EECS 545 Project Progress Report
Sparse Kernel Density Estimates
B.Gopalakrishnan, G.Bellala, G.Devadas, K.Sricharan

1 Introduction

Density estimation forms the backbone for numerous machine learning algorithms like classification, clustering and level set estimation. One of the most common nonparametric methods for density estimation is the Kernel Density Estimate (KDE). The standard KDE assigns equal weights for all the kernels. The number of kernels in the standard KDE is therefore equal to the size of the training data.

A. Motivation

When the training data available is large, the standard KDE becomes intractable for subsequent use. One specific application which suffers from this drawback of standard KDE’s is Flow cytometry data analysis. Flow cytometry data (FCD) is biological data which distinguishes the major categories of leukocytes in blood. It is primarily used for classifying patients by their disorder. The FCD for each patient has up to 10,000 data samples, each of 6 dimensions.

Recently, Carter et.al [1] have developed an algorithm which can be used for low dimensional representation of the collective flow cytometry data of the patients. Their algorithm requires computation of the Kullback Liebler (KL) divergence matrix between the underlying densities corresponding to each patient. Computation of KL divergence estimates using standard KDE’s is extremely time consuming in the context of flow cytometry because of the huge amount of data involved. For $n$ data samples, the order of complexity for computing KL divergence is $O(n^2)$.

We seek to develop KDE’s which are sparse in the weight coefficients. Our motivation behind developing sparse KDE’s is a need for reduced computational complexity while computing KL divergence estimates.

B. Related Work

Sparse weight coefficients have been observed previously in literature while developing kernel density estimates. Schafföner et.al.[2] presented an algorithm for sparse KDE by regression of the empirical cumulative density function. Similarly, Chen et.al.[3] constructed a sparse kernel density estimate using an orthogonal forward regression that incrementally minimizes the leave-one-out test score. Weston et.al.[4] extended the Support vector technique of solving linear operator equations to the problem of density estimation, which induces sparsity by the nature of the SVM. Our work here is primarily motivated by the promising results of Girolami et.al.[5], where the authors considered minimization of the Integrated Squared Error to estimate sparse kernel densities.

C. Contribution

All of the methods described above do not explicitly impose sparsity constraints. These
methods therefore cannot produce kernel density estimates with a specified degree of sparsity. Given the large amount of data involved in real life applications (for example in FCD analysis), it becomes imperative to develop sparser estimates than those provided by the above algorithms (at the cost of reduced quality of the KDE’s).

To this end, we have developed methods that can generate KDE’s with a specified degree of sparsity. The developed methods can also provide sparse solutions while satisfying constraints on the quality of the estimates. These methods therefore provide for choosing a trade-off between the sparsity and the quality of the density estimates.

## 2 Penalized Sparse KDE using ISE

The **Integrated Squared Error (ISE)** between the true density and the estimated density is defined as

\[
\int (f(x) - \sum_{i=1}^{n} \alpha_i k_\sigma(x - x_i))^2 dx
\]  

For gaussian kernels, the ISE reduces to

\[
\alpha^T Q \alpha - c^T \alpha
\]  

where \(\alpha\) is the vector of weights, \(Q_{ij} = k_{\sqrt{2}\sigma}(x_i, x_j), c_i = (2/m) \sum_{j=1}^{m} k_\sigma(y_j, x_i)\), \(x_i\)’s are independent realization of \(f\) used as training data and \(y_i\)’s are independent realizations of \(f\) used as test data.

Girolami et.al. estimate KDE’s by minimizing the ISE.

\[
(P_{ISE}) \min_{\alpha} \alpha^T Q \alpha - c^T \alpha
\]  

Observe that the Gram matrix \(Q\) is Positive Semi-definite. Therefore, the ISE is convex in \(\alpha\)’s. This optimization problem can be solved efficiently using Sequential Minimal Optimization (SMO)[5].

They observed that the weights obtained from minimizing the ISE were sparse. The sparse estimates can be explained by observing that the term \(c^T \alpha\) in the objective function is a convex combination of positive weights. Such a convex combination is minimized by assigning a unit weight to the largest, and setting the rest to zero - a sparse estimate.

In order to increase the sparsity further, we can impose additional penalties on the weight coefficients. The \(l_1\) penalty is a popular choice for inducing sparsity. However, in the problem of Kernel density estimation, the weights are subject to the constraint \[\sum_{i=1}^{n} \alpha_i =\]
1, and therefore the $l_1$ penalty becomes redundant. To circumvent this problem, we consider alternative choices of penalty terms.

