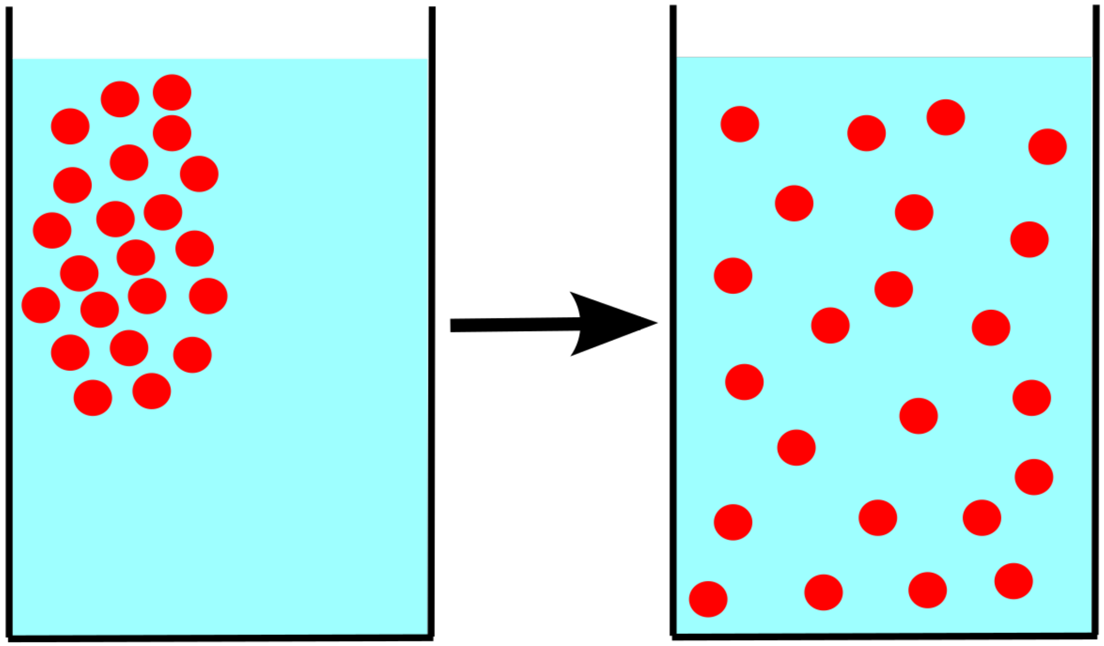
Diffusion on a Graph

Diffusion



Due to random motion, molecules of a high concentration will tend to flow towards a region in space where the concentration is lower.

Examples: A dye injected into solution spreading through a container, or heat spreading from a region of high temperature to a region of lower temperature.

The Diffusion Equation

x=0 x=L

Consider diffusion in one dimension (x) over time (t) and let u(x,t) be the concentration of the substance that is diffusing. Then

$$\frac{\partial u}{\partial t} = D\frac{\partial^2 u}{\partial x^2}$$

is the diffusion equation with diffusion coefficient D. One would also need to supply **initial values** for u, $u(x,0)=u_0(x)$, and **boundary conditions** at each boundary.

The Diffusion Equation

To describe diffusion in a domain with more than one dimension, the second partial derivative operator is replaced with the Laplacian operator. Then the diffusion equation is,

$$\frac{\partial u}{\partial t} = D\nabla^2 u$$

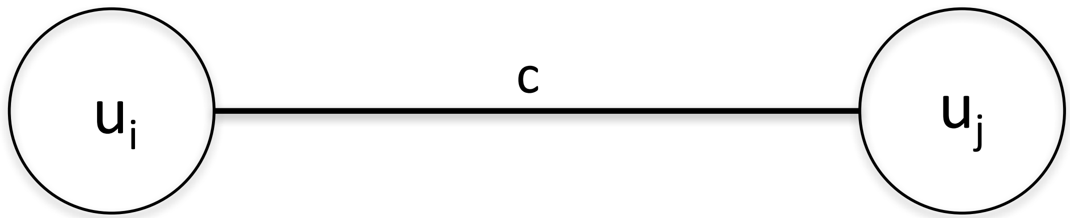
where in three dimensions

$$\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$$

Laplacian operator

Diffusion on a Graph

What if the diffusing substance moves along edges of a graph from node to node? In this case, the domain is discrete, not a continuum.



