Lab 4 — 2D Regression problem

October 5th, 2018
Outline the problem:

- Load regress1.mat
- Plot Y as function of X
- Least squares fit of data with polynomial of order 0-5
  - Using SVD
- Plot the fit
- Plot the squared errors as function of order of poly
Plot x as function of y

- This is the data we want to fit
Outline the solution:

1. Think
2. Write the expression
3. \( X = USV_t \)
4. Rotate into S-space
5. “Trim” to relevant expression
6. Solve for the combination of \( \beta \)s that minimize the expression
7. Rotate back
8. Fit your data
9. Assess your work:
   1. Plot it
   2. Calculate the error
Step 1 — Thinking

• Want to find an equation to fit our data.

• What’s our question?
• Finding the beta that fits data best, produces the smallest error from the data points
Step 1 — Thinking

- Want to find an equation to fit our data.
- What’s our question?
- Finding the beta that fits data best, produces the smallest error from the data points

$$\min_\beta ||\bar{y} - \beta \bar{x}||^2$$
Step 1 — Thinking

• 1st order example:

\[ y = mx + b \]

- High school notation

\[ \hat{y} = \beta_1 \hat{x} + \beta_0 \]

- Math tools notation

• Write it in polynomial notation:

\[ \hat{y} = \beta_1 \hat{x} + \beta_0 \hat{x}^0 \]
Step 2 — writing expression

- Write it in polynomial notation:
  \[ \hat{y} = \beta_1 x' + \beta_0 x^0 \]

- Higher orders:
  \[ y = \beta_2 x^2 + \beta_1 x' + \beta_0 x^0 \]

- Re-write in matrix form:
  \[ \begin{pmatrix} \beta_2 \\ \beta_1 \\ \beta_0 \end{pmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}^2 + \beta_1 \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}' + \beta_0 \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} \begin{bmatrix} \beta_2 \\ \beta_1 \\ \beta_0 \end{bmatrix} \]

\( \hat{y} = X \beta \) call this vector \( \beta \)

\( \min_{\beta} \| \hat{y} - X \beta \|_2^2 \) call this matrix \( X \)
Step 3 — SVD

- Our expression:
  \[ \min_{\beta} \| y - X\beta \|^2 \]

- Do svd of matrix X:
  \[ = \min_{\beta} \| U^Ty - US\sigma V^T\beta \|^2 \]

- Rearrange:
  \[ = \min_{\beta} \| U^Ty - SV\sigma V^T\beta \|^2 \]

- Why do we want to rotate y by Ut?
Step 4 — rotate into S-space

• We rotated y and β to the “S space”

\[
= \min_\beta \| U^T \hat{y} - \sigma \sqrt{\beta} \|^2
\]

• Rewrite:

\[
= \min_\beta \| \hat{y}^* - \sigma \beta^* \|^2
\]

• Notice:
  • How many non-zero values does S have for a first order polynomial?

Don’t lose track of your variables!

\[
U^T \hat{y} = \hat{y}^*
\]

\[
\sqrt{\beta} = \beta^*
\]
Step 5 — Trim

- Only 2 non-zero S values in this example:

\[
\begin{bmatrix}
\mathbf{y}^* \\
\mathbf{y}_{e}^*
\end{bmatrix} = S \begin{bmatrix} \mathbf{\beta}^* \end{bmatrix} = \begin{bmatrix} y^*_1 \\
\vdots \\
y^*_n \end{bmatrix} - \begin{bmatrix} s_1 \beta_1^* \\
\vdots \\
s_n \beta_n^* \end{bmatrix}
\]

- The betas we choose will only affect \(y^*_1\) and \(y^*_2\)

\[
\mathbf{y}^* = S \mathbf{\beta}^*
\]

- How much can we minimize this expression?

Don’t lose track of your variables!

\[
U^T \mathbf{y}^* = \mathbf{y}^*
\]

\[
\sqrt{\mathbf{\beta}} = \mathbf{\beta}^*
\]
Step 6 — solve for $\beta_{opt}$

- How much can we minimize this expression?

$$\hat{y} - s \hat{\beta}$$

$$\begin{bmatrix} y_1^* \\ y_2^* \end{bmatrix} = \begin{bmatrix} s_1 & 0 \\ 0 & s_2 \end{bmatrix} \begin{bmatrix} \beta_1^* \\ \beta_2^* \end{bmatrix}$$

Don’t lose track of your variables!

$$U^T \hat{y} = \hat{y}^*$$

$$U^T \beta = \beta^*$$

$$y_1^* - s \hat{\beta}^* = 0$$

$$y_2^* - s \hat{\beta}_{opt}^* = 0$$

$$y_1^* = s \hat{\beta}_{opt}^*$$

$$y_2^* = s \hat{\beta}_{opt}^*$$

$$s_1 \hat{y}_1^* = \hat{\beta}_{opt}^*$$

$$s_2 \hat{y}_2^* = \hat{\beta}_{opt}^*$$
Step 7 — going back

• Good thing we kept track of our variables!