The overall objective is to obtain sparse estimates of the density which approximate the true density in some sense. We have developed the following methods.

### 2.1 Weighted $l_1$ penalty

Clearly, the sparsity achieved by Girolami et.al. can be improved by increasing the contribution of the convex term $c^T \alpha$ in the objective function of $P_{ISE}$. The modified objective function then reads as

$$(P_{Wl_1}) \min_\alpha \alpha^T Q \alpha - \lambda c^T \alpha \quad (4)$$

where $\lambda$ is the regularization parameter which controls the extent of sparsity. The objective function in (4) stays convex, and can be solved using SMO.

Observe that $P_{Wl_1}$ can be viewed as a weighted $l_1$ penalized version of $P_{ISE}$, with the penalty term $(\lambda - 1)c^T \alpha$.

### 2.2 Negative $l_2$ penalty

When we constrain the solution to lie on the hyperplane $\sum_{i=1}^n \alpha_i = 1$, observe that a negative $l_2$ penalty will induce sparsity. The objective function obtained when imposing a negative $l_2$ penalty is

$$\min_\alpha \int (f(x) - \sum_{i=1}^n \alpha_i k_\sigma(x - x_i))^2 dx - \lambda \sum_{i=1}^n \alpha_i^2 \quad (5)$$

As earlier, this reduces to

$$(P_{l_2}) \min_\alpha \alpha^T \hat{Q} \alpha - c^T \alpha \quad (6)$$

where $\hat{Q}_{ij} = k_\sqrt{\sigma_i}(x_i, x_j) - \lambda \delta_{ij}$, and $c_i = (2/m) \sum_{j=1}^m k_\sigma(y_j, x_i)$. Observe that for $\lambda = 0$, our problem reduces to that of [5].

Since $\hat{Q}$ is P.S.D, observe that the above objective function is convex for $\lambda = 0$ but starts becoming non-convex as we increase the value of $\lambda$.

**Continuation Search:** In order to deal with this non-convex optimization problem, we employed a continuation search strategy. We initialized the SMO algorithm with equal weights for all kernels. We then iteratively solved the optimization problem for increasing values of $\lambda$. At each iteration, we initialized the SMO algorithm with the weights obtained from the previous iteration. We observed that this produced results such that the sparsity consistently increased as we increased the value of $\lambda$, while the quality of the estimate consistently decreased.
Algorithm 1: Continuation Search to solve $(P_{l_2})$

Step 1: Initialize $\lambda^{(0)} = 0$ and $\alpha_i^{(0)} = 1/n$, $i=1,2,...n$

Step 2: $\alpha_{i}^{(j+1)} = \text{SMO}(\hat{Q}^{(j)}, c, \alpha_i^{(j)})$

Step 3: $\lambda^{(j+1)} = \lambda^{(j)} + \epsilon$, where $\epsilon$ is a small value

Update $\hat{Q}^{(j+1)}$

Step 4: Compute KL Divergence between the standard KDE and the KDE with the weights $\alpha_i^{(j+1)}$

Step 5: Goto Step 2 if KL divergence less than threshold

We have employed a threshold of twice the initial KL divergence between the standard KDE and the KDE with no penalty.

The above algorithm was found to produce good results despite the non-convex nature of the optimization problem.

The SMO reduces the optimization problem to a sequence of lower dimensional optimization problems. We observed that in the lower dimensions, the objective function appeared to be convex for the value of $\lambda$ that was used to produce the sparse estimate. Figure 1 illustrates how the objective function changes from a convex function to a concave function as we increase $\lambda$ from the minimum to the maximum eigenvalue of the Gram matrix $Q$. The objective function corresponding to the $\lambda$ that was used to produce the sparse estimate is shown in red and looks convex. This provides an explanation for the good quality of solutions obtained using the continuation search strategy in conjunction with SMO.

### 2.3 $l_0$ penalty

Given that we are looking for sparse solutions, a common sense approach would be to impose a penalty on the number of non-zero weight coefficients. The $l_0$ penalized objective function is

$$(P_{l_0}) \quad \min_{\alpha} \alpha^T Q \alpha - c^T \alpha + \lambda \|\alpha\|_0$$

(7)

The above objective function is non-convex, and can only be solved using combinatorial search methods, which are intractable given the large dimension of the problem. It is therefore of little practical value in its current form.