Let c be the diffusion rate across the edge, then the amount of substance that moves from node j to node i over a time period dt is $c(u_j - u_i)dt$ and from node i to node j is $c(u_i - u_j)dt$. So

$$\frac{du_i}{dt} = c(u_j - u_i)$$
$$\frac{du_j}{dt} = c(u_i - u_j)$$

Diffusion on a Graph

Diffusion to and from node i must take into consideration all nodes in the graph. The connectivity of the graph is encoded in the adjacency matrix. Here we assume that we are working with a simple graph.

$$\frac{du_i}{dt} = cA_{i1}(u_1 - u_i) + cA_{i2}(u_2 - u_i) + \cdots + cA_{in}(u_n - u_i)$$

or

$$\boxed{\frac{du_i}{dt} = c\sum_{j=1}^{n} A_{ij}(u_j - u_i)}$$

Diffusion on a Graph

Rewriting the last expression,

$$\frac{du_i}{dt} = c\sum_{j=1}^{n} A_{ij}u_j - cu_i \underbrace{\sum_{j=1}^{n} A_{ij}}_{\text{Degree of node i, } d_i}$$

$$= c\sum_{j=1}^{n} A_{ij}u_j - cu_i d_i$$

We now make use of the Kronecker delta, δ_{ij}

$$\delta_{ij} = \begin{cases} 0, \text{ if } i \neq j \\ 1, \text{if } i = j \end{cases}$$

Diffusion on a Graph

so $$c u_i d_i = c \sum_{j=1}^{n} \delta_{ij} u_j d_j$$

$$\frac{du_i}{dt} = c \sum_{j=1}^{n} A_{ij} u_j - c \sum_{j=1}^{n} \delta_{ij} u_j d_j$$

Define the n-dimensional vector $\vec{u} = \begin{pmatrix} u_1 \\ \cdots \\ u_n \end{pmatrix}$

Then $c\sum_{j=1}^{n} A_{ij} u_j = c[A\vec{u}]_i$ inner product of row i of A with $\vec{u}$

Next define the $n \times n$ degree matrix $D = \begin{bmatrix} d_1 & 0 & 0 \\ 0 & \ldots & 0 \\ 0 & 0 & d_n \end{bmatrix}$

Then $c\sum_{j=1}^{n} \delta_{ij} u_j d_j = c[D\vec{u}]_i$ inner product of row i of D with $\vec{u}$

The Graph Laplacian

so
$$\frac{du_i}{dt} = c\sum_{j=1}^{n} A_{ij}u_j - c\sum_{j=1}^{n} \delta_{ij}u_j d_j$$

becomes
$$\frac{d\vec{u}}{dt} = cA\vec{u} - cD\vec{u}$$
$$= c(A - D)\vec{u}$$

or
$$\frac{d\vec{u}}{dt} + c(D - A)\vec{u} = \vec{0}$$

We now define the Graph Laplacian matrix,

$$\boxed{L \equiv D - A}$$

The equation for diffusion on a graph is then

$$\boxed{\frac{d\vec{u}}{dt} + cL\vec{u} = \vec{0}}$$

The Graph Laplacian

What's inside of *L*?

$$L_{ij} = \begin{cases} d_i \text{ , if } i = j \\ -1 \text{ , if } i \neq j \text{ and there is an edge} \\ 0 \text{ , if } i \neq j \text{ and there is no edge} \end{cases}$$

Is *L* symmetric? Yes, why?

Solving the Graph Diffusion Equation

$$\frac{d\vec{u}}{dt} + cL\vec{u} = \vec{0}$$

This is a linear system of ODEs, so it is solvable. Also, since *L* is symmetric it has real eigenvalues and **orthogonal eigenvectors**, $\vec{v}_i$, i= $1, \cdots, n$.

Now write the solution as a linear combination of these eigenvectors, noting that the coefficients change over time: $\vec{u} = \sum_{i=1}^{n} a_i(t)\vec{v}_i$.

Insert this into the ODE, $$\sum_{i=1}^{n} \frac{da_i}{dt}\vec{v}_i + \sum_{i=1}^{n} ca_i L\vec{v}_i = 0$$

$$\Longrightarrow \quad \sum_{i=1}^{n} \left(\frac{da_i}{dt} + ca_i\lambda_i\right)\vec{v}_i = 0$$

where λ_i is an eigenvalue of *L*.

