Regressions and approximation

Prof. Graeme Bailey

http://cs1114.cs.cornell.edu

(notes modified from Noah Snavely, Spring 2009)

Cornell University
Computer Science

Object recognition

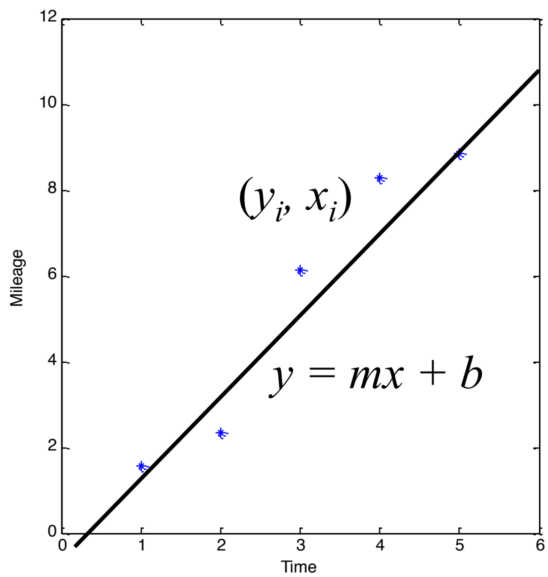
1. Detect features in two images
2. Match features between the two images
3. Select three matches at random
4. Solve for the affine transformation *T*
5. Count the number of inlier matches to *T*
6. If *T* is has the highest number of inliers so far, save it
7. Repeat 3-6 for N rounds, return the best *T*

A slightly trickier problem

- What if we want to fit *T* to more than three points?
  - For instance, all of the inliers we find?

- Say we found 100 inliers
- Now we have 200 equations, but still only 6 unknowns
- *Overdetermined* system of equations
- This brings us back to linear regression

Linear regression, > 2 points



- The 'best' line won't necessarily pass through any data point

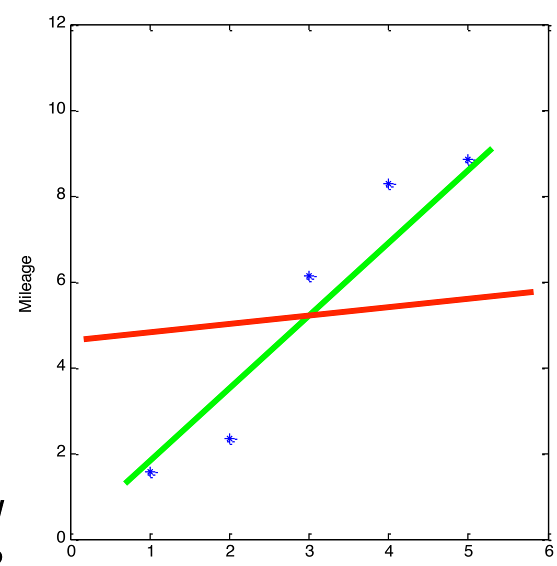
Some new definitions

- No line is perfect – we can only find the *best* model, the line y = mx + b, out of all the imperfect ones. This process is called *optimisation*.

- We'll define a function *Cost*(*m*,*b*) that measures how far a line is from the data, then use that to find the best line
  - I.e., the model [*m*,*b*] that minimizes Cost(*m*,*b*)
  - Such a function Cost(*m*,*b*) is called an *objective function*
  - Often, we're looking for a sequence of approximations which successively reduce the value of Cost(*m*,*b*)
  - There may or may not be a 'limit' to which these approximations converge.

Line goodness

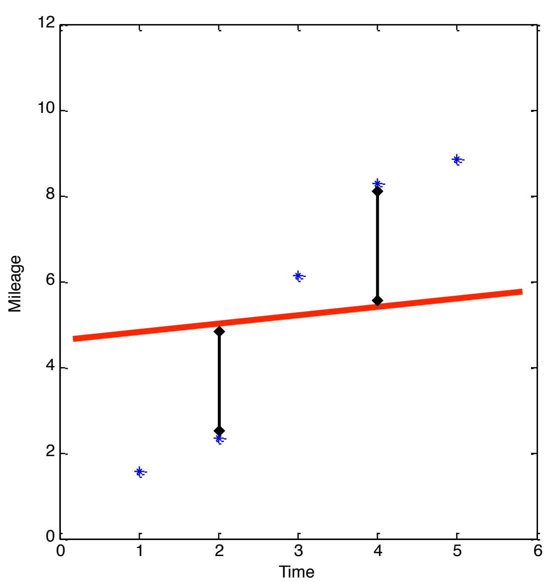
- What makes a line good versus bad?
  - This is actually a very subtle question
  - In reality, our cost function is a *distance function* or *metric* which can measure the distance between two functions in the space of possible functions.
  - There may be several different functions having the same distance from a given target function … how might we choose the 'best'?