Recently, Wakin et. al.[6] have shown that a Weighted $l_1$ minimization problem $\hat{P}_{l_0}$ can be viewed as a relaxed version of the original problem involving the $\|\alpha\|_0$ norm

$$(\hat{P}_{l_0}) \quad \min_{\alpha} \alpha^T Q \alpha - c^T \alpha + \lambda w^T \alpha$$

(8)

where the weights $w_i$ are given by

$$w_i = \begin{cases} \frac{1}{\alpha_i} & \text{if } \alpha_i^* \neq 0; \\ \infty & \text{if } \alpha_i^* = 0. \end{cases}$$
where $\alpha^*$ is the optimal solution to $P_{l_0}$. Further, they have proposed an iterative scheme for obtaining the solution to the relaxed problem. The details of the algorithm are included below

\begin{algorithm}
\caption{Iterative algorithm to solve ($P_{l_0}$)}
\begin{enumerate}
\item [Step 1:] Set iteration count $l$ to zero and $w^{(0)}_i = 1$, $i=1,2,\ldots,n$
\item [Step 2:] Set $\alpha^{(l)}_i = \operatorname{argmin} \alpha^T Q \alpha - c^T \alpha + \lambda w^{(l)}_i \alpha$
\item [Step 3:] Update the weights for each $i=1,2,\ldots,n$,
\[
    w^{(l+1)}_i = \frac{1}{\alpha^{(l)}_i + \epsilon}
\]
\item [Step 4:] Terminate after specified iterations $l_{max}$
\end{enumerate}
\end{algorithm}

where $n$ is the number of kernels. $\epsilon > 0$ is a parameter introduced for stability. In our algorithm, we set $\epsilon = 1/(.1 \times n)$ and $l_{max} = 6$. Note that the optimization problem in Step 2 is convex. SMO was again used to solve this problem.

2.4 Kernel density estimation: A view from kernel feature space

Given i.i.d samples $x_1, x_2, \ldots, x_n \in \mathbb{R}^d$ generated from a multi variate Gaussian distribution $f(x; \theta)$ with unknown mean $\theta$ and covariance matrix $\sigma^2I$, the Maximum likelihood (ML) estimate of $\theta$ is known to be given by $\hat{\theta} = \sum_{i=1}^n x_i / n$. It has been shown by Kim et.
al. [7] that the standard kernel density estimate of \( f \) can be viewed in high dimensional feature space as

\[
\hat{f}(x) = \frac{1}{n} \sum_{i=1}^{n} \langle \Phi(x), \Phi(x_i) \rangle = \left\langle \Phi(x), \frac{1}{n} \sum_{i=1}^{n} \Phi(x_i) \right\rangle
\]

where \( \hat{\theta} = \frac{1}{n} \sum_{i=1}^{n} \Phi(x_i) \) can be considered as the Maximum Likelihood (ML) estimate of \( \theta \).

\[
\hat{\theta} = \arg \max_{\theta} -||\Phi(x) - \theta||^2
\]

\[
= \arg \min_{\theta} ||\Phi(x) - \theta||^2
\]

The above problem can be reformulated as follows

\[
\hat{\alpha} = \arg \min_{\alpha} ||\Phi(x) - \sum_{i=1}^{n} \alpha_i \Phi(x_i)||^2
\]

which reduces to the following quadratic optimization problem

\[
\min_{\alpha} \alpha^T \tilde{Q}\alpha - c^T\alpha
\]

where \( \tilde{Q}_{ij} = k_\sigma(x_i, x_j) \) and \( c_i = (2/m) \sum_{j=1}^{m} k_\sigma(y_j, x_i) \).

The resemblance between this objective function (KFS) and the ISE defined in (1) is striking. Indeed, the KFS can be used in conjunction with the above penalization strategies instead of the ISE to obtain sparse estimates.

Given the similar nature of the ISE and KFS, we chose to restrict our exploration of this alternative measure to developing sparse estimates with the negative \( l_2 \) penalty alone.

3 Results

3.1 Synthetic Data

We applied the different methods developed to obtain sparse KDE’s on one dimensional synthetic data sets. We used the KL divergence between the standard KDE and the sparse KDE as a measure to assess the quality of the sparse KDE’s.

Figure 2 compares the standard KDE and the sparse KDE obtained using \( (P_l) \) for different values of \( \lambda \) for the flare solar data. We can see that the percentage of non-zero coefficients decreases from 88.5% to 3% while the KL divergence increases from 0.37572 to 0.38327.