Solving the Graph Diffusion Equation

Now take the inner product of both sides of the last equation with each of the eigenvectors, recalling that they form an orthogonal set. This leads to *n* differential equations for the coefficients $a_i(t)$.

$$\frac{da_i}{dt} + c\lambda_i a_i = 0 \text{ , } i=1,\ldots,n$$

These ODEs are uncoupled and linear, so they have simple exponential solutions:

$$a_i(t) = a_i(0)e^{-c\lambda_i t}$$

where $a_i(0)$ is the initial value of the coefficient.

Since each coefficient has such a solution, then by the superposition principle, a linear combination of these is also a solution. Thus, the general solution to the graph diffusion differential equation is

$$\boxed{\vec{u}(t) = \sum_{i=1}^{n} a_i(0)e^{-c\lambda_i t}\vec{v}_i}$$

Spectral Solution

Solving the Graph Diffusion Equation

How do we find the initial values of the coefficients?

Use the initial distribution of *u* among the nodes.

$$\vec{u}(0) = \sum_{i=1}^{n} a_i(0)\vec{v}_i$$

Now take the inner product of both sides with an eigenvector

$$\vec{u}(0) \cdot \vec{v}_j = a_j(0)\left|\vec{v}_j\right|^2$$

$$\Rightarrow \quad \boxed{a_j(0) = \frac{\vec{u}(0) \cdot \vec{v}_j}{\left|\vec{v}_j\right|^2}}$$

Spectral Properties of the Graph Laplacian

By spectral properties, we mean properties of the eigenvalues and eigenvectors.

Since *L* is symmetric, its eigenvalues are real and its eigenvectors are orthogonal

Is *L* singular or non-singular?

Look at any row *i.* The diagonal element is the degree of the node, d_i. All the other elements are either 0 or, for each edge, -1. There are exactly d_i of these, so if you sum across any row of *L* you get $d_i - d_i = 0$. This is true for any of the rows. So the sum of all columns of the matrix is $\vec{0}$. Therefore, *L* is singular. That is, it has at least one zero eigenvalue. Call it $\lambda_1 = 0$.

Spectral Properties of the Graph Laplacian

What is the eigenvector associated with the zero eigenvalue? That is, the vector $\vec{v}_1$ such that $L\vec{v}_1 = \vec{0}$?

It must be a vector of 1s, $\vec{1}$. Why?

Because, $L\vec{1}$ is the sum of the columns of *L*, which we know equals $\vec{0}$.

Does *L* have any negative eigenvalues?

Suppose that $\lambda_2 < 0$. Then the term in the spectral solution

$$a_2(0)e^{-c\lambda_2 t}\vec{v}_2 \to \infty \quad \text{as} \quad t \to \infty$$

which we know can't happen (think about spreading die, does its concentration go to infinity anywhere in the domain?)

So *L* has non-negative eigenvalues, which is called positive semidefinite.

Spectral Properties of the Graph Laplacian

Suppose that the graph has two components. How is that reflected in the eigenvalues?

Label the nodes so that the first n_1 correspond to one component and the last $n_2 = n - n_1$ correspond to the other component. This results in a block diagonal graph Laplacian matrix

$$L = \begin{bmatrix} \overbrace{\begin{bmatrix} L_1 \end{bmatrix}}^{n_1} & 0 \\ 0 & \overbrace{\begin{bmatrix} L_2 \end{bmatrix}}^{n_2} \end{bmatrix}$$

Spectral Properties of the Graph Laplacian

$$L = \begin{bmatrix} \overbrace{\begin{bmatrix} L_1 \end{bmatrix}}^{n_1} & 0 \\ 0 & \overbrace{\begin{bmatrix} L_2 \end{bmatrix}}^{n_2} \end{bmatrix}$$

Define $\vec{v}_1 = (\overbrace{1,1,1,\cdots}^{n_1}, 0,0,0,0 \cdots)$

$\vec{v}_2 = (0,0,0,\cdots, \overbrace{1,1,1,1 \cdots}^{n_2})$

Then $L\vec{v}_1 = L_1\vec{1} = \vec{0}$ and $L\vec{v}_2 = L_2\vec{1} = \vec{0}$

So $\vec{v}_1$ and $\vec{v}_2$ are both eigenvectors of L with 0 eigenvalues, and L has two 0 eigenvalues.

Spectral Properties of the Graph Laplacian

In general, the number of 0 eigenvalues of the graph Laplacian is equal to the number of components of the graph.

One can order the eigenvalues of L from smallest to largest. Then

$$\lambda_1 = 0$$

If $\lambda_2 \neq 0$ then the graph is connected

If $\lambda_2 = 0$ then the graph is disconnected

So λ_2 is called the algebraic connectivity of the graph.

Asymptotic Solution

Recall that the spectral solution to the graph diffusion equation is

$$\vec{u}(t) = \sum_{i=1}^{n} a_i(0) e^{-c\lambda_i t} \vec{v}_i$$

Suppose that the graph is connected. Then $\lambda_1 = 0$ and all other eigenvalues are positive. There are *n* terms in the solution above. What happens to these terms as $t \rightarrow \infty$?

All approach 0,except for the first term, which is independent of time. So the asymptotic solution is just the first term of the spectral solution,

$$\vec{u}_\infty = a_1(0)\vec{v}_1$$

Asymptotic Solution

$$\vec{u}_\infty = a_1(0)\vec{v}_1$$

But $$a_1(0)\vec{v}_1 = \left(\frac{\vec{u}(0)\cdot\vec{v}_1}{|\vec{v}_1|^2}\right)\vec{1} = \left[\frac{u_1(0)+\cdots+u_n(0)}{n}\right]\vec{1}$$

So $$\boxed{\vec{u}_\infty = \left[\frac{u_1(0)+\cdots+u_n(0)}{n}\right]\vec{1}}$$

How can we interpret this physically?

In the long term, each node in the connected graph gets the same share of the dye (or whatever is diffusing), which is equal to the total amount initially present divided by the number of nodes.

Asymptotic Solution

This can also be derived directly from the diffusion equation

$$\frac{d\vec{u}}{dt} + cL\vec{u} = \vec{0}$$

Set time derivative to 0,

$$L\vec{u}_{\infty} = \vec{0}$$

So the equilibrium vector is an eigenvector of the graph Laplacian corresponding to the 0 eigenvalue, which is what we just saw using the different approach. The length of the equilibrium vector is just the sum of the initial values of u:

$$|\vec{u}_{\infty}| = \sum_{i=1}^{n} u_i(0)$$

Asymptotic Solution

We can rewrite this equilibrium equation by deconstructing the graph Laplacian

$$L\vec{u}_\infty = \vec{0}$$

or

$$(D\text{-}A)\vec{u}_\infty = \vec{0}$$

$$\Rightarrow \quad D\vec{u}_\infty = A\vec{u}_\infty$$

$$\Rightarrow \quad \vec{u}_\infty = D^{-1}A\vec{u}_\infty$$

Hold on now, is *D* invertible? Yes, as long as the graph is connected

Asymptotic Solution

Since $$\vec{u}_\infty = D^{-1} A \vec{u}_\infty$$

$$* \qquad u_{\infty,i} = \frac{\sum_{j=1}^{n} A_{ij} u_{\infty,j}}{d_i}$$

where we note that $A_{ii} = 0$ when there are no self-edges. If This is the same at each node, as would be the case for diffusion in a uniform medium, then

$$\sum_{j=1}^{n} A_{ij} u_{\infty,j} = d_i u_{\infty,i}$$

and the equation * is satisfied.

In the end, all nodes share equal amounts of the substance that was initially introduced.

Heterogeneous Diffusion

So far we have thought of the graph as an unweighted graph. That is, diffusion between any pair of nodes has the same rate, *c*. This is called homogeneous diffusion. More generally, each edge can have its own diffusion rate, which is called heterogeneous diffusion. So now the adjacency matrix has weights as its elements (or 0s), and

$$L_{ij} = \begin{cases} s_i \text{ , if } i = j \\ -c_{ij} \text{ , if } i \neq j \text{ and there is an edge} \\ 0 \text{ , if } i \neq j \text{ and there is no edge} \end{cases}$$

where degree s_i is the sum of the weighted edges (i.e., the strength of the node) at node i and c_{ij} is the weight connecting nodes i and j.

Now,

$$u_{\infty,i} = \frac{\sum_{j=1}^{n} C_{ij} u_{\infty,j}}{s_i}$$

This equation is satisfied if all nodes end up with equal concentrations: $u_{\infty,i} = u_{\infty,j}$ for each i, j. That is, once again, at equilibrium the nodes equally divide the initial amount of substance.

