Regression
Chapter 10 of *Foundations of Machine Learning*, Pt. II

Shashank Singh

Carnegie Mellon University
11-745 Spring ’15

April 20, 2015
1 Regression Framework

2 More Algorithms
   - Support Vector Regression
   - Lasso

3 Online Algorithms
   - Widrow-Hoff Algorithm
   - Online Dual SVR

4 Nonparametric Algorithms
   - Local Polynomial Regression
Problem Statement

- size-\( m \) sample \( S = \{(x_1, y_1)\}_{i=1}^m \in (\mathcal{X} \times \mathcal{Y})^m \) with
  - \( x_1, \cdots, x_m \sim D \) for some unknown distribution \( D \) on \( \mathcal{X} \)
  - \( (y_1, \cdots, y_m) = (f(x_1), \cdots, f(x_m)) \) for some unknown \( f : \mathcal{X} \to \mathcal{Y} \)

- loss function \( L : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R} \)

- want to find a hypothesis \( h \) minimizing the risk

\[
R(h) = \mathbb{E}_{x \sim D} [L(h(x), f(x))] .
\]
Fitting an $\epsilon$-tube

- Rather than fitting a line to the data, we fit an $\epsilon$-tube
- Accordingly, we use the $\epsilon$-insensitive loss

$$L_\epsilon(h(x), f(x)) = \max\{0, |h(x) - f(x)| - \epsilon\}$$

- This encourages a data-sparse solution, since points with $|\hat{h}(x_i) - y_i| \leq \epsilon$ incur no loss
Support Vector Regression

- Apply feature map $\Phi : \mathcal{X} \rightarrow \mathbb{R}^k$ corresponding to kernel $K$
- Then fit a linear model:

$$\hat{f}(x) = \hat{w} \cdot \Phi(x) + \hat{b}$$

- Thus, our empirical risk is

$$\hat{R}(\hat{w}, \hat{b}) = \frac{1}{n} \sum_{i=1}^{n} L_\varepsilon(\hat{w} \cdot \Phi(x_i) + \hat{b}, y_i)$$

- Finally, we add an $\ell_2$ regularization term on $w$ and parameter $C$, giving the optimization problem:

$$\min_{w \in \mathbb{R}^k, b \in \mathbb{R}} \frac{1}{2} \|w\|_2^2 + C \frac{1}{n} \sum_{i=1}^{n} L_\varepsilon(\hat{w} \cdot \Phi(x_i) + \hat{b}, y_i)$$
Deriving the Dual Problem

- Add slack variables $\xi^-, \xi^+ \in \mathbb{R}^m$ which encode point loss:

$$\min_{w \in \mathbb{R}^k, b \in \mathbb{R}, \xi^-, \xi^+ \in \mathbb{R}^m} \frac{1}{2} \|w\|_2^2 + \frac{C}{n} \sum_{i=1}^n \xi_i^- + \xi_i^+$$

subject to

$$\hat{w} \cdot \Phi(x_i) + b \geq y_i - \varepsilon - \xi_i^-,$$

$$\hat{w} \cdot \Phi(x_i) + b \leq y_i + \varepsilon + \xi_i^+$$

- $\xi_i^-$ and $\xi_i^+$ encode loss of under/overestimating $y_i$, resp.
- From here, we can derive the dual (as with SVMs):

$$\max_{\alpha, \alpha' \in \mathbb{R}^k} - \varepsilon (\alpha' + \alpha)^T 1_k + (\alpha' - \alpha)^T y$$

$$- \frac{1}{2} (\alpha' - \alpha)^T K (\alpha' - \alpha)$$

subject to

$$0 \leq \alpha_i, \alpha'_i \leq C,$$

$$(\alpha' - \alpha)^T 1_k = 0$$
Computing SVR Solutions

- Both the primal and the dual are quadratic programs
  - can be solved via standard algorithms
- Can recover the hypothesis $h$ from $(\alpha, \alpha')$ via

  $$h(x) = b + \sum_{i=1}^{m}(\alpha' - \alpha)K(x_i, x), \quad \forall x \in \mathcal{X}$$

  where, for any $x_j$ with $\alpha_j \in (0, C)$ or $\alpha'_j \in (0, C)$

  $$b = -\sum_{i=1}^{m}(\alpha' - \alpha)K(x_i, x_j) + y_j \pm \varepsilon$$

- Thus, we can solve either the primal or the dual
- By complementarity, only points outside $\varepsilon$-tube are support vectors
- $\varepsilon$ provides trade-off between sparsity and accuracy
SVR Generalization Bounds

**Theorem 10.8:**
- $K : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ PDS kernel with feature map $\Phi : \mathcal{X} \rightarrow \mathbb{H}$
- $H := \{ x \mapsto w \cdot \Phi(x) : \|w\|_{\mathbb{H}} \leq \Lambda \}$
- $r := \sup \left\{ \sqrt{K(x, x)}, \frac{|f(x)|}{\Lambda} : x \in \mathcal{X} \right\}$

Then, w.p. $\geq 1 - \delta$,

$$R_\varepsilon(h) \leq \hat{R}(h) + \frac{2r\Lambda}{\sqrt{m}} \left( 1 + \sqrt{\frac{\log(1/\delta)}{2}} \right)$$

and

$$R_\varepsilon(h) \leq \hat{R}(h) + \frac{2r\Lambda}{\sqrt{m}} \left( \sqrt{\frac{\text{Tr}[K]}{mr^2}} + 3 \sqrt{\frac{\log(1/\delta)}{2}} \right).$$

- The theorem follows from results proven last time.
Strengths and Weaknesses of SVR

**Strengths:**
- Strong theoretical guarantees (gen. error and stability)
- Data-sparsity (assuming $\varepsilon$-insensitive loss)
- Non-linearity and expert knowledge integration via PDS kernels
- Regularization

**Weaknesses:**
- Two parameters to select ($C$ and $\varepsilon$; heuristics exist)
- Computationally expensive for large training sets
  - Can make kernel evaluation more efficient via low-rank approximations
Unlike KRR and SVR, Lasso is not kernelizable, so let

\( \mathcal{X} \subseteq \mathbb{R}^N \)

\( H = \{ x \mapsto w \cdot x + b : w \in \mathbb{R}^N, b \in \mathbb{R} \} \)

We optimize the following:

\[
\min_{w \in \mathbb{R}^N, b \in \mathbb{R}} \lambda \|w\|_1 + \sum_{i=1}^{N} (w \cdot x_i + b - y_i)^2
\]

This induces sparsity in \( w \)
For some $\Lambda_1 > 0$, this optimization problem is equivalent to

$$\min_{w \in \mathbb{R}^N, b \in \mathbb{R}} \sum_{i=1}^{N} (w \cdot x_i + b - y_i)^2 \quad \text{subject to} \quad \|w\|_1 \leq \Lambda_1$$

L1 regularization

L2 regularization
Rademacher complexity of lasso (with bounded $L_1$ norm)

**Theorem 10.10:**

- $\mathcal{X} \subseteq \mathbb{R