3)$ = 8,008,000

So the sparse version is (roughly) 1 million times faster

(Obviously the exact numbers will depend on the hidden constant factors in the big-Oh notation.)

matrix multiplication of an NxK matrix with a KxM matrix results in an NxM matrix with computational complexity O(NMK)

<u>GP - General - O(n^3)</u>

```
common = np.linalg.inv((sigma**2) * np.eye(n_full) + knn)
```
n^3

```
mean = np.dot(np.dot(knn, common), y_full)
n*n  *  n*n   *  n*1 = n*1
```
n*n*1 = **n^2**

```
covariance = knn - np.dot(np.dot(knn, common), knn)
```
n^3

<u>GP - Sparse - O(mn^2)</u>

```
SIGMA = np.linalg.inv(kmm + (sigma**(-2.0)) * np.dot(kmn, knm))
```
nm^2 + m^3

```
MU = (1.0 / sigma)**2 * np.dot(np.dot(np.dot(kmm, SIGMA), kmn), y_full)
m*m  *  m*m  *   m*n   *  n*1   = m*m*m    +   m*m*n   +  m*n*1
```
m^3 + nm^2 + mn

```
A = np.dot(np.dot(kmm, SIGMA), kmm)
m*m  *  m*m  *  m*m  = m*m*m  + m*m*m
```
2m^3

```
B = np.dot(np.dot(kmm_inv, A), kmm_inv)
```

m*m * m*m * m*m = m*m*m   + m*m*m
2m^3

mean_induced = np.dot(np.dot(knm, kmm_inv), MU)
n*m * m*m * m*1 =   n*m*m   + n*m*1
nm^2 + nm
covariance_induced = knn - np.dot(np.dot(knm, kmm_inv), kmn) + np.dot(np.dot(knm, B), kmn)
n*m * m*m * m*n = n*m*m + n*m*n   +
n*m * m*m * m*n = n*m*m + n*m*n
mn^2 + mn^2