Kernel Methods : Homework 2

Thibaud Ehret & Sammy Khalife

01/02/2014

1)

The formula for the projection on the i^{th} eigenvector is $\sum_{j=1}^{n} \alpha_j^{(i)}(\Phi(x_j) - m)$.

$$\begin{aligned}
\sum_{j=1}^{n} \alpha_j^{(i)}(\Phi(x_j) - m) &= \sum_{j=1}^{n} \alpha_j^{(i)}\Phi(x_j) - \sum_{j=1}^{n} \alpha_j^{(i)} m \\
&= \sum_{j=1}^{n} \alpha_j^{(i)}\Phi(x_j) - m\left(\sum_{j=1}^{n} \alpha_j^{(i)}\right) \\
&= \sum_{j=1}^{n} \alpha_j^{(i)}\Phi(x_j) - \frac{1}{n}\left(\sum_{u=1}^{n} \Phi(x_u)\right)\left(\sum_{j=1}^{n} \alpha_j^{(i)}\right) \\
&= \sum_{j=1}^{n} \alpha_j^{(i)}\Phi(x_j) - \left(\sum_{u=1}^{n} \frac{1}{n}\left(\sum_{j=1}^{n} \alpha_j^{(i)}\right)\Phi(x_u)\right) \\
&= \sum_{j=1}^{n} \alpha_j^{(i)}\Phi(x_j) - \left(\sum_{j=1}^{n} \frac{1}{n}\left(\sum_{u=1}^{n} \alpha_u^{(i)}\right)\Phi(x_j)\right) \\
&= \sum_{j=1}^{n} \left(\alpha_j^{(i)} - \frac{1}{n}\left(\sum_{u=1}^{n} \alpha_u^{(i)}\right)\right)\Phi(x_j)
\end{aligned}$$

We note $\beta_j = \alpha_j^{(i)} - \frac{1}{n}\left(\sum_{u=1}^n \alpha_u^{(i)}\right)$, so that the vector can be written $\sum_{j=1}^n \beta_j^{(i)}\Phi(x_j)$. Therefore after injecting into the expression of Ψ,

$$
\begin{aligned}
\Psi(x) &= \sum_{i=1}^d \langle \sum_{j=1}^n \beta_j^{(i)}\Phi(x_j), \Phi(x) - m\rangle \left(\sum_{j=1}^n \beta_j^{(i)}\Phi(x_j)\right) + m \\
&= \sum_{i=1}^d \left(\sum_{j=1}^n \beta_j^{(i)}\langle\Phi(x_j), \Phi(x)\rangle\right)\left(\sum_{j=1}^n \beta_j^{(i)}\Phi(x_j)\right) - \sum_{i=1}^d \left(\sum_{j=1}^n \beta_j^{(i)}\langle\Phi(x_j), m\rangle\right)\left(\sum_{j=1}^n \beta_j^{(i)}\Phi(x_j)\right) + m \\
&= \sum_{i=1}^d \left(\sum_{j=1}^n \beta_j^{(i)}K(x_j, x)\right)\left(\sum_{j=1}^n \beta_j^{(i)}\Phi(x_j)\right) - \sum_{i=1}^d \left(\sum_{j=1}^n \beta_j^{(i)}\langle\Phi(x_j), \frac{1}{n}\sum_{u=1}^n \Phi(x_u)\rangle\right)\left(\sum_{j=1}^n \beta_j^{(i)}\Phi(x_j)\right) \\
&\quad + \frac{1}{n}\sum_{u=1}^n \Phi(x_u) \\
&= \sum_{i=1}^d \left(\sum_{j=1}^n \beta_j^{(i)}K(x_j, x)\right)\left(\sum_{j=1}^n \beta_j^{(i)}\Phi(x_j)\right) - \sum_{i=1}^d \left(\sum_{j=1}^n \beta_j^{(i)}\frac{1}{n}\sum_{u=1}^n K(x_j, x_u)\right)\left(\sum_{j=1}^n \beta_j^{(i)}\Phi(x_j)\right) \\
&\quad + \frac{1}{n}\sum_{u=1}^n \Phi(x_u) \\
&= \sum_{i=1}^d \left(\sum_{j=1}^n \beta_j^{(i)}K(x_j, x)\right)\left(\sum_{j=1}^n \beta_j^{(i)}\Phi(x_j)\right) - \sum_{i=1}^d \left(\sum_{j=1}^n \beta_j^{(i)}\frac{1}{n}\sum_{u=1}^n K(x_j, x_u)\right)\left(\sum_{j=1}^n \beta_j^{(i)}\Phi(x_j)\right) \\
&\quad + \frac{1}{n}\sum_{u=1}^n \Phi(x_u) \\
&= \sum_{j=1}^n \left(\sum_{u=1}^n \left(\sum_{i=1}^d \beta_u^{(i)}\beta_j^{(i)}\right) K(x_u, x) - \sum_{u=1}^n \left(\sum_{i=1}^d \beta_u^{(i)}\beta_j^{(i)}\right)\left(\frac{1}{n}\sum_{v=1}^n K(x_u, x_v)\right) + \frac{1}{n}\right)\Phi(x_j)
\end{aligned}
$$

