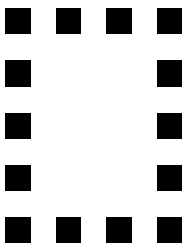
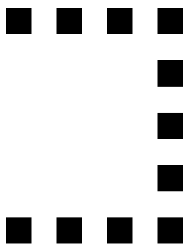
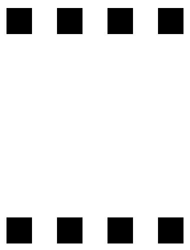
An Application of Orthogonal Projections: Novelty Filter

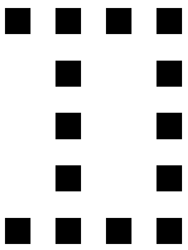
Jen-Mei Chang, Ph.D.
Department of Mathematics and Statistics
California State University, Long Beach

Description

The following 5×4 arrays of black squares can be realized in $\mathbb{R}^{20}$ by associating each entry with a dimension and giving each black square a numerical value of 1 and each empty square a numerical value of 0.

$\mathbf{v}_1 =$  $\mathbf{v}_2 =$  $\mathbf{v}_3 =$ 

and

$\mathbf{y} =$ 

Here, we want to study how different $\mathbf{y}$ is from the set spanned by the $\mathbf{v}_i$'s. This difference can be captured by the residual of the orthogonal projection of $\mathbf{y}$ onto span $\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$. This is a common practice in pattern recognition (e.g., face recognition, object matching, motion detection, etc) where the residual gives us an idea of what feature is novel, something that is not present in the gallery set. This is precisely why the method is called a novelty filter.

Notice that before we can apply the orthogonal projection, we need to make sure the basis is orthogonal. Therefore, we start by finding an **orthogonal basis** for $W = \text{span}\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$, call it $\{\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3\}$. Then find the orthogonal projection of $\mathbf{y}$ onto each $W_i = \text{span}\{\mathbf{u}_i\}$ for each i, denoted by $\hat{\mathbf{y}}_i$. The result of the projections is shown in Figure 1.

Further notice that $\hat{\mathbf{y}}_i = \dfrac{\mathbf{y}^T\mathbf{u}_i}{\mathbf{u}_i^T\mathbf{u}_i}\mathbf{u}_i$ for each $i = 1, 2, 3$, and $\hat{\mathbf{y}} = \hat{\mathbf{y}}_1 + \hat{\mathbf{y}}_2 + \hat{\mathbf{y}}_3$. We can then write $\mathbf{y}$ as a sum of orthogonal vectors, one in W and one in the orthogonal complement of W, $W^\perp$, i.e., $\mathbf{y} = \hat{\mathbf{y}} + \mathbf{z}$, where $\hat{\mathbf{y}} \in W$ and $\mathbf{z} \in W^\perp$. See Figure 2 for an illustration. Can you see that the three black squares in the second column of $\mathbf{y}$ is what $\mathbf{y}$ is novel from W. That is, one can never obtain those three black squares from linear combinations of the $\mathbf{u}_i$'s.

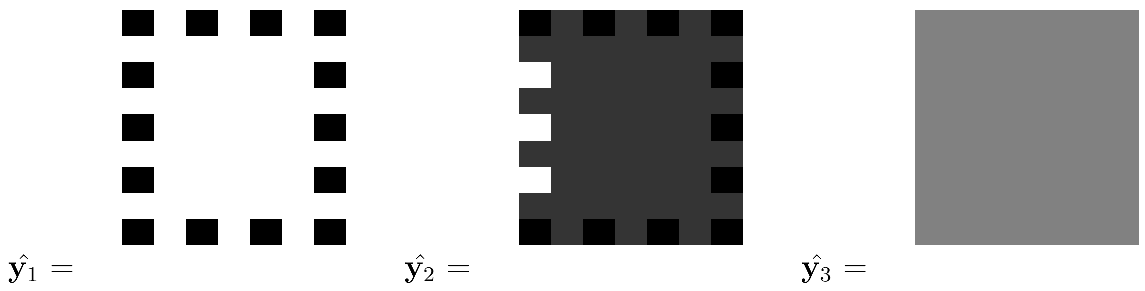


Figure 1: Orthogonal projections of $\mathbf{y}$ onto the subspace $W = \text{span}\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$ expressed separately in each direction.

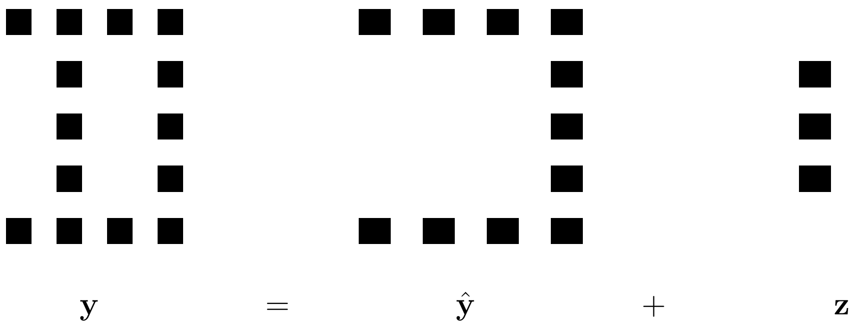


Figure 2: $\mathbf{y}$ written as a sum of two orthogonal vectors $\hat{\mathbf{y}}$ and $\mathbf{z}$, where $\hat{\mathbf{y}} \in W$ and $\mathbf{z} \in W^{\perp}$.

Sample MATLAB Code

Here is a sample MATLAB code used to produce the results above.

```
%% black square = 1, blank square = 0

%% gallery
v1 = [1,1,1,1,1,1,0,0,0,1,1,0,0,0,1,1,1,1,1,1]';
v2 = [1,0,0,0,1,1,0,0,0,1,1,0,0,0,1,1,1,1,1,1]';
v3 = [1,0,0,0,1,1,0,0,0,1,1,0,0,0,1,1,0,0,0,1]';

%% probe
y = [1,0,0,0,1,1,1,1,1,1,1,0,0,0,1,1,1,1,1,1]';

%% define basis vectors for the training subspace
V = [v1 v2 v3];
[Q,R] = qr(V,0);
u1 = Q(:,1); u2 = Q(:,2); u3 = Q(:,3);

%% orthogonal projection onto each direction
y1_hat = ((y'*u1)/(u1'*u1)).*u1;
y2_hat = ((y'*u2)/(u2'*u2)).*u2;
y3_hat = ((y'*u3)/(u3'*u3)).*u3;
```

```
y_hat = Q*Q'*y;
z = y - y_hat;

%% graph the results:
I = ones(5,4);

temp1 = reshape(y1_hat,5,4); temp1 = I - temp1;
figure, imagesc(temp1), colormap(gray), axis off
temp2 = reshape(y2_hat,5,4); temp2 = I - temp2;
figure, imagesc(temp2), colormap(gray), axis off
temp3 = reshape(y3_hat,5,4); temp3 = I - temp3;
figure, imagesc(temp3), colormap(gray), axis off
y_hat = reshape(y_hat,5,4); y_hat = I - y_hat;
figure, imagesc(y_hat), colormap(gray), axis off
z = reshape(z,5,4); z = I - z;
figure, imagesc(z), colormap(gray), axis off
```