03/07/2016

Ridge Regression

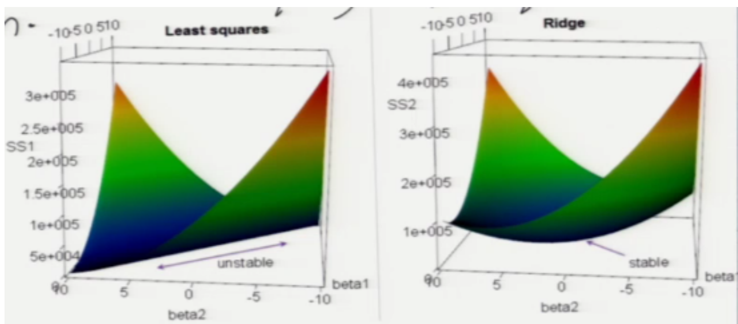
(aka Tikhonov regularization)

- (1) + (A) + ℓ_2 penalized mean loss (d).
- Optimization problem $J(w)$:

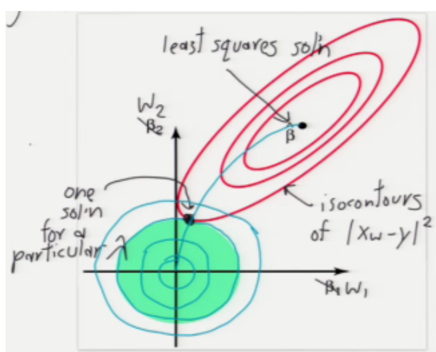
$$\boxed{\text{Find } w \text{ that minimizes } |Xw - y|^2 + \lambda||w'||^2}$$

  Where w' is w with components α replaced by 0. X has fictitious dimension but we DON'T penalize α.

- Adds a penalty term to encourage small $|w'|$ – called shrinkage.
- Why? Guaranteed positive definite normal equations; always unique solution. e.g. when $d > n$ always semi-definite.



- Reduces over-fitting by reducing variance by penalizing large weights.

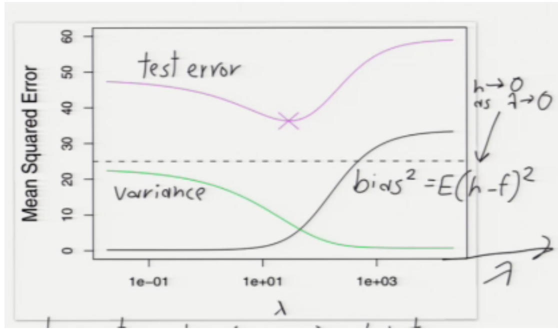


- Setting $\nabla J = 0$ gives equations,

$$(X^T X + \lambda I')w = X^T y$$

- where I' is identity matrix with bottom right set to zero.
- Algorithm: solve for w. Return $h(z) = w^T x$.
- Increasing $\lambda \Rightarrow$ more regularization; smaller $|w'|$.
- Given our data model $y = Xv + e$, where e is noise.
- Variance of ridge regression is $\text{Var}(x^T(X^T X + \lambda I')^{-1} X^T e)$.

- As $\lambda \to \infty$, variance $\to 0$, but bias increases.



- λ is a hyper-parameter; tune by (cross)-validation.
- Ideally features should be "normalized" to have same variance.
- Alternative: Use asymmetric penalty by replacing I' with other diagonal matrix.

Bayesian justification for ridge regression

- Assign a prior probability on w': a Gaussian centered at 0.
- Posterior probability $\approx$ likelihood of $w\cdot$ prior $P(w') \leftarrow$ Gaussian PDF.
- Maximize the log posterior, ℓn likelihood + $\ell n P(w')$ = -const$|Xw - y|^2$ - const$|w'|^2$ - constant.
- This method (using likelihood, but maximizing posterior) is called maximum a posteriori (MAP).