Figure 3 compares the standard KDE and the sparse KDE obtained using the Kernel Feature Space method for different values of \( \lambda \) for the flare solar data. We can see that the
Figure 2: KDE's on Flare solar data set using ISE with negative $l_2$ penalty

Figure 3: KDE's on Flare solar data set using KFS with negative $l_2$ penalty
<table>
<thead>
<tr>
<th>Method</th>
<th>Dataset</th>
<th>No Penalty</th>
<th>With Penalty</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Sparsity</td>
<td>KL Div</td>
</tr>
<tr>
<td>$(P_{l2})$</td>
<td>Banana</td>
<td>14.5%</td>
<td>0.0518</td>
</tr>
<tr>
<td></td>
<td>Flare Solar</td>
<td>88.5%</td>
<td>0.3757</td>
</tr>
<tr>
<td></td>
<td>Breast Cancer</td>
<td>84.5%</td>
<td>0.3083</td>
</tr>
<tr>
<td>$(P_{Wl1})$</td>
<td>Banana</td>
<td>14.5%</td>
<td>0.0518</td>
</tr>
<tr>
<td></td>
<td>Flare Solar</td>
<td>88.5%</td>
<td>0.3757</td>
</tr>
<tr>
<td></td>
<td>Breast Cancer</td>
<td>84.5%</td>
<td>0.3083</td>
</tr>
<tr>
<td>$(P_{l0})$</td>
<td>Banana</td>
<td>14.5%</td>
<td>0.0518</td>
</tr>
<tr>
<td></td>
<td>Flare Solar</td>
<td>88.5%</td>
<td>0.3757</td>
</tr>
<tr>
<td></td>
<td>Breast Cancer</td>
<td>84.5%</td>
<td>0.3083</td>
</tr>
<tr>
<td>KFS</td>
<td>Banana</td>
<td>79.5%</td>
<td>0.0147</td>
</tr>
<tr>
<td></td>
<td>Flare Solar</td>
<td>95.5%</td>
<td>0.0068</td>
</tr>
<tr>
<td></td>
<td>Breast Cancer</td>
<td>89.5%</td>
<td>0.1240</td>
</tr>
</tbody>
</table>

Table 1: Comparison of methods for Synthetic Data

The percentage of non-zero coefficients decreases from 95.5% to 3% while the KL divergence increases from 0.0068 to 0.38109.

Figure 4 illustrates the sparse density estimates obtained by the different methods for comparable KL divergence. Table 3.1 compares the sparsity and KL divergence with and without penalty for the different methods developed.

It is clear from the results provided that the sparse KDE methods we have developed produce significant improvement in sparsity of the estimates while producing KDE’s of nearly similar quality as the standard KDE.

### 3.2 Flow Cytometry Data

We have data from 20 patients with Mixed Lineage Leukemia (MLL) and 23 patients with Chronic Lymphocytic Leukemia (CLL). The sparse KDE techniques developed above were applied to this collection of Flow Cytometry Data. Table 3.2 lists the sparsity achieved by the different penalty methods in conjunction with the ISE for comparable KL divergence values.

Table 3.3 lists the average figures for the increase in sparsity vs. increase in KL divergence for flow cytometry data, where the average increase in sparsity and KL divergence are computed as

$$SP_{avg} = \frac{1}{N} \sum_{i=1}^{N} \frac{SP_{\text{initial}}^{(i)} - SP_{\text{sparse}}^{(i)}}{SP_{\text{initial}}^{(i)}}$$

$$KL_{avg} = \frac{1}{N} \sum_{i=1}^{N} \frac{KL_{\text{sparse}}^{(i)}}{KL_{\text{initial}}^{(i)}}$$  \hspace{1cm} (9)$$

where $N$ is the total number of patients, $SP_{\text{initial}}^{(i)}$ and $KL_{\text{initial}}^{(i)}$ are the number of non-zero weight coefficients and KL divergence (w.r.t the standard KDE) respectively for the
density estimate of the ith patient with no penalty, and $SP_{\text{sparse}}^{(i)}$ and $KL_{\text{sparse}}^{(i)}$ are the corresponding quantities for the penalized density estimate.

where $N$ is the total number of patients. The FCD of each patient can be viewed as realizations of a density $p_i$ lying on a manifold. In [1], the authors show that the KL divergence between the underlying densities $\{p_i\}$ can be used to approximately compute the Fisher information distance between the densities on the manifolds. Specifically, the fisher distance $D_F(p_1, p_2) \approx \sqrt{D_{KL}(p_1, p_2)}$. We can therefore generate a dissimilarity matrix using the KL divergence between the different patients and subsequently, we can use any Euclidean embedding method to obtain a low dimensional representation of the collective FCD of the patients. We used the classical Multi Dimensional Scaling (cMDS) method to find the low dimensional embedding of the original data. Figure 5 shows the scatter plot of the low dimensional representation of the different patients. The dissimilarity matrix is computed using the standard and sparse KDE’s.

