Ongoing Attempts

Hossein Goli

April 2024

1 Introduction

In this section, we present a new certification rule that applies a static superpixel segmentation matrix, as defined in our paper, which we will refer to as matrix A.

Lemma 1.1 (Neyman-Pearson for Gaussians with different means)**.** *Let $X \sim \mathcal{N}(x, \sigma^2)$ and $Y \sim \mathcal{N}(x+\delta, \sigma^2)$. Let $h : \mathbb{R}^d \to \{0, 1\}$ be any deterministic or random function. Then:*

1. *If $S = \{z \in \mathbb{R}^d : \delta^T z \leq \beta\}$ for some β and $\mathbb{P}(h(X) = 1) > \mathbb{P}(X \in S)$, then $\mathbb{P}(h(Y) = 1) \geq \mathbb{P}(Y \in S)$*
2. *If $S = \{z \in \mathbb{R}^d : \delta^T z \geq \beta\}$ for some β and $\mathbb{P}(h(X) = 1) \leq \mathbb{P}(X \in S)$, then $\mathbb{P}(h(Y) = 1) \leq \mathbb{P}(Y \in S)$*

First, we define a new prediction rule unlike cohens method as follows:

$$g(x) = \arg\max_{\epsilon \in \mathcal{Y}} \mathbb{P}(f(A(x + \epsilon)) = c) \tag{1}$$

where $\epsilon \sim \mathcal{N}(0, \sigma^2 \mathbf{I})$

An important point is that I am using A instead of $A(x + \epsilon)$ and it is ok because we will derive a certification rule for this new $g(x)$ and not use cohens method which will not typically be applicable. Also if successful it will be better because computing $A(x + \epsilon)$ is costly and also the superpixels change with noise and if we could derive a bound that only uses A which can be computed on only a single not noisy image, then we have solved all the problems we previously had.

For brevity, define the random variables

$$\begin{aligned}
X &:= x + \epsilon = \mathcal{N}(x, \sigma^2 \mathbf{I}) \\
Y &:= x + \delta + \epsilon = \mathcal{N}(x + \delta, \sigma^2 \mathbf{I}) \\
X' &:= Ax + A\epsilon = \mathcal{N}(Ax, \sigma^2 AA^T) \\
Y' &:= Ax + A\delta + A\epsilon = \mathcal{N}(Ax + A\delta, \sigma^2 AA^T)
\end{aligned}$$

In this notation, we know that

$$\mathbb{P}(f(AX) = c_A) \geq p_A \quad \text{and} \quad \mathbb{P}(f(AX) = c_B) \leq p_B \tag{8}$$

For brevity I will call A as simply A.

Define the half-spaces:

$$\begin{aligned}
S_1 &:= \left\{z : (A\delta)^T (z - Ax) \leq \sigma \|AA^T \delta\| \Phi^{-1}(p_A)\right\} \\
S_2 &:= \left\{z : (A\delta)^T (z - Ax) \geq \sigma \|AA^T \delta\| \Phi^{-1}(1 - p_B)\right\}
\end{aligned}$$

Lemma 1.2. $\mathbb{P}(X' \in S_1) = p_A$

Proof.

$$\begin{aligned}
\mathbb{P}(X' \in S_1) &= \mathbb{P}\left((A\delta)^T(X' - Ax) \le \sigma\|AA^T\delta\|\Phi^{-1}(p_A)\right) \\
&= \mathbb{P}\left((A\delta)^T A\mathcal{N}(0, \sigma^2\mathbf{I}) \le \sigma\|AA^T\delta\|\Phi^{-1}(p_A)\right) \\
&= \mathbb{P}\left(\sigma\|AA^T\delta\|Z \le \sigma\|\sigma\|AA^T\delta\|\Phi^{-1}(p_A)\right) \quad \text{(where } Z \sim \mathcal{N}(0,1)) \\
&= \Phi(\Phi^{-1}(p_A)) \\
&= p_A
\end{aligned}$$

Lemma 1.3. $\mathbb{P}(X' \in S_2) = p_b$

Similarily it can be shown that $\mathbb{P}(X' \in S_2) = p_b$

Now by applying Neyman-Pearson for X' and Y' which are Gaussians with different means ($A\delta$) and seting $h(z) = 1[f(z) = c_a]$, we get the following:

$$\mathbb{P}(f(Y') = c_a) \ge \mathbb{P}(Y' \in S_1)$$

It is similar to show $\mathbb{P}(f(Y') = c_b) \le \mathbb{P}(Y' \in S_2)$

To guarantee robustness we only need to set $\mathbb{P}(Y' \in S_2) \le \mathbb{P}(Y' \in S_1)$ to complete the result.