Kernels

- Recall: with d input features, degree-p polynomials blow up to $\mathcal{O}(d^p)$ features.
- Today we use magic to use those features without computing them!
- Observation: In many learning algorithms:
  - The weights can be written as a linear combination of input samples.
  - We can use inner products of $\phi(x)'$s only $\Rightarrow$ don't need to compute $\phi(x)$!
  - Suppose $w = X^T a = \sum_{i=1}^{n} a_i X_i$ for some $a \in \mathbb{R}^n$.
  - Substitute this identity into algorithms and optimize n dual weights (aka dual parameters a, instead of $d+1$ primal weights w.

Kernel Ridge Regression

- Center X and y so their means are zero; $X_i \leftarrow X_i - \mu_x$. By centering the matrix we minimize the penalization of α
- This lets us replace I' with I in normal equations:

$$(X^T X + \lambda I)w = X^T y$$
$$\Rightarrow w = \frac{1}{\lambda}(X^T y - X^T X w) = X^T a \qquad \text{where } a = \frac{1}{\lambda}(y - Xw)$$

- This shows that w is a linear combination of samples. To compute a:

$$\lambda a = (y - XX^T a) \Rightarrow a = (XX^T + \lambda I)^{-1} y$$

- a is the dual solution; solves the dual form of ridge regression:

$$\boxed{\text{Find } a \text{ that minimizes } |XX^T - y|^2 + \lambda |X^T a|^2}$$

- Regression function is:

$$h(z) = w^T z = a^T X z = \sum_{i=1}^{n} a_i (X_i^T z) \Leftarrow \text{weighted sum of inner products}$$

- Let $k(x, z) = x^T z$ be kernel function.
- Let $K = XX^T$ be nxn kernel matrix. Note $K_{ij} = k(X_i, X_j)$.
- K is singular if $n > d$. In that case no solution if $\lambda = 0$.
- Summary of kernel ridge regression:
  - Solve $(K + \lambda I)a = y$ for a $\Leftarrow \mathcal{O}(n^3)$ time.
  - $K_{ij} = k(X_i, X_j)\ \forall i, j \Leftarrow \mathcal{O}(n^2 d)$ time.
  - for each test point z: $h(z) = \sum_{i=1}^{n} a_i k(X_i, z) \Leftarrow \mathcal{O}(nd)$ time.
- Do not use X directly: only $k(\cdot, \cdot)$.
- Dual: solve nxn linear system.
- Primal: solve dxd linear system.

The Kernel Trick

(aka kernelization)

- The polynomial kernel of degree p is $k(x, z) = (x^T z + 1)^p$.
- Theorem: $(x^T z + 1)^p = \phi(x)^T \phi(z)$ where $\phi(x)$ contains every monomial in x of degree $0 \ldots p$.
- Example for $d = 2, p = 2$.

$$\begin{aligned}
(x^T z + 1)^2 &= x_1^2 z_1^2 + x_2^2 z_2^2 + 2x_1 z_1 x_2 z_2 + 2x_1 z_1 + 2x_2 z_2 + 1 \\
&= \begin{bmatrix} x_1^2 & x_2^2 & \sqrt{2x_1 x_2} & \sqrt{2x_1} & \sqrt{2x_2} & 1 \end{bmatrix} \cdot \begin{bmatrix} z_1^2 \\ z_2^2 \\ \sqrt{2z_1 z_2} \\ \sqrt{2z_1} \\ \sqrt{2z_2} \\ 1 \end{bmatrix} \\
&= \phi(x)^T \phi(z)
\end{aligned}$$

- Key win: compute $\phi(x)^T \phi(z)$ in $\mathcal{O}(d)$ time instead of $\mathcal{O}(d^p)$ even though $\phi(x)$ has length $\mathcal{O}(d^p)$.
- Kernel ridge regression replaces X_i with $\phi(X_i)$:
  - Let $k(x, z) = \phi(x)^T \phi(z)$.