## Notes on the density estimation algorithms

These comments relate to the density estimation code **version 3**, using the variable KDE class based on Scipy's KDE code.

# Method

The implementation of our method of building crowd density estimates is very similar to variable kernel density estimation. In variable KDE the Gaussian 'bump' placed on each point has variable bandwidth. In our case, the bandwidths are given by the data as uncertainty values (errors).

For each MAC address we have a set of estimated positions and associated error values. From this data we wish to build a 2-D density estimate for that particular MAC address. We do this by evaluating a Gaussian kernel function on chosen grid points. We iterate over the data points (the estimated positions) and for each data point we compute the contribution of that point to all the evaluation grid points at once, using the linear algebra functionality of Numpy. That is, we evaluate the bivariate normal kernel, given by

$$\hat{f}(\mathbf{y}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{|2\pi\boldsymbol{\Sigma}|}} \cdot e^{-((\mathbf{y}-\boldsymbol{\mu})'\boldsymbol{\Sigma}^{-1}(\mathbf{y}-\boldsymbol{\mu}))/2} \tag{1}$$

where $\mathbf{y}$ is $(2 \times M)$-array of $M$ evaluation grid points, $\boldsymbol{\mu}$ is the length-2 vector of the data point (estimated position), and $(\mathbf{y} - \boldsymbol{\mu})$ is a $(2 \times M)$-array holding the $x$ and $y$ distances between the data point and all $M$ grid points.

# Issues

(1) What to do when either $\sigma_x = 0$ or $\sigma_y = 0$? (this occurs!) In either case, $\boldsymbol{\Sigma}$ is singular (and $\boldsymbol{\Sigma}^{-1}$ does not exist), and the 2-D Gaussian bump collapses to a 1-D Gaussian 'ridge'. For example, if $\sigma_x = 0$, the Gaussian bump collapses to a Gaussian ridge centered on $\boldsymbol{\mu} = (x_i, y_i)$ and width $\sigma_y$. If an evaluation point is not exactly on this ridge, i.e. $x_p \neq x_i$, then theoretically the contribution to this point from the Gaussian ridge is zero! An alternative approach is to set the gridpoint nearest to $\boldsymbol{\mu} = (x_i, y_i)$ to unity. This is the appropriate approach when both $\sigma_x$ and $\sigma_y$ are zero. Another approach is to build the ridge anyway, ignoring the difference $\delta x$.