Therefore

$$
\gamma_j = \sum_{u=1}^n \left(\sum_{i=1}^d \beta_u^{(i)}\beta_j^{(i)}\right)\left(K(x_u, x) - \left(\frac{1}{n}\sum_{v=1}^n K(x_u, x_v)\right)\right) + \frac{1}{n}
$$

2)

$$
\begin{aligned}
f(y) &= \|\Phi(y) - \Psi(x)\|^2 \\
&= \langle\Phi(y) - \Psi(x), \Phi(y) - \Psi(x)\rangle \\
&= \langle\Phi(y), \Phi(y)\rangle - 2\langle\Phi(y), \Psi(x)\rangle + \langle\Psi(x), \Psi(x)\rangle \\
&= K(y, y) - 2\langle\Phi(y), \Psi(x)\rangle + \langle\Psi(x), \Psi(x)\rangle
\end{aligned}
$$

Using the the fact that $\Psi(x) = \sum_{i=1}^n \gamma_i \Phi(x_i)$,

$$\begin{aligned} f(y) &= K(y,y) - 2\langle \Phi(y), \sum_{i=1}^n \gamma_i \Phi(x_i\rangle + \langle \sum_{i=1}^n \gamma_i \Phi(x_i), \sum_{i=1}^n \gamma_i \Phi(x_i)\rangle \\ &= K(y,y) - 2\sum_{i=1}^n \gamma_i \langle \Phi(y), \Phi(x_i)\rangle + \sum_{i=1}^n \sum_{j=1}^n \gamma_i \gamma_j \langle \Phi(x_i), \Phi(x_j)\rangle \\ &= K(y,y) - 2\sum_{i=1}^n \gamma_i K(y, x_i) + \sum_{i=1}^n \sum_{j=1}^n \gamma_i \gamma_j K(x_i, x_j) \end{aligned}$$

The idea behind $\Psi(x)$ is to denoise x in the feature space, by keeping only the participation over the d first principal components we can hope to have consider the signal without its noise. Therefore ooptimizing $f(y)$ corresponds to finding the best y in the original space such that it is as close as possible to the "denoised" version in the feature space, therefore the y that optimizes f for a given x correspond to the best denoised version in the original space.

3)

In the case of $K(x,x') = exp(-\frac{||x-x'||^2}{2\sigma^2})$,

$$f(y) = \sum_{i=1}^n \sum_{j=1}^n \gamma_i \gamma_j exp(-\frac{||x_i - x_j'||^2}{2\sigma^2}) - 2\sum_{i=1}^n \gamma_i exp(-\frac{||y-x_i||^2}{2\sigma^2})$$

$$\nabla f(y) = \frac{2}{\sigma^2} \sum_{i=1}^n (y - x_i)\gamma_i exp(-\frac{||y-x_i||^2}{2\sigma^2})$$

We see that a stationary point satisfies :

$$y = \frac{\sum_{i=1}^n x_i \gamma_i exp(-\frac{||y-x_i||^2)}{2\sigma^2})}{\sum_{i=1}^n \gamma_i exp(\frac{-||y-x_i||^2)}{2\sigma^2})}$$

We can apply a fixed point method :

$$y_{k+1} = \frac{\sum_{i=1}^n x_i \gamma_i exp(-\frac{||y_k-x_i||^2)}{2\sigma^2})}{\sum_{i=1}^n \gamma_i exp(\frac{-||y_k-x_i||^2)}{2\sigma^2})}$$

With initialization at different points.

Or we can use a gradient descent method with a Newton step with the Hessian :

$$H_f(y) = \frac{2}{\sigma^2} \sum_{i=1}^n \gamma_i exp(-\frac{||y-x_i||^2}{2\sigma^2})[\frac{Id}{n} - (y-x_i)(y-x_i)^t]$$

