CSC320 Tutorial Notes

Marcus Brubaker (t4brubak@cdf.toronto.edu)

Feb 3, 2006

Gaussian Weight Function

The Gaussian function is an extremely important function in many areas. Most of this importance is derived from its signifigance as the probability density function for the normal distribution. As a weighting function it expresses the idea that we want points close to the center to be important and points far away to be relatively insignificant.

The Gaussian is of the form

$$\Omega(x) = e^{(-\frac{(x-\mu)^2}{2\sigma^2})}$$

where μ is the location of the center and σ determines what close means. In terms of a probability distribution, μ is the mean and σ is the standard deviation. *[Draw picture of a Gaussian here. Label μ and σ for reference later.]*

Some properties of the Gaussian:

- $\int_{-\infty}^{\infty} e^{(-\frac{(x-\mu)^2}{2\sigma^2})} dx = \sigma\sqrt{2\pi}$ Note that in this form , Ω is not a vaild PDF.

- The maximum of $e^{(-\frac{(x-\mu)^2}{2\sigma^2})}$ is 1 and is located at $x = \mu$.

- The Gaussian is said to be "light-tailed" because the area under the curve is heavily concentrated around μ and therefore has very little weight in the tails. For instance,

$$\int_{\mu-\sigma}^{\mu+\sigma} e^{(-\frac{(x-\mu)^2}{2\sigma^2})} \approx .683\sigma\sqrt{2\pi}$$

$$\int_{\mu-2\sigma}^{\mu+2\sigma} e^{(-\frac{(x-\mu)^2}{2\sigma^2})} \approx .954\sigma\sqrt{2\pi}$$

$$\int_{\mu-3\sigma}^{\mu+3\sigma} e^{(-\frac{(x-\mu)^2}{2\sigma^2})} \approx .997\sigma\sqrt{2\pi}$$

Sketch the areas to emphasize how much its mass is concentrated around μ.

Sliding Window Algorithm

Lets take a row of pixels along which we want to estimate derivatives.

$$I = (47, 37, 27, 12, 6, 5, \ldots)^T$$

corresponding to pixels

$$p = (1, 2, 3, 4, 5, 6, \ldots)^T$$

The data is from a noised quadratic.

For simplicity we'll fit a linear model (i.e., the degree of the polynomial is 1) and use a window of ± 2 pixels. Then the first (symmetric) window in the row is

$$
\begin{aligned}
I_w &= (47, 37, 27, 12, 6)^T \\
p_w &= (1, 2, 3, 4, 5)^T
\end{aligned}
$$

To estimate the derivatives of $I(3)$ we set up a system like we showed last week in tutorial

$$X \begin{bmatrix} I(3) \\ I'(3) \end{bmatrix} = I_w$$

where

$$
\begin{aligned}
X &= \begin{bmatrix}
1 & (p_w(1) - 3) \\
1 & (p_w(2) - 3) \\
1 & (p_w(3) - 3) \\
1 & (p_w(4) - 3) \\
1 & (p_w(5) - 3)
\end{bmatrix} \\
&= \begin{bmatrix}
1 & (1 - 3) \\
1 & (2 - 3) \\
1 & (3 - 3) \\
1 & (4 - 3) \\
1 & (5 - 3)
\end{bmatrix}
\end{aligned}
$$

is the coefficient matrix. Solving gives us

$$\begin{bmatrix} I(3) \\ I'(3) \end{bmatrix} = \begin{bmatrix} 25.8 \\ -10.7 \end{bmatrix}$$

Then at the next step we move the window to compute the derivative at $I(4)$ so

$$
\begin{aligned}
I_w &= (37, 27, 12, 6, 5)^T \\
p_w &= (2, 3, 4, 5, 6)^T \\
X &= \begin{bmatrix}
1 & (2 - 4) \\
1 & (3 - 4) \\
1 & (4 - 4) \\
1 & (5 - 4) \\
1 & (6 - 4)
\end{bmatrix}
\end{aligned}
$$

and by similarly solving

$$X \begin{bmatrix} I(4) \\ I'(4) \end{bmatrix} = I_w$$

we get

$$\begin{bmatrix} I(4) \\ I'(4) \end{bmatrix} = \begin{bmatrix} 17.4 \\ -8.5 \end{bmatrix}$$

and we can repeat this over and over for every pixel in the row.

Speed   This may seem like a lot of work to do for every pixel and it is! However, notice that in both examples the X matrix worked out to be the same thing! This means that we don't have to recompute X and, if we compute the psuedo-inverse of X, we don't have to solve a system every time and instead just have to do a matrix multiplication to get all the coefficients.

However, we may not want all of the coefficients. For instance, we may just want the derivatives in which case the first coefficient is irrelevant. In this case we can speed things up even further,

$$\begin{bmatrix} I(-) \\ I'(-) \\ \vdots \\ I^{(n)}(-) \end{bmatrix} = X^* I_w$$

and

$$I^{(i)} = x_i^* \cdot I_w$$

where x_i^* is the ith row of X^*. This means that we can compute only the coefficients we need and the work required to do this is a dot product of two vectors.

For the examples above (a centered window of width 5 and a degree 1 fit) then

$$X = \begin{bmatrix} 1 & -2 \\ 1 & -1 \\ 1 & 0 \\ 1 & 1 \\ 1 & 2 \end{bmatrix}$$

and

$$X^* = \begin{bmatrix} .2 & .2 & .2 & .2 & .2 \\ -.2 & -.1 & 0 & .1 & .2 \end{bmatrix}$$

so to compute the first derivative of a pixel, all that is needed is to compute the dot product between $(-.2, -.1, 0, .1, .2)^T$ and the image values I_w. For instance,

$$\begin{aligned} I'(3) &= (-.2, -.1, 0, .1, .2)^T \cdot (47, 37, 27, 12, 6)^T \\ &= -10.7 \end{aligned}$$

3