Last time

• Basic concepts
  – Loss functions
  – Estimation vs inference
  – Decision boundaries
  – Overfitting
  – Regularization
  – Model selection
  – Structural error vs approximation error
This time

- Basis functions
- Normal equations
- QR
- SVD
$$p(y|x, \theta) = \mathcal{N}(y|w^T x, \sigma^2)$$
Polynomial Regression

\[ f(x) = w_0 + w_1 x + w_2 x^2 + \cdots + w_d x^D \]

\[ f(x) = \mathbf{w}^T \phi(x) = \sum_{j=1}^{d} w_k \phi_j(x) \]

Line denotes posterior mode \( \arg \max_y p(y|x) \)

Error bars denote 95% credible interval \( p(y \in I|x) = 0.95 \)
Polynomial Regression

\[ f(x) = w^T \phi(x) = \sum_{j=1}^{d} w_k \phi_j(x) \]

\[ f(x) = w_0 + w_1 x_1 + w_2 x_2 \]

\[ f(x) = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_1^2 + w_4 x_2^2 \]

\[ f(x) = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_1^2 + w_4 x_2^2 + w_5 x_1 x_2 \]

Interaction term
Polynomial basis

- Linear regression can fit nonlinear functions, provided the nonlinearity is fixed

$$\Phi = \begin{pmatrix} 1 & x_1 & x_1^2 & x_1^3 \\ \vdots \\ 1 & x_n & x_n^2 & x_n^3 \end{pmatrix}$$
Radial basis functions (RBF)

- Measure distance to examplars

\[ \phi(x) = [K(x, \mu_1), \ldots, K(x, \mu_d)], \quad K(x, \mu) = \exp\left(-\frac{\|x - \mu\|^2}{2\sigma^2}\right) \]
RBF vs polynomials
Categorical features

- Not meaningfully ordered, so use 1-of-K encoding to embed into a vector space

\[ \phi(x) = [I(x = r), I(x = g), I(x = b)] \]
\[ \phi(x) = [1, I(x = r), I(x = g)] \]
\[ p(y|x, \theta) = \mathcal{N}(y|w_0 + w_1 I(x = r) + w_2 I(x = g), \sigma^2) \]
\[ E(y|x = r, \theta) = w_0 + w_1, \quad E(y|x = g, \theta) = w_0 + w_2, \quad E(y|x = b, \theta) = w_0 \]
Standardization

- Often need to ensure features are on same scale (numerics, ridge)

\[
z_{ij} = \frac{x_{ij} - \bar{x}_j}{\sigma_j}
\]
Listing 1:

```
%Part of codename{linregDist.demoPolyfitDegree}
m = linregDist;
m.transformer = chainTransformer({rescaleTransformer, polyBasisTransformer(degree)});
m = fit(m, 'X', xtrain, 'y', ytrain);
ypredTest = predict(m, xtest);
testMse = mean((ypredTest - ytest).^2);
```
MLE for linear regression (least squares)

\[
p(D|w, \sigma^2) = \prod_{i=1}^{n} \mathcal{N}(y_i|w^T x_i, \sigma^2)
\]
\[
= (2\pi\sigma^2)^{-\frac{n}{2}} \exp \left( -\frac{1}{2\sigma^2} (y - Xw)^T(y - Xw) \right)
\]
\[
J(w, \sigma^2) = -\log p(y|X, w, \sigma^2) \quad \text{Negative log likelihood}
\]
\[
= \frac{n}{2} \log(\sigma^2) + \frac{1}{2\sigma^2} RSS(w)
\]
\[
RSS(w) = ||Xw - y||_2^2 = (y - Xw)^T(y - Xw)
\]
Normal equations

\[ \nabla_w RSS(w) = 0 \]

See book for derivation

\[ \hat{w} = (X^T X)^{-1} X^T y = \left( \sum_{i=1}^{n} x_i x_i^T \right)^{-1} \left( \sum_{i=1}^{n} y_i x_i \right) \]

MLE = OLS estimate

Uncertainty in estimate – see later
Geometry of least squares

\[ X = \begin{pmatrix} 1 & 1 \\ 1 & -1 \\ 0 & 0 \end{pmatrix}, \quad y = \begin{pmatrix} 3 \\ 2 \\ 3 \end{pmatrix} \]

Minimize RSS by orthogonal projection of \( y \) into column space of \( X \)

\[ \hat{y} = w_1 x_{:,1} + \cdots + w_d x_{:,d} \]