Don’t lose track of your variables!

- Remember: $\beta$ is the solution, not $\beta^*$
- The green $\beta_{opt}$ is what we plug in our original expression:

$$\min_{\beta} \| y - X\beta \|^2$$
Summary of steps:

1. Think
2. Write the expression
3. \( X = USV_t \)
4. Rotate into S-space
5. “Trim” to relevant expression
6. Solve for the combination of \( \beta \)s that minimize the expression
7. Rotate back
8. Fit your data
9. Assess your work:
   1. Plot it
   2. Calculate the error
Summary of steps:

1. Think
2. Write the expression
3. \( X = USV^t \)
4. Rotate into S-space
5. “Trim” to relevant expression
6. Solve for the combination of \( \beta \)s in the expression
7. Rotate back
8. Fit your data
9. Assess your work:
   1. Plot it
   2. Calculate the error
Different orders:
1. Think
2. Write the expression
3. \( X = USV_t \)
4. Rotate into S-space
5. “Trim” to relevant expression
6. Solve for the combination of \( \beta \)s that minimize the expression
7. Rotate back
8. Fit your data
9. Assess your work:
   1. Plot it
   2. Calculate the error
Reducing dimensionality

- Complicated to work with high dimensional data
- Reducing dimensionality is good
- Example for visualization: 2d —> 1d

Check out the animation! http://setosa.io/ev/principal-component-analysis/
Reducing dimensionality

- Complicated to work with high dimensional data
- Reducing dimensionality is good
- Example for visualization: 2d $\rightarrow$ 1d

![2D data](original_data_set)

**Project onto pc1**

Not much use for 2d...
Let’s use this concept in higher dimensions

![1D data](pc1)

12 Neurons — intro

- Recording from 12 neurons simultaneously
- Record 150 observations

<table>
<thead>
<tr>
<th></th>
<th>n1</th>
<th>n2</th>
<th></th>
<th>n12</th>
</tr>
</thead>
<tbody>
<tr>
<td>obs1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>obs2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>..</td>
<td></td>
</tr>
<tr>
<td>obs150</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Reducing dimensionality

- Low dim representation of the 12 neurons
- OR low dim representation of the 150 observations
Thinking... it’s important

- What would it mean to reduce the dimensionality of the 12 neurons?
- Simple ideal example:

\[
\begin{align*}
&n_1, n_2, \ldots, n_6 \\
n_7, n_8, \ldots, n_{12}
\end{align*}
\]

And

\[
\begin{align*}
&n_1, n_2, \ldots, n_6 \\
n_7, n_8, \ldots, n_{12}
\end{align*}
\]
Thinking... it's important

- What would it mean to reduce the dimensionality of the 12 neurons?
- Simple ideal example:

At some time point:

\[
\begin{align*}
\text{n1, n2, \ldots n6} & \quad \uparrow \\
\text{n7, n8, \ldots n12} & \quad \downarrow
\end{align*}
\]

And

\[
\begin{align*}
\text{n1, n2, \ldots n6} & \quad \downarrow \\
\text{n7, n8, \ldots n12} & \quad \uparrow
\end{align*}
\]
Thinking... it's important

- What would it mean to reduce the dimensionality of the 12 neurons?
- Simple ideal example:

At some other time point:

```
| n1, n2, ... n6  |
| n7, n8, ... n12 |
```

And

```
| n1, n2, ... n6  |
| n7, n8, ... n12 |
```

Activity
Thinking... it’s important

- What would it mean to reduce the dimensionality of the 12 neurons?
- Simple ideal example:

All you need is 1D:

```
n1, n2, ... n6
n7, n8, ... n12
```

And

```
n1, n2, ... n6
n7, n8, ... n12
```

Activity
Thinking… it’s important

- What would it mean to reduce the dimensionality of the 12 neurons?
- Simple ideal example:

All you need is 1D:

Activity

n1, n2,                  …               n12

n1, n2, … n6
n7, n8, … n12
And
n1, n2, … n6
n7, n8, … n12
Thinking... it’s important

- What would it mean to reduce the dimensionality of the 12 neurons?
- Simple ideal example:

We can say...
- Neurons 1:6 are highly correlated
- Neurons 7:12 are highly correlated
- Neurons 1:6 are not at all correlated with neurons 7:12
Thinking... it’s important

• What would it mean to reduce the dimensionality of the 12 neurons?
• Simple ideal example:

We can say...
• Neurons 1:6 are highly correlated
• Neurons 7:12 are highly correlated
• Neurons 1:6 are not at all correlated with neurons 7:12

And

How does each neuron correlate with all the other 11 neurons?
Correlation across neurons:

- 12x12 table
- How can we write this in linear algebra form?
Correlation across neurons:

- 12x12 table
- How can we write this in linear algebra form?
Correlation across neurons:

- 12x12 table
- How can we write this in linear algebra form?
- MtM — covariance matrix of M
Important first step!