Figure 5 illustrates the MDS plot of the flow cytometry data for the different sparse methods. We can clearly see that the low dimensional embedding obtained using the sparse estimates is comparable to the embedding obtained using the standard KDE.
Table 2: Comparison of methods for FCD of 4 different patients: 2 with CLL and 2 with MLL

<table>
<thead>
<tr>
<th>Method</th>
<th>Dataset</th>
<th>No Penalty</th>
<th>With Penalty</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Sparsity</td>
<td>KL Div</td>
</tr>
<tr>
<td>$(P_l^2)$</td>
<td>CLL1</td>
<td>31%</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>CLL2</td>
<td>24.5%</td>
<td>1.158</td>
</tr>
<tr>
<td></td>
<td>MLL1</td>
<td>14.5%</td>
<td>1.431</td>
</tr>
<tr>
<td></td>
<td>MLL2</td>
<td>37%</td>
<td>0.8411</td>
</tr>
<tr>
<td>$(P_{W1})$</td>
<td>CLL1</td>
<td>31%</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>CLL2</td>
<td>24.5%</td>
<td>1.158</td>
</tr>
<tr>
<td></td>
<td>MLL1</td>
<td>14.5%</td>
<td>1.431</td>
</tr>
<tr>
<td></td>
<td>MLL2</td>
<td>37%</td>
<td>0.8411</td>
</tr>
<tr>
<td>$(P_0)$</td>
<td>CLL1</td>
<td>30%</td>
<td>0.8902</td>
</tr>
<tr>
<td></td>
<td>CLL2</td>
<td>30.5%</td>
<td>1.0486</td>
</tr>
<tr>
<td></td>
<td>MLL1</td>
<td>14%</td>
<td>1.4478</td>
</tr>
<tr>
<td></td>
<td>MLL2</td>
<td>45%</td>
<td>0.7659</td>
</tr>
</tbody>
</table>

Table 3: Comparison of average improvement in sparsity vs quality of estimates for FCD

<table>
<thead>
<tr>
<th>Method</th>
<th>$SP_{avg}(%)$</th>
<th>$KL_{avg}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(P_l^2)$</td>
<td>67.76</td>
<td>1.7654</td>
</tr>
<tr>
<td>$(P_{W1})$</td>
<td>62.41</td>
<td>1.7546</td>
</tr>
<tr>
<td>$(P_0)$</td>
<td>42.97</td>
<td>1.6218</td>
</tr>
</tbody>
</table>

4 Discussion

Of the different methods proposed, the performance of the the negative $l_2$ and $l_0$ penalties were better when compared to the weighted $l_1$ penalty method. The performance of the ISE and the KFS objective functions when used with the negative $l_2$ penalty were quite similar as expected.

We note that the sparsity for the 1D synthetic data is much more than the sparsity achieved in the case of the 6 dimensional FCD for similar quality of estimates. This agrees with our intuition that the representation of signals is easier in lower dimensions.

To conclude, the methods we have developed allow us to specify the trade-off between the sparsity of the estimates and the desired quality (in terms of KL divergence). From the results provided for both Synthetic data and the real life flow cytometry data, we can see that it is possible to obtain extremely sparse KDE’s by allowing for a slight reduction in the quality of the density estimates.

5 Extensions

The methods developed can be extended to other choices of objective functions instead of ISE (such as KL divergence), and other choices of penalty functions (such as $l_p$ norm,
entropy etc.) The optimization problem with a negative $l_2$ penalty using KL divergence is as follows

$$\min_{\mathbf{z}} -\frac{1}{m} \sum_{j=1}^{m} \log(\sum_{i=1}^{n} \alpha_i k_\sigma(y_j, x_i)) - \lambda \sum_{i=1}^{n} \alpha_i^2$$

$$\text{s.t.} \quad \sum_{i=1}^{k} \alpha_i = 1$$

Unlike the ISE or KFS, this objective function is not convex leading to the necessity for an efficient algorithm to solve the optimization problem. Note that the above objective function is strikingly similar to the log likelihood function of a gaussian mixture model, which can be solved efficiently using the popular Expectation Maximization algorithm.

We have formulated the above problem for the special case when $\lambda = 0$, into the form of an EM algorithm. The results we have obtained for this case have been promising. Currently, work is being done to try and solve the penalized version of this problem using the EM algorithm.

**References**