Final Error = \( (Xw - y) \)
Orthogonal projection

• Projection of $y$ onto $X$

\[
\text{Proj}(y; X) = \text{argmin}_{\hat{y} \in \text{span}\{x_1, \ldots, x_n\}} \| y - \hat{y} \|_2.
\]

• Let $r = y - \hat{y}$. Residual must be orthogonal to $X$. Hence

\[
x_j^T (y - \hat{y}) = 0 \Rightarrow X^T (y - Xw) = 0 \Rightarrow w = (X^T X)^{-1} X^T y
\]

• Prediction on training set

\[
\hat{y} = X\hat{w} = X(X^T X)^{-1} X^T y \overset{\text{def}}{=} Hy
\]

Hat matrix

• Residual is orthogonal

\[
X^T (y - Hy) = X^T (y - X\hat{w}) = X^T y - X^T X(X^T X)^{-1} X^T y = 0
\]
Solving for offset

• Let us separate $w_0$ from the other weights

$$J(w, \hat{w}_0) = \sum_{i=1}^{n} (y_i - x_i^T w - w_0)^2$$

• One can show (homework) that

$$\hat{w}_0 = \frac{1}{n} \sum_i y_i - \frac{1}{n} \sum_i x_i^T w = \bar{y} - \bar{x}^T w$$

• And

$$\hat{w} = (X_c^T X_c)^{-1} X_c^T y_c = \left[ \sum_{i=1}^{n} (x_i - \bar{x}) (x_i - \bar{x})^T \right]^{-1} \left[ \sum_{i=1}^{n} (y_i - \bar{y}) (x_i - \bar{x}) \right]$$

• For 1d data:

$$w_1 = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sum_i (x_i - \bar{x})^2} = \frac{\sum_i x_i y_i - n \bar{x} \bar{y}}{\sum_i x_i^2 - n \bar{x}^2}$$
$$w_0 = \bar{y} - w_1 \bar{x}$$
Solving for $\sigma^2$

- One can show

$$\hat{\sigma}^2 = \frac{1}{n}(y - X\hat{w})^T(y - X\hat{w}) = \frac{1}{n} \sum_{i=1}^{n} (y_i - x_i^T\hat{w})^2$$
Colinearity

- Consider if $x_1 = x_2$

\[ w_1 x_1 + w_2 x_2 = (w_1 + w_2)x_1 = (w_1 + w_2)x_2 \]

What solution should we return?
Null space

- Consider rank 2 matrix (2nd = avg of 1 + 3)

\[
X = \begin{pmatrix}
1 & 2 & 3 \\
4 & 5 & 6 \\
7 & 8 & 9 \\
10 & 11 & 12 \\
13 & 14 & 15 \\
\end{pmatrix}, \quad y = \begin{pmatrix}
16 \\
17 \\
18 \\
19 \\
20 \\
\end{pmatrix}
\]

- Let \( z \) be in the null space of \( X \), ie \( Xz = 0 \). Then

\[
Xw = y \Rightarrow X(w + cz) = y
\]

\[X = \text{reshape}(1:15, [3 5])'; \quad y = (16:20)';\]
\[w = X\backslash y; \quad z = [1;-2;1];\]
\[c = \text{rand}; \quad \text{assert(approxeq(norm(X*(w+c*z) - y), 0))}\]

- What solution should we return?
Condition number

• Suppose X is full rank so solution is theoretically unique. May be hard to find numerically.

\[ X = \begin{pmatrix} 1 & 1 \\ \delta & 0 \\ 0 & \delta \end{pmatrix}, \quad X^TX = \begin{pmatrix} 1 + \delta^2 & 1 \\ 1 & 1 + \delta^2 \end{pmatrix}, \quad \kappa(X^TX) = \kappa(X)^2 \]

• We see methods for finding the MLE that do not invert \( X^T X \)

• Each method will resolve the ambiguity issue in a different way
QR decomposition

- We find a set of orthonormal vectors $q_j$ that span successive columns of $X$ (using Gram-Schmidt orthogonalization)

\[
\begin{align*}
x_1 &= r_{11}q_1 \\
x_2 &= r_{12}q_1 + r_{22}q_2 \\
    & \vdots \\
x_n &= r_{1n}q_1 + \cdots + r_{nn}q_n
\end{align*}
\]

\[
\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} =
\begin{pmatrix} q_1 \\ q_2 \\ \vdots \\ q_n \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ r_{22} & \cdots \\ \vdots \\ r_{nn} \end{pmatrix}
\]
QR decomposition

