Section 2: Least Squares

Least squares regression:

\[
\min_\beta \sum_n (y_n - \beta x_n)^2
\]

“In the space of measurements:

[Diagram showing a line of best fit with error bars.

[Gauss, 1795 - age 18]
\[
\sum_{n} (y_n - \beta x_n)^2
\]
\[ \sum_n (y_n - \beta x_n)^2 \]

\[ \hat{\beta} = \arg \min_{\beta} \sum_n (y_n - \beta x_n)^2 \]
\[ \min_{\beta} \sum_n (y_n - \beta x_n)^2 \]
can solve this with calculus… [on board]

... or, with linear algebra!
\[ \min_{\beta} || \vec{y} - \beta \vec{x} ||^2 \]

Geometry:
Note: this is a 2-D cartoon of the N-D vectors, not the two-dimensional (x,y) measurement space of previous plots!

Note: partition of sum of squared data values:
\[ ||\vec{y}||^2 = (||\beta_{opt} \vec{x}||^2 + ||\vec{y} - \beta_{opt} \vec{x}||^2) \]
explained residual

\[ \vec{y} \]
\[ \vec{x} \]
\[ \beta_{opt} \vec{x} \]
Multiple regression: \[
\min_{\beta} ||\bar{y} - \sum_k \beta_k \bar{x}_k||^2 = \min_{\beta} ||\bar{y} - X\beta||^2
\]

2D example:

Solution via the “Orthogonality Principle”:

Construct matrix \(X\), containing columns \(\bar{x}_1\) and \(\bar{x}_2\)

Orthogonality: \(X^T(\bar{y} - X\bar{\beta}) = \bar{0}\)

Alternatively, can solve using SVD...

\[
\min_{\beta} ||\bar{y} - X\bar{\beta}||^2 = \min_{\beta} ||\bar{y} - USV^T\bar{\beta}||^2
\]
\[= \min_{\beta} ||U^T\bar{y} - SV^T\bar{\beta}||^2\]
\[= \min_{\bar{\beta}^*} ||\bar{y}^* - S\bar{\beta}^*||^2\]

where \(\bar{y}^* = U^T\bar{y}, \quad \bar{\beta}^* = V^T\bar{\beta}\)

Solution: \(\beta^*_{opt,k} = y_k^* / s_k\), for each \(k\)

or \(\bar{\beta}^*_{opt} = S^* \bar{y}^* \Rightarrow \bar{\beta}_{opt} = VS^*U^T\bar{y}\)

[on board: transformations, elliptical geometry]
Fitting a parametric model (general)

Experimental Data \( \{x_n, y_n\} \rightarrow \bar{y}_n \)

Model \( f_\beta(\bar{x}) \)

To fit model \( f_\beta(\bar{x}) \) to data \( \{x_n, y_n\} \),
optimize parameters \( \beta \) to minimize an error function:
\[
\min_\beta \sum_n E(\bar{y}_n, f_\beta(\bar{x}_n))
\]

Ingredients: data, model, error function, optimization method

Optimization

- Heuristics, exhaustive search, (pain & suffering)
- Iterative descent, (possibly) nonunique
- Convex
- Iterative descent, guaranteed
- Quadratic
- Smooth (C^2)
- Closed-form guaranteed

Interpretation warning: fitting a line does not guarantee data actually lie along a line

These 4 data sets give the same regression fit, and same error:

[Anscombe, 1973]
Polynomial regression

\[ y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 \]

(to be continued, when we get to “statistics”...)

Weighted Least Squares

\[
\min_{\beta} \sum_{n} [w_n (y_n - \beta x_n)]^2 = \min_{\beta} \| W(\vec{y} - \beta \vec{x}) \|^2
\]

Solution via simple extensions of basic regression solution
(i.e., let \( \vec{y}^* = W\vec{y} \) and \( \vec{x}^* = W\vec{x} \) then solve for \( \beta^* \) )
“Trimming”... discard points with large error (note: a special case of weighted least squares)

Trimming can be done iteratively (discard outlier, re-fit, repeat), a so-called “greedy” method. When should you stop?

More generally, use a “robust” error metric. For example:

\[
f(d) = d^2
\]

\[
f(d) = \log(c^2 + d^2)
\]

“Lorentzian”

Note: generally can’t obtain solution directly (i.e., requires an iterative optimization procedure, such as gradient descent).
In some cases, can use iteratively re-weighted least squares (IRLS)...

