Prof Anant Sahai

Today: Finish Movie Model

"Checking your work" - validating with the data

Dimensionality Reduction for classification - happening in lab!

Nonlinear Models & Local Approximation:
Control around equilibria

Illustrative Conceptual Example: Movie Ratings.

Illustrative Conceptual Example: Movie Ratings.

\[ R = \begin{bmatrix}
  r_{i1} & r_{i2} & \cdots & r_{in} \\
  r_{21} & r_{22} & \cdots & r_{2n} \\
  \vdots & \vdots & \ddots & \vdots \\
  r_{m1} & r_{m2} & \cdots & r_{mn}
\end{bmatrix} \]

\[ r_{ij} \text{ rating of person } j \text{ for movie } i \]

\[ r_{ij} \in \mathbb{R} \]

Real number score.

16 - philosophy: Make a model.

Learn the model from data.

Use the model to make predictions.

Can modify our model:

Each movie has a vector \( \vec{u} \) of qualities associated.

Each person has a vector \( \vec{s} \) of sensitivities.

SVD gave us: (Inner Product Form)

\[ R = \sum_{k=1}^{K} \sigma_k \vec{u}_k \vec{v}_k^T \]

\[ r_{ij} = \sum_{k=1}^{K} \sigma_k \vec{u}_k^T \vec{v}_i \]

\[ \text{Perf} \quad \text{items} \]

\[ R = \sum_{k=1}^{K} \sigma_k \vec{u}_k \vec{v}_i^T \]

Now our learned model has:

\[ \vec{y} = \begin{bmatrix}
  \vec{u}_1^T \\
  \vec{u}_2^T \\
  \vdots \\
  \vec{u}_K^T
\end{bmatrix}
\]

Recall \( \vec{w}'s \) are \( m \) long.

\( \vec{s}'s \) are \( n \) long.

\[ \vec{w} \]

Recall \( \vec{w}'s \) are \( m \) long.
How to use this for prediction?

E.g. Given a new person, how can we extract their sensitivities? Assume we’re given $\mathbf{r}^2$: how does this person rate the movies?

We need to go from $\mathbf{r}^2$ to $\mathbf{S}$ - estimated sensitivities on known movies for this person.

Can do this by least squares?

Aside: Let’s look at a person we already know.

For the $i^{th}$ person: $\mathbf{r}_i = \sum_k \sigma_k \mathbf{u}_k (\mathbf{E}^T)_k$.

This is a scalar - the position in vector $\mathbf{r}_i$.

How can I get $\sigma_j$ from $\mathbf{r}_j$?

$$(\sigma_j^2) = \frac{1}{\sigma_j} \mathbf{u}_j^T \mathbf{r}_j$$

- I can get the $k^{th}$ sensitivity by projecting $\mathbf{r}_j$ onto the $k^{th}$ principal component $(\mathbf{E}_k)$ and scaling by $\frac{1}{\sigma_j}$ to fit in model.

Similarly, we can get $\sigma_j$ from $\mathbf{r}_j$: $\mathbf{r}_j = [r_1, r_2, \ldots, r_m]$.

$$b > (\sigma_j) = \frac{1}{\sigma_j} \mathbf{u}_j^T \mathbf{r}_j$$

$\Rightarrow$ So for a new person with ratings $\mathbf{r}$,

Can set $\mathbf{S} = \begin{bmatrix} 1/\sigma_1 & \mathbf{E}_1^T \mathbf{r} \\ \mathbf{E}_2^T \mathbf{r} & \vdots \\ \mathbf{E}_m^T \mathbf{r} & \vdots \end{bmatrix}$

Note: Dividing by $\sigma_j$ is a warning flag: beware of small $\sigma_j$.

And similarly for a new movie $\mathbf{S} = \begin{bmatrix} 1/\sigma_1 & \mathbf{E}_1^T \mathbf{r} \\ \mathbf{E}_2^T \mathbf{r} & \vdots \\ \mathbf{E}_m^T \mathbf{r} & \vdots \end{bmatrix}$.

Can mess things up.

So, given a new person (who comes with ratings $\mathbf{r}$ on the $m$ known movies),
and a new movie (which comes with ratings by the known people)

we can predict the rating as:

\[ r = \sum_{k=1}^{n} \frac{1}{\sigma_k} \hat{e}_k \]

Fit as above.

**Question:** How do we check our work?

E.g. How do we justify that 4 is the right number of components to use? (e.g., qualities are important.

From Model-order selection problem on the HW, we know we need to use some extra data.

- **Train:** people
- **People:** hold-out people
- **Train:** movies
- **Movies:** hold-out movies

1. Fit model on training data: i.e., the SVM of that purple block, etc.
2. Test it out on held-out data:
   - Use data to estimate goodness for held-out movies.
   - Use test to check quality of predictions.

   Compare predictions to actual ratings (use Mean-Squared Error)

This procedure, sweeping through candidate sets of components, is called model order choice.

**Recall:** This is the “silver standard” (like a “unit test” for) learning from data.

The Gold Standard is always an “integration test” or “practical test”

See if it works in practice.
Step back to classification.

Want to reduce the dimensions of data.

Can use the PCA approach: Project onto \( w_i \) if data is columns or \( v_i \) if data is rows.

Heuristic: Based on hope that accurately capturing the pattern in the data helps with classification.

By reducing the dimensions, we get an easier problem.