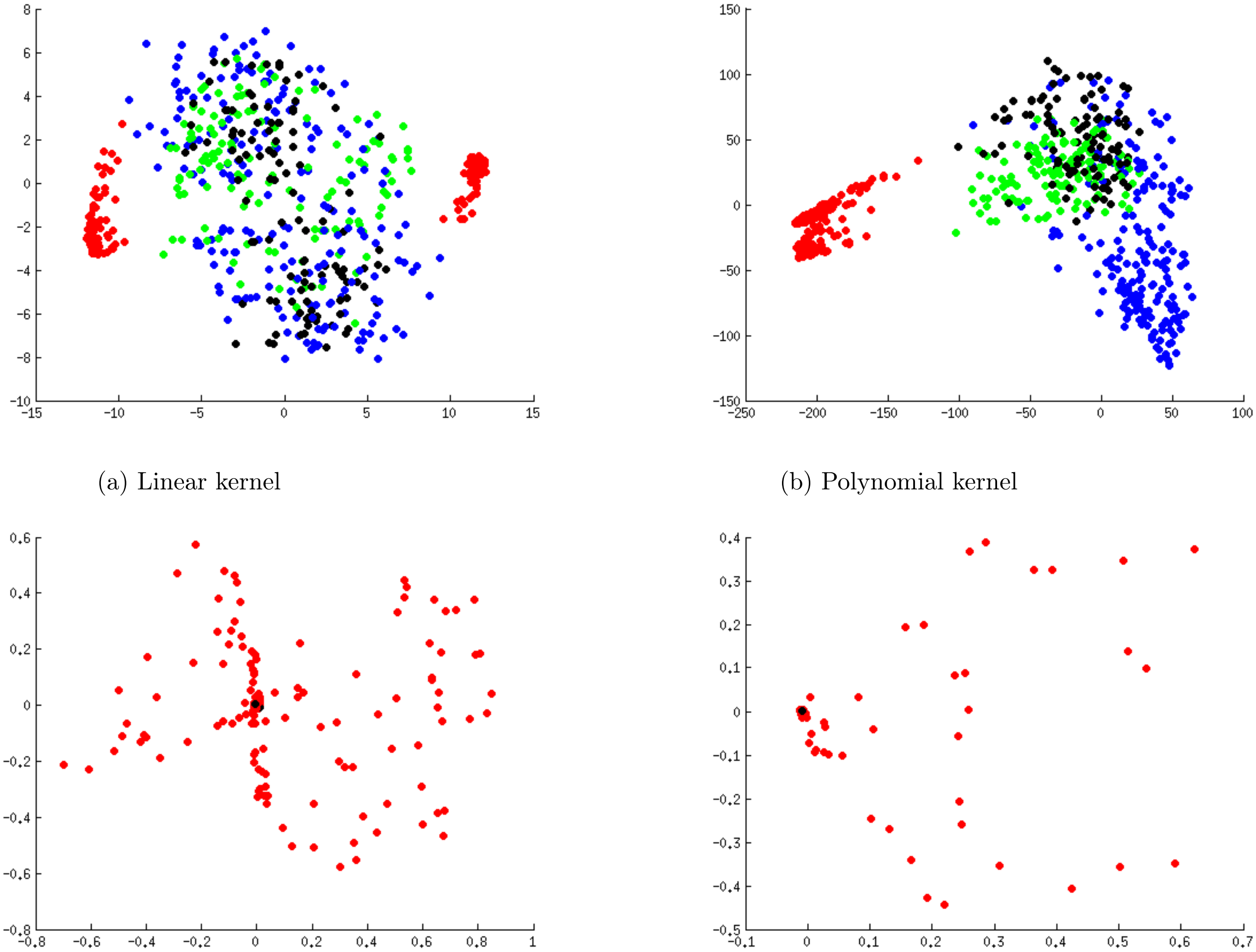
$$y_{k+1} = y_k - H_f^{-1}(y_k)\nabla y_k$$

4)

Data used : (http://statweb.stanford.edu/ tibs/ElemStatLearn/datasets/zip.info.txt)

This dataset is composed of normalized handwritten digits, automatically scanned from envelopes by the U.S. Postal Service. The original scanned digits are binary and of different sizes and orientations; the images here have been deslanted and size normalized, resulting in 16 x 16 grayscale images.

We represent in figures a, b,c and d the projection on the first 2 dimensions for the linear kernel, the polynomial kernel (with $p = 2$) and the gaussian kernel with parameter $\sigma = 0.5$ and $\sigma = 1$ respectively.