• What if the baseline of a neuron is higher?
• Do we care about baseline?
• Probably not… here we care about the correlation across neurons

\[ \begin{align*}
\text{obs1} & \quad \text{obs2} \\
\vdots & \quad \vdots \\
\text{obs150} & \\
\end{align*} \]
Important first step!

- Let’s center the data:
- Subtract mean of each column
What would you do if you want to cluster the 150 observations?

- How would you center the data?
- How would you get the covariance matrix?
Back to neurons

- Want a lower dimensional representation:
  - How many dimensions? Which dimensions?
- Ready for eigen decomposition of $Mt^*M$
- $\text{eig}(MtM) = V \cdot \text{lambda} \cdot V^t$
- Where $V$ is an orthogonal matrix, $V^t$ is the transpose of $V$
- Lambda is a diagonal matrix

$$MtM = \begin{bmatrix} V & \Lambda & V^t \end{bmatrix}$$

- Orthogonal
- Diagonal
- Orthogonal
Back to neurons

- Want a lower dimensional representation:
  - How many dimensions? Which dimensions?
- Ready for eigen decomposition of $Mt*M$
- $\text{eig}( MtM ) = V * \lambda * V^t$
- Where $V$ is an orthogonal matrix, transpose of $V$
- $\Lambda$ is a diagonal matrix

This must look familiar...

$$MtM = \begin{bmatrix} V & \Lambda & V^t \end{bmatrix}$$

- Orthogonal
- Diagonal
- Orthogonal
SVD vs. Eigen decomp

- Clear similarities (decomposing a matrix into 3, 2 ortho 1 diagonal)
- SVD can decompose any matrix — Eigen can’t
  - Matrix must be square for Eigen
- Eigen decomposition is a special case of SVD

\[
M = U S V^t
\]

\[
M^t M = V \Lambda V^t
\]
SVD vs. Eigen decomp

- You can get to Eigen decomposition of $M^t M$ through the SVD of $M$:

  Define symmetric matrix:

  $$
  C = M^T M \\
  = (USV^T)^T (USV^T) \\
  = VS^T U^T USV^T \\
  = V (S^T S) V^T
  $$

$$
M^t M = \begin{bmatrix}
\text{Orthogonal} \\
\text{Diagonal} \\
\text{Orthogonal}
\end{bmatrix}
$$
SVD vs. Eigen decomp

• Columns of V are the Eigen vectors of Mt*M
• What about the S matrix from svd(M)? How does it relate to Eigen decomposition?

Define symmetric matrix:

\[ C = M^T M \]
\[ = (USV^T)^T (USV^T) \]
\[ = V S^T U^T U S V^T \]
\[ = V (S^T S) V^T \]

• “rotate, stretch, rotate back”
• matrix C “summarizes” the shape of the data
Eigen decomp

• Eigen vector 1 (e1) gets scaled by eigenvalue $s_1^2$

\[
\vec{u}_k, \text{ the } k\text{th columns of } V, \\
\text{is called an } \textit{eigenvector} \text{ of } C: \\
C\vec{u}_k = V(S^T S)V^T \vec{u}_k \\
= V(S^T S)\hat{e}_k \\
= s_k^2 V\hat{e}_k \\
= s_k^2 \vec{u}_k
\]

• output is a rescaled copy of input
• scale factor $s_k^2$ is called the \textit{eigenvalue} associated with $\vec{u}_k$
Reducing dimensionality

- What reduced dimensionality do we want?
Reducing dimensionality

- Project data onto e1 — get most variance
- Project data onto e3 (small eigenvalue in this example) — get much less variance.

**Visualizing variance of PC1**

**Visualizing variance of PC3**
Reducing dimensionality

- In this example, because eigenv1 and eigenv2 >> eigenv3 and the others, we can plot our data in 2 dimensions.
Plotting the Eigenvectors:

- Spike plot of 12d vectors:
- What does each element/index represent?
Reducing dimensionality

- Complicated to work with high dimensional data
- Reducing dimensionality is good
- Example for visualization: 2d $\rightarrow$ 1d

Check out the animation! http://setosa.io/ev/principal-component-analysis/
Summary

Data: 150x12 matrix of 12 neurons
1. Think about row and columns
2. What do you want to know?
3. Center data
4. Compute the right covariance matrix
5. Eigen decomposition
6. Visualize S values — choose reduced dimensionality
7. Answer questions about your data:
   1. Plot it in the reduced dimensionality — think about what this means
   2. Plot eigenvectors — think about what they mean
Happy Friday — the weekend is so close!