In particular, in 2d or 3d can visualize.

Example: Three categories +, X, O

First & Most Basic Approach: Which group is the new point closest to?

An approach: Compute the centers of each group, and assign distortion of center.

LAB will do this approach.

How can we justify our approach to learn the pattern by SVD?

We are viewing \( \mathbf{R} = \sum_{i=1}^{k} \sigma_i \mathbf{u}_i \mathbf{v}_i^T \) as an approximation to \( \mathbf{R} \).

In what sense is this a good approximation?
$\mathbb{R}^c$ certainly has its column all in an $l$-dim subspace.

So $\mathbb{R}^c$ is "simple" a low-dimensional.

Is $\mathbb{R}^c$ close to $\mathbb{R}$?

We will use a favorite squared error.

We want $\| \mathbb{R} - \hat{\mathbb{R}} \|_F^2$ to be as small as possible.

Frobenius Norm

Treating $\mathbb{R}$ and $\hat{\mathbb{R}}$ like bi vectors.

Underlying model:

\[
\text{Rinv} = \left[ \frac{e_i}{k_{oi}} \sigma_e (\frac{g}{b_{o,i}}), (5_{.j}) \right] + \sigma_i \text{wij}
\]

Hope fully small.

Key Question: Why does it help?

\[
\| \mathbb{R} - \hat{\mathbb{R}} \|_F = \| \mathbb{U} \Sigma \mathbb{V}^T - \hat{\mathbb{R}} \|_F
\]

Preacttive

\[
\frac{b_{o,i}}{\Sigma}
\]

\[
\| \mathbb{U} \Sigma \mathbb{V}^T - \mathbb{U} \hat{\mathbb{R}} \|_F
\]

Transposing

\[
\| \mathbb{V} \Sigma \mathbb{T} - \mathbb{R} \mathbb{U} \|_F
\]

Preacttive

\[
\frac{b_{o,i}}{\Sigma}
\]

\[
\| \mathbb{V} \Sigma \mathbb{T} - \mathbb{V} \hat{\mathbb{R}} \mathbb{U} \|_F
\]

Transposing

\[
\| \mathbb{V} - \mathbb{U} \hat{\mathbb{R}} \mathbb{V} \|_F
\]

Still a rank-$l$ matrix.
Since \( R = \sum_{k=1}^{\min(r,s)} \sigma_k (U^k \bar{g}_k)^{T} \)

\[ U^TRV = \sum_{k=1}^{\min(r,s)} \sigma_k (U^k \bar{g}_k) (U^k \bar{g}_k)^{T} \]

just a vector.

\[ \Rightarrow U^TRV \text{ has rank at most } k. \]

So, \( U^TRV \) should be the best rank- \( k \) approximation.

Intuitively clear! Should just keep the \( k \) biggest \( \sigma \)s.

Want \( U^k \bar{g}_k \) to be the \( k \)-th column of the identity. So \( \bar{g}_k = \bar{g}_k \). Similarly we want \( U^k \bar{g}_k \) to be the \( k \)-th row of the identity. So \( \bar{g}_k = \bar{g}_k \).

The full theorem is called the Eckart–Young–Mirsky Theorem.

\[ \Rightarrow \text{The best rank-} k \text{ approximation to a matrix} \]

is given by keeping the first \( k \) terms in the SVD.

---

Step Way Back: All of our 16A/B models have been linear except switches/comparators \( \in \text{nonlinear} \).

Real World is largely nonlinear. \( G \in \text{nonlinear} \).

Almost all mechanical systems are nonlinear.
Consider a pendulum:

\[ \begin{align*}
\text{e.g.: Get sines & cosines in equations governing motion} \\
\text{Need a tool & philosophy for attacks nonlinear problems}
\end{align*} \]

**Tool:** Local Approximation: Just pretend they are linear.

**Taylor Series:**
\[ f(x) = f(x_0) + f'(x_0)(x-x_0) \]
\[ + \frac{f''(x_0)}{2!}(x-x_0)^2 \]
\[ + \ldots \]

When \( x \) is close to \( x_0 \), the higher-order terms contribute very little.

\[ \Rightarrow \text{lump all dropped terms into the disturbance} \]

Consider:
\[ f(x, u) = 32 + x^3 u^2 \]

expand around \((x_0, u_0)\)

\[ f(x_0 + \delta x, u_0 + \delta u) = 32 + (x_0 + \delta x)^3 (u_0 + \delta u)^2 \]

\[ = 32 + (x_0^3 + 3x_0^2 \delta x + 3x_0 \delta x^2 + \delta x^3) \]
\[ \cdot (u_0^2 + 2u_0 \delta u + \delta u^2) \]

If \( \delta x \) and \( \delta u \) are both tiny, then \((\delta x)(\delta u)\)

\[ \approx \delta x^2, \delta u^2, \ldots \]

All higher powers are extremely small.

\[ f(x_0 + \delta x, u_0 + \delta u) \approx 32 + x_0^3 u_0^2 + \frac{3x_0^2 u_0^2}{2} \delta x \]
\[ + \frac{3x_0 u_0}{2} \delta u \]
\[ \Rightarrow \text{lump this into disturbance} \]

This gives an approximate "linear" model in the neighborhood of \((x_0, u_0)\).

\[ \text{Actually "affine" since there is} \]
\[ \text{the constant term in front} \]