Lemma 1.4. $\mathbb{P}(Y' \in S_1) = \Phi\left(\Phi^{-1}(p_A) - \frac{\delta^T A\delta}{\sigma\|AA^T\delta\|}\right)$

Proof.

$$\begin{aligned}
\mathbb{P}(Y' \in S_1) &= \mathbb{P}\left((A\delta)^T(Y - Ax) \le \sigma\|AA^T\delta\|\Phi^{-1}(p_A)\right) \\
&= \mathbb{P}\left((AA^T\delta)^T\mathcal{N}(0, \sigma^2\mathbf{I}) \le \sigma\|AA^T\delta\|\Phi^{-1}(p_A) - \delta^T A\delta\right) \\
&= \mathbb{P}\left(\sigma\|AA^T\delta\|Z \le \sigma\|AA^T\delta\|\Phi^{-1}(p_A) - \delta^T A\delta\right) \quad \text{(where } Z \sim \mathcal{N}(0,1)) \\
&= \mathbb{P}\left(Z \le \Phi^{-1}(p_A) - \frac{\delta^T A\delta}{\sigma\|AA^T\delta\|}\right) \\
&= \Phi\left(\Phi^{-1}(p_A) - \frac{\delta^T A\delta}{\sigma\|AA^T\delta\|}\right)
\end{aligned}$$

□

Lemma 1.5. $\mathbb{P}(Y' \in S_2) = \Phi\left(\Phi^{-1}(p_A) + \frac{\delta^T A\delta}{\sigma\|AA^T\delta\|}\right)$

Now by setting $\mathbb{P}(Y' \in S_1) \ge \mathbb{P}(Y' \in S_2)$, gives us the following certification result:

Theorem 1.6 (PPRS General Elipsoid)**.** *PPRS outputs the same class for all δ satisfying the following:*

$$\frac{\delta^T A\delta}{\|AA^T\delta\|} \le \frac{\sigma}{2}(\Phi^{-1}(p_A) - \Phi^{-1}(p_B))$$

To see that this is indeed a correct generalization of Cohens method, simply put A as $\mathbb{I}$ and we will get the same bound namely:

$$\frac{\delta^T \mathbb{I}\delta}{\|\mathbb{I}\mathbb{I}^T\delta\|} \le \frac{\sigma}{2}(\Phi^{-1}(p_A) - \Phi^{-1}(p_B))$$

$$\rightarrow \|\delta\| \le \frac{\sigma}{2}(\Phi^{-1}(p_A) - \Phi^{-1}(p_B))$$

By using the properties of the PPRS matrix A we could simplify the bound even more. we set A as a $d \times d$ matrix in which each superpixel cluster in each column is $\frac{1}{S_i}$ where S_i is the size of the cluster and 0 in other elements of the column. It is easy to show that $AA^T = A$ by having the mentioned constraint and we could derive:

$$\frac{\delta^T A\delta}{\|AA^T\delta\|} = \frac{\delta^T A\delta}{\|A\delta\|} = \frac{\delta^T A\delta}{\sqrt{\delta^T A^T A\delta}} = \sqrt{\delta^T A\delta}$$

Thus it will be an ellipsoid which the semiaxes will be the eigenvectors of A.

Theorem 1.7 (PPRS Superpixel Cool Theorem)**.** *PPRS outputs the same class for all δ satisfying the following:*

$$\sqrt{\delta^T A\delta} \leq \frac{\sigma}{2}(\Phi^{-1}(p_A) - \Phi^{-1}(p_B))$$

Define $C(x) = \frac{\sigma}{2}(\Phi^{-1}(p_A) - \Phi^{-1}(p_B))$

Theorem 1.8 (Certified Volume of PPRS Superpixel)**.** *The certified volume of PPRS superpixel will be the following: The volume V of the n-dimensional ellipsoid defined by $\sqrt{x^T Ax} \leq C(x)$:*

$$V = \frac{\pi^{n/2}}{\Gamma\left(\frac{n}{2}+1\right)} \cdot \frac{C(x)^n}{\sqrt{\det(A}}$$

□

2 Future Challenges

In the above theorems, we were using a static superpixel algorithm while many pixel partioning are dynamic and how to adapt a dynamic partitioning matrix remains a future challenge