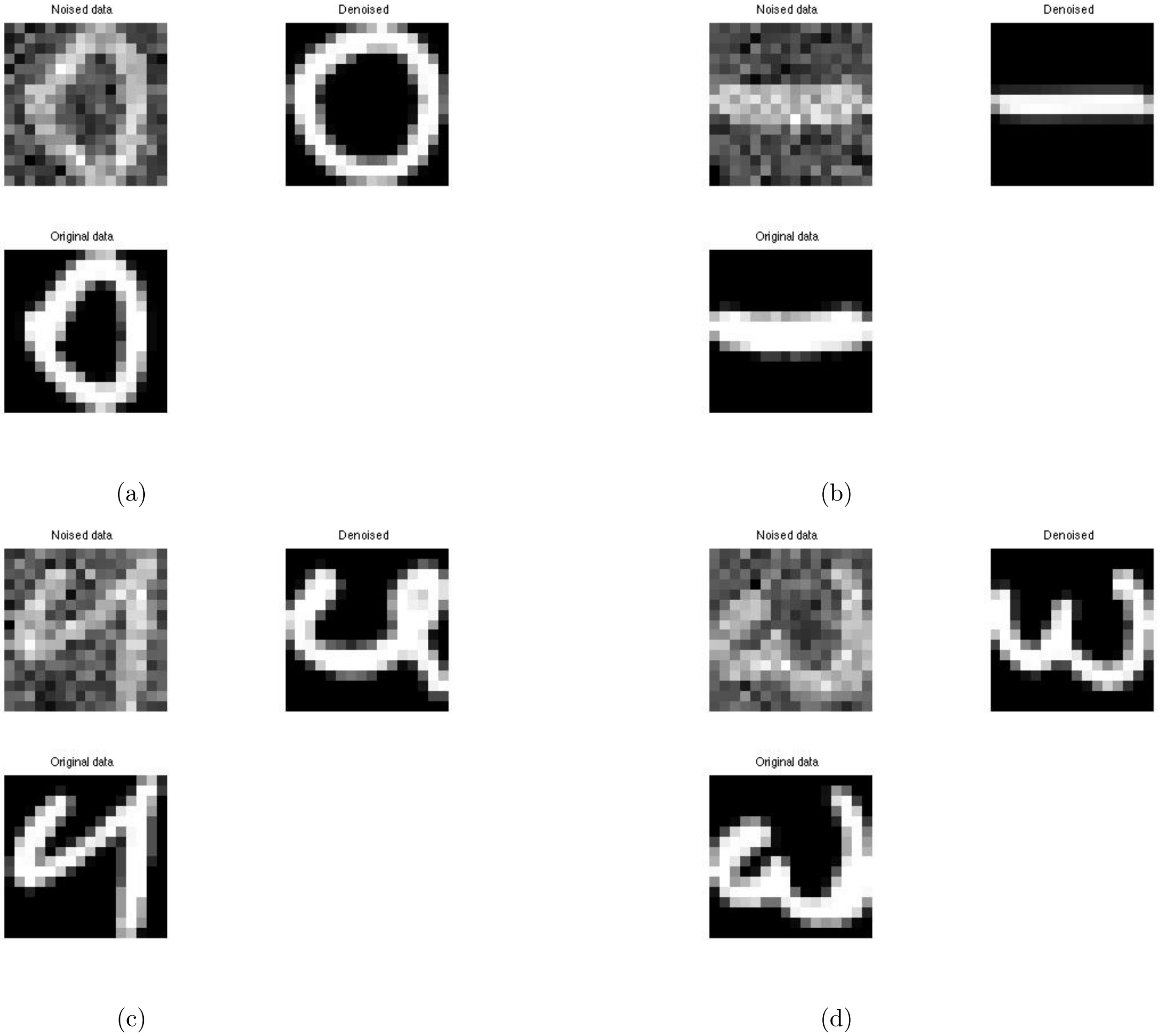
(a) Linear kernel

(b) Polynomial kernel

(c) Gaussian kernel with $\sigma = 0.5$

(d) Gaussian kernel with $\sigma = 1$

Figure 1: Visualization for different kernels, the class of 0 is in blue, the class of 1 is in red, the class of 2 is in green and the class of 3 is in black



(a)

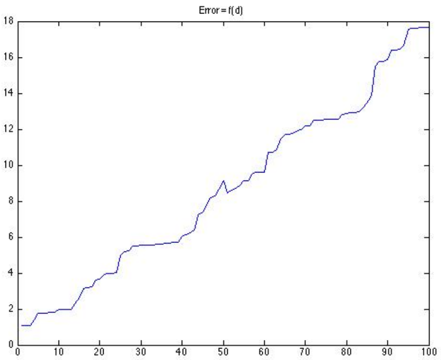
(b)

(c)

(d)

Figure 2: Results on different digits

These results have been obtained with a Gaussian Kernel, with **100 training samples**, and with **d**=**50**. We see that the minimization process that projects the noised data over the vector spanned by the eigenvectors simplify the geometry. Increasing d does not help to achieve better denoising, since we are closer and closer to the initial noised data x. In our implementation example, the error plot is globally increasing with respect to the number of points, with a minimum in $d = 1$. In the general case, one should expect a trade off and a global minimum located strictly after d=1.



(a) Error plot as a function of (d-1)