- Can make $Q$ and $R$ be square $m \times m$ matrices so we can write $Q^T Q = QQ^T = I$

$$
\begin{pmatrix}
  x_1 \\
  x_2 \\
  \vdots \\
  x_n
\end{pmatrix} =
\begin{pmatrix}
  q_1 \\
  q_2 \\
  \vdots \\
  q_n
\end{pmatrix}
\begin{pmatrix}
  r_{11} & r_{12} & \cdots & r_{1n} \\
  0 & r_{22} & \cdots & \vdots \\
  \vdots & \vdots & \ddots & \vdots \\
  0 & 0 & \cdots & r_{nn}
\end{pmatrix}
$$
Least squares with QR

• We have

\[ \hat{w} = (X^T X)^{-1} X^T y \]
\[ = (R^T Q^T QR)^{-1} R^T Q^T y \]
\[ = (R^T R)^{-1} R^T Q^T y \]
\[ = R^{-1} R^{-T} R^T Q^T y \]
\[ = R^{-1} Q^T y \]

• Let \( z = Q^T y \). Solve \( w = R^{-1} z \) by back substitution, \( w = R \backslash z \).

\[
\begin{align*}
[Q, R] &= \text{qr}(X, 0); \\
w &= R \backslash (Q' * y); \\
\text{Shorthand} & \quad w = X \backslash y;
\end{align*}
\]
Basic solution

- Let $r = \text{rank}(X)$. Basic solution has $r$ non-zeros.
- $w=X\backslash y$ returns one of many possible basic solutions.

\[
X = \begin{pmatrix}
1 & 2 & 3 \\
4 & 5 & 6 \\
7 & 8 & 9 \\
10 & 11 & 12 \\
13 & 14 & 15
\end{pmatrix}, \quad y = \begin{pmatrix}
16 \\
17 \\
18 \\
19 \\
20
\end{pmatrix}
\]

\[
X = \text{reshape}(1:15, [3 5])'; \quad y = (16:20)';
\]

\[
w = X\backslash y \% [-7.5, 0. 7.83]
\]

\[
norm(X*w - y) \% 0.00
\]

\[
w = [0,-15,15.3333]';
\]

\[
norm(X*w - y) \% 0.00
\]
\[ A = U \Sigma V^T = \sigma_1 \begin{pmatrix} u_1 \end{pmatrix} \begin{pmatrix} - v_1^T \end{pmatrix} + \cdots + \sigma_r \begin{pmatrix} u_r \end{pmatrix} \begin{pmatrix} - v_r^T \end{pmatrix} \]

\[ U^T U = I \]
\[ V^T V = VV^T = I \]
Truncated SVD

- Rank k approximation to a matrix

\[ A_k = \sum_{j=1}^{k} \sigma_j u_j v_k^T = U_{:,1:k} \Sigma_{1:k,1:k} V_{:,1:k}^T \]

Equivalent to PCA
load clown; % built-in image
[U, S, V] = svd(X, 0);
k = 20;
Xhat = (U(:,1:k)*S(1:k,1:k)*V(:,1:k)');
image(Xhat);
SVD for least squares

- We have

\[
\hat{w} = (X^TX)^{-1}X^Ty \\
X^TXw = X^Ty \text{ (premultiply by } X^TX) \\
VDU^T UDV^T w = VDU^T y \text{ (SVD expansion)} \\
VD^2V^T w = VDU^T y \text{ (since } U^TU = I \text{ and } DD = D^2) \\
D^2V^T w = DU^T y \text{ (premultiply by } V^T) \\
V^T w = D^{-1}U^T y \text{ (premultiply by } D^{-2}) \\
w = VD^{-1}U^T y \text{ (premultiply by } V) \\
\]

```
[U,D,V]=svd(X,0);
Dinv = diag(1./(diag(D)));
w = V*Dinv*U'*y;
```

What if \( D_j = 0 \) (so rank of \( X \) is less than \( d \))?
If $D_j=0$, use

$$w = VD^\dagger U^T y \overset{\text{def}}{=} X^\dagger y, \quad D^\dagger = \text{diag}(\sigma_1^{-1}, \ldots, \sigma_r^{-1}, 0, \ldots, 0)$$

```matlab
function B = pinv(A)
[U, S, V] = svd(A, 0);
s = diag(S);
r = sum(s > tol); \% rank
w = diag(ones(r, 1) ./ s(1:r));
B = V(:,1:r) * w * U(:,1:r)';
```

Of all solutions $w$ that minimize $||Xw - y||$, the pinv solution also minimizes $||w||$

```matlab
w = X\y;
w2 = pinv(X)*y;
[norm(w) norm(w2)]
>> 10.8449  10.8440
```