Iteratively Re-weighted Least Squares (IRLS)

initialize: \( w_n^{(0)} = 1 \)

\[
\beta^{(i)} = \arg \min_{\beta} \sum_{n} \omega_n^{(i)} (y_n - \beta x_n)^2
\]

\[
\omega_n^{(i+1)} = \left| \frac{f'(y_n - \beta^{(i)} x_n)}{y_n - \beta^{(i)} x_n} \right|
\]

(one of many variants)
Constrained Least Squares

Linear constraint:
\[
\arg \min_{\beta} \| \tilde{y} - X \beta \|^2, \quad \text{where } \epsilon^T \beta = 1
\]

Quadratic constraint:
\[
\arg \min_{\beta} \| X \beta \|^2, \quad \text{where } \| \beta \|^2 = 1
\]

Can be solved exactly using linear algebra (SVD)...
Standard Least Squares regression

Error is *vertical* distance (in the "dependent variable") from the fitted line...

\[ \arg \min_{\beta} \| y - \beta x \|^2 \]

Total Least Squares Regression
(a.k.a "orthogonal regression")

Error is squared distance from the fitted line...

expressed as: \( \min_{\hat{u}} \| D\hat{u} \|^2 \), where \( \| \hat{u} \|^2 = 1 \)

Note: “data” matrix \( D \) now includes both \( x \) and \( y \) coordinates

Variance of data \( D \), projected onto axis \( \hat{u} \):
\[ \| USV^T \hat{u} \|^2 = \| SV^T \hat{u} \|^2 = \| \hat{u}^* \|^2 = \| \hat{u}^{**} \|^2 \]
where \( D = USV^T \), \( \hat{u}^* = V^T \hat{u} \), \( \hat{u}^{**} = S \hat{u}^* \)

Set of \( \hat{u} \)'s of length 1 (i.e., unit vectors)
Set of \( \hat{u}^* \)'s of length 1 (i.e., unit vectors)
First two components of \( \hat{u}^{**} \) (rest are zero!), for three example \( S \)'s.
Total Least Squares Regression
(a.k.a “orthogonal regression”)

Error is squared distance from the fitted line...

expressed as: \( \min \| D\hat{u} \|^2 \), where \( \|\hat{u}\|^2 = 1 \)

Note: “data” matrix \( D \) now includes both \( x \) and \( y \) coordinates

---

Olympic gold medalists
(Rio, 2016)

3D geometry:
Javelin, Discus, Shotput…

---

Principal Component Analysis (PCA)

The shape of a data cloud can be summarized with an ellipse (ellipsoid), centered around the mean, using a simple procedure:
1. Subtract mean of all data points, to re-center around origin
2. Assemble centered data vectors in rows of a matrix, \( D \)
3. Compute the SVD:
   \[ D = USV^T \]
   or just use the smaller matrix
   \[ C = D^T D = V S^T S V^T = V \Lambda V^T \]
4. Columns of \( V \) are the *principal components* (axes) of the ellipsoid, diagonal elements \( s_k \) or \( \sqrt{\lambda_k} \) are the corresponding principle radii, and their product is the volume.
Dynamical structures, such as point attractors, line attractors, and limit cycles, arising from different subspaces, and how it varies from trial to trial. has been used in the literature to relate population recordings and network models. More activity structure used to relate neuronal recordings and network models: population activity all three components. Below we describe these studies, organized by the aspect of population circuit from which the neurons were recorded [32, 35]. The same dimensionality reduction model neuron. Dimensionality reduction can be used to obtain a concise summary of the state strongly influences the future state. Thus, two similar pat-

Consistent with these shared features, muscle responses in terms of signals that a recurrent or feedback-driven may reflect computations performed by internal and feedback somatosensory cortex. We found that the dominant signals in mo-

This finding might suggest that motor cortex activity primarily re-

Likewise, a linear model (leave-one-out-cross-validated) could be successfully decoded from the neural population using both amplitude (e.g., $a.u.$), and $3$ with corresponding eigenvalues $= 0.80$ and $= 0.5$. Indeed, we will suggest below that muscle-like encoding muscle-like commands and the need to ensure low trajectory tangling. Network simulations confirm that low trajectory tangling encode muscle-like commands and the need to ensure low tangling. Progress may be generated reliably. Counter-rotation is expected given the fact that single-neuron response profiles combined with the fact that single-neuron response profiles generally seem appealing given the present data. For example, pushing before lifting. In contrast, neural trajectories co-rotated: they orbited in opposing directions for forward muscle trajectories behaved differently. Muscle trajectories ($D$ and $4G$). Despite this basic similarity, neural and somatosensory trajectories followed repeating orbits throughout the moment when the system must rely on external commands. The same discrepancy may reflect computations performed by internal and feedback somatosensory cortex. We found that the dominant signals in mo-

| CC BY 4.0 Open Access | rec: 25 Jan 2019, publ: 25 Jan 2019 |

Example: PCA for dimensionality reduction and visualization

Eigenvectors/eigenvalues

- An eigenvector of a matrix is a vector that is rescaled by the matrix (i.e., the direction is unchanged)
- The corresponding scale factor is called the eigenvalue
- For matrix $C = D^{T}D = V\Lambda V^{T}$ the columns of $V$ (denoted $\hat{v}_{k}$) are eigenvectors, with corresponding eigenvalues $\lambda_{k}$:

$$C\hat{v}_{k} = V\Lambda V^{T}\hat{v}_{k} = V\Lambda\hat{v}_{k} = \lambda_{k}\hat{v}_{k}$$