Residual errors

- The difference between what the model predicts and what we observe is called a *residual error*
  - Consider the data point (x,y) and the model (line) y=mx+b
  - The model [m,b] predicts (x,mx+b)
  - The residual is y – (mx + b), the error from using the *model* to acquire the y value for that value of x
  - Each residual is the vertical distance to the proposed line
  - How do we measure the cumulative effect of all these residuals? There are many options here.
  - Once we've decided on our formula, how do we actually compute the least bad model (line)?   Called optimisation.

Linear vertical error fitting

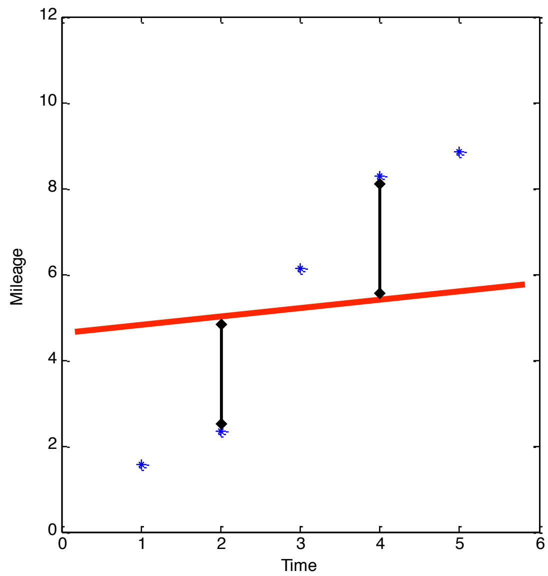


$$\mathrm{Cost}(m,b) = \sum_{i=1}^{n} |y_i - (mx_i + b)|$$

This is a reasonable cost function, but we usually use something slightly different

What are the benefits and negatives of this cost metric?

Least squares fitting



$$\text{Cost}(m,b) = \sum_{i=1}^{n} |y_i - (mx_i + b)|^2$$

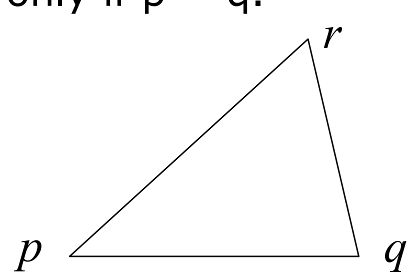
We prefer to make this a **squared** distance

Called "least squares"

What about the benefits and negatives of this version?

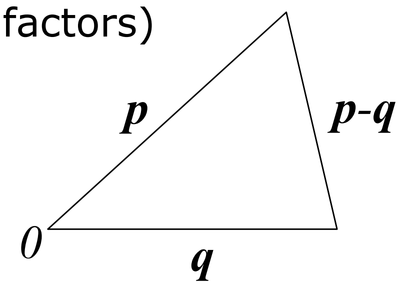
Measuring distance

- It's interesting to see how we can make precise the notions of distance, length, and angle.
- In particular, we'll see how to articulate our biases (preferences) so that we can compute for them.
- Anything which satisfies the following conditions is allowed to be called a *distance metric*
  - We'll use d(p,q) to denote the distance between p and q.
  - d(p,q) ≥ 0 always, and d(p,q) = 0 if and only if p = q.
  - d(p,q) = d(q,p) always
  - d(p,q) ≤ d(p,r) + d(r,q) always.
    - This last condition is called the triangle inequality, and says that it's never shorter to go via somewhere else!



Measuring length

- If we had a way of measuring the length of a vector **v** from p to q, then the length of **v** would be d(p,q)
- Anything which satisfies the following conditions is allowed to be called a *length* or *norm* for vectors
  - We'll use ||**v**|| to denote the norm of **v**
  - $||\mathbf{v}|| \geq 0$ always, and $||\mathbf{v}|| = 0$ if and only if $\mathbf{v} = \mathbf{0}$
  - $||a\mathbf{v}|| = |a|\ ||\mathbf{v}||$ always (stretching factors)
  - $||\mathbf{v} + \mathbf{w}|| \leq ||\mathbf{v}|| + ||\mathbf{w}||$ always.
    - This last condition is again the triangle inequality!