The Filter Matrix

Define a new matrix, *W*, which I'll call a filter matrix

$$W \equiv D^{-1}A$$

Then

$$\Rightarrow \quad \boxed{\vec{u}_\infty = W\vec{u}_\infty}$$

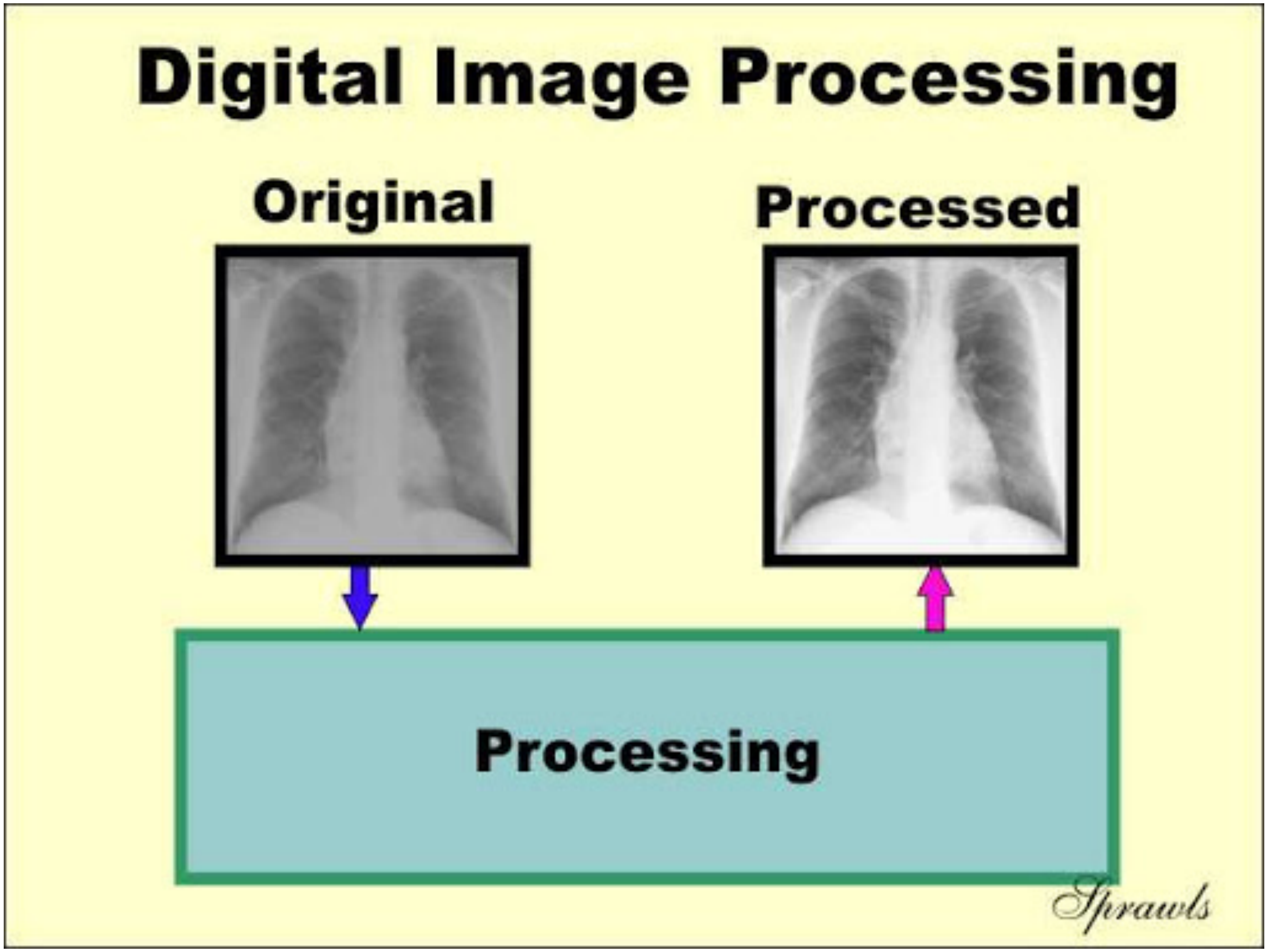
How can we interpret this? One way is to think about the following **first-order linear recursion** or **difference equation**:

$$\vec{u}_{k+1} = W\vec{u}_k$$

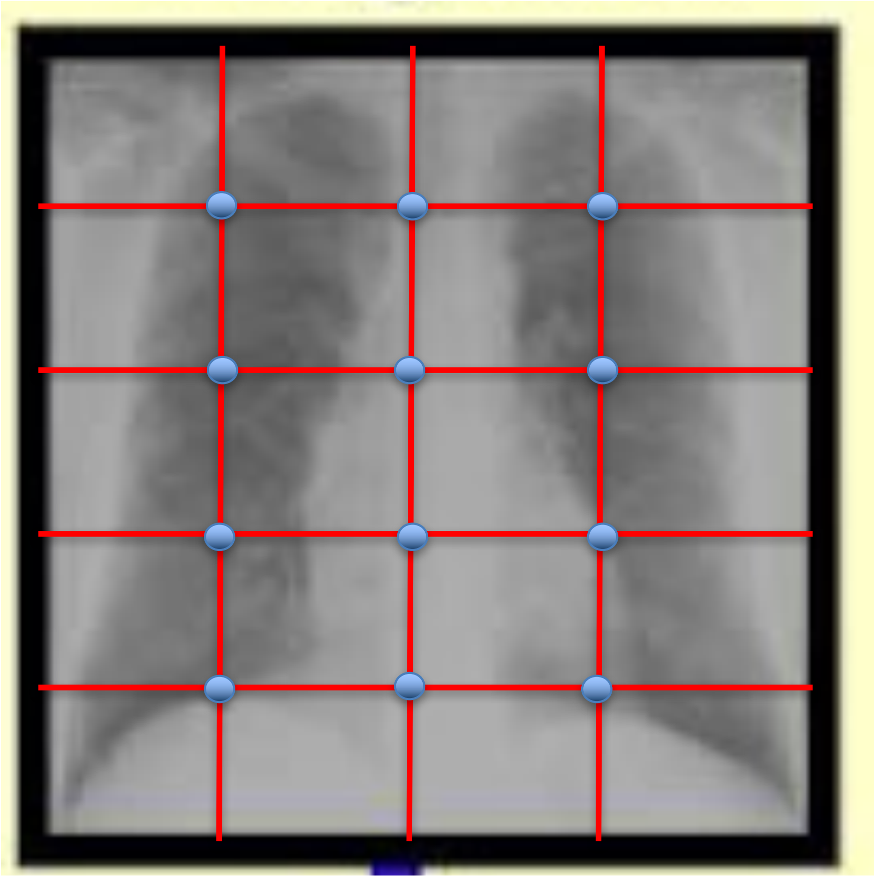
starting from the initial vector $\vec{u}_0$. What happens if you iterate forever?

Then ultimately if the system converges (and it will), the k+1 iterate will be the same as the k iterate. This gives the equilibrium equation above. The equilibrium vector $\vec{u}_\infty$ is therefore the fixed point of the recursion with the filter matrix.

Image Processing

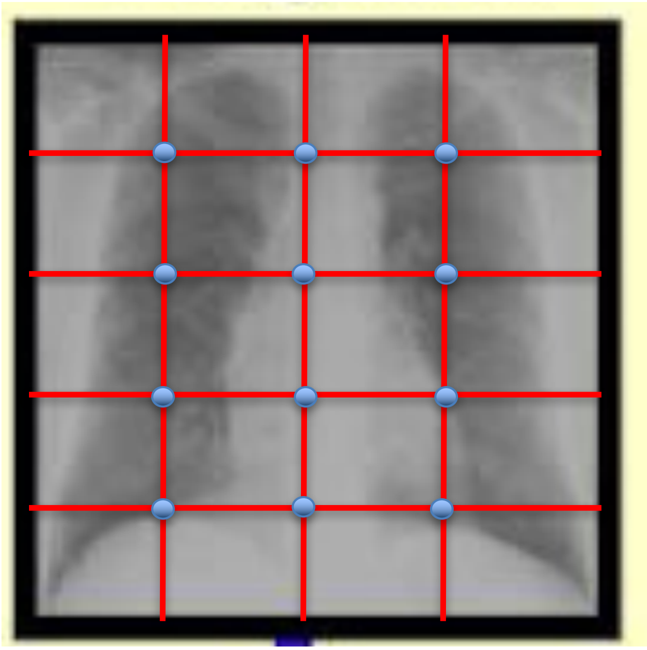


Pixelated Image as a Network



Each point on the grid has a grey level: 0=white, 1=black, with shades of grey corresponding to intermediate values. These grid points are the **nodes** of a network and their grey level is the value of *u* at that node.

Pixelated Image as a Network

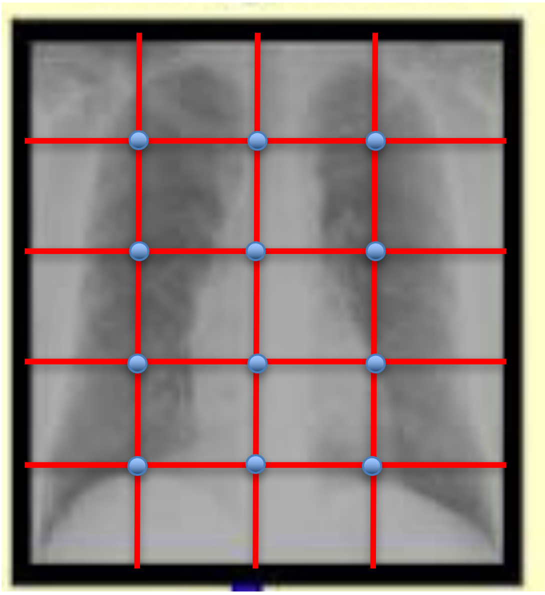


What are the edges?

Most generally, connect each node pair with an edge, but make the edges weighted (c_{ij}) by the affinity of each of the connected nodes. Here, affinity means how similar they are.

Similar could mean location (nearest neighbors get highest affinity). Or it could mean the grey level of the pixels (similar *u* values of nodes). If it is based on location, then the edge weights are fixed. If based on grey level, edge weights are functions of the *u* values.

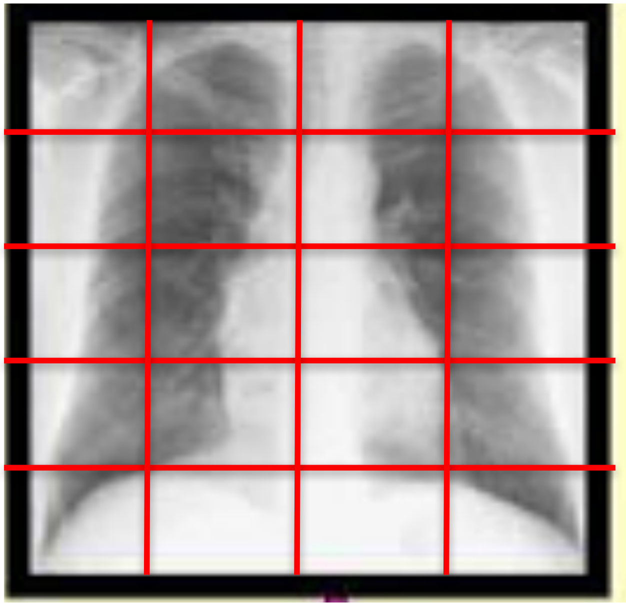
Modifying an Image With the Filter Matrix



Let $\vec{u}_0$ be the grey level values of the original image. Image processing multiplies that vector by the filter matrix to get a product vector, $\vec{u}_1 = W\vec{u}_0$. It typically does this more than once, applying the recursion relation we saw earlier, $\vec{u}_{k+1} = W\vec{u}_k$.

The effect this has on the image depends on *W*. If the affinities are chosen to be neighbors in physical space, then there is a static network and the effect of the filtering will be to average out differences among neighbors. The image becomes smoother.

Modifying an Image With the Filter Matrix



If the affinities are chosen according to similarity in grey level, then applying the filter will have the effect of making similar pixels more similar, which tends to sharpen the image. In this case, the weights c_{ij} and therefore the elements of W are updated with each iteration of the recursion formula; there is a dynamic network.

In practice, combinations of affinity based on location and based on grey levels are used, and iteration stops when some measure of image goodness has been reached.

For a Great Video on Image Processing

The following link has an hour-long lecture by a researcher from Google on how the graph Laplacian is used in image processing:

https://www.youtube.com/watch?v=_ltmFYCr7ag

The End