- If **p** and **q** are vectors from the origin to the points p and q respectively, then d(p,q) = ||**p** - **q**||

Measuring length

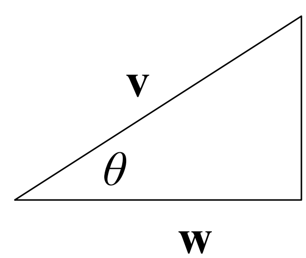
- Examples
  - In 2D, if $\mathbf{v} = x\mathbf{i} + y\mathbf{j}$ then define $||\mathbf{v}|| = \sqrt{(x^2 + y^2)}$. This gives the usual Pythagorean or Euclidean distance.
  - Also in 2D, we could define $||\mathbf{v}|| = \max(|x|, |y|)$. Notice that a 'circle' around the origin would look to us like a square!! (A circle is a set of points equidistant from its centre.)
  - In (countably) infinite dimensions, if $\mathbf{v} = \Sigma\, v_n\mathbf{e}_n$ then we can define $||\mathbf{v}|| = \sqrt{\Sigma\, v_n^2}$ (assuming that series converges), or even as $\max(|v_n|)$, by analogy with the two previous examples.
  - For functions defined on the interval [a , b] we could again extend the analogies to define ||f|| = either the square root of the integral from a to b of f(x) squared, or the integral of the absolute value of f(x).
  - Notice the different properties emphasised by the Euclidean distances versus the max flavours. There is no 'right answer', the choice as to which is 'best' depends on the context.

Measuring angle

- There's a curious, yet surprisingly powerful way to define the angle between two vectors. Anything which satisfies the following conditions is allowed to be called a *dot, scalar* or *inner product.*
  - We'll use **v.w** to denote the inner product of **v** and **w**.
  - **v.v** ≥ 0 always, and **v.v** = 0 if and only if **v** = **0**
  - **v.w** = **w.v** always ← *(assuming our scalars are reals; if they're complex numbers then we take the complex conjugate of the RHS)*
  - **v.**(a**w**) = a(**v.w**)
  - **v**.(**u** + **w**) = **v.u** + **v.w** always.
- If we have an inner product, then we can use that to define a norm (via $||\mathbf{v}||^2 = \mathbf{v.v}$) and then a metric. We *define* the angle θ between vectors **v** and **w** via the definition **v.v** = ||**v**|| ||**w**|| cos θ

Measuring angle

- Examples
  - In 2D, if $\mathbf{v} = v_1\,\mathbf{i} + v_2\,\mathbf{j}$ and $\mathbf{w} = w_1\,\mathbf{i} + w_2\,\mathbf{j}$ then we can define $\mathbf{v.w} = v_1w_1 + v_2w_2$. This gives the usual Pythagorean or Euclidean angles and distances.
  - This can be extended to (countably) infinite dimensions and to function spaces in the same way (so the inner product of two functions f and g would be the integral of f(x)g(x) over [a , b]).
  - Perhaps surprisingly, there is no analogous example of inner product from which we can obtain the max norms (we can prove that no such inner product exists – it's not just that we couldn't see how to do it!!).
  - Having a definition of angle allows us to project the component of one vector onto another. So if **ŵ** is the unit vector in the direction of **w**, then the component of **v** in the direction of **w** is (||**v**|| cos θ) **ŵ** = ((**v.w**) / (**w.w**)) **w**



Applications

- Now that we know how to formalise distance, we can define it so that we weight differently according to what we regard as important distinctions
  - If it satisfies the definition it can be used as a distance metric!
  - Given a sense of distance, we can ask about convergence (namely getting closer to a limit). So then a sequence (of functions) $f_n \longrightarrow f$ if $d(f_n , f) \longrightarrow 0$.
  - Notice that if $f_n \longrightarrow f$ then necessarily $d(f_n , f_m) \longrightarrow 0$, ie the terms of the sequence must eventually get closer together. This is nicer since we don't have to know what the limit function is in order to show that a sequence ought to converge!! (*Cauchy sequence*.)
  - Suppose we wanted to ensure that our target line *t* was 'close' to the data points and also 'close' to 'the slope' of the data. We could estimate the data's slope by taking the (weighted?) mean μ of the slopes of the piecewise linear path through the data points and then define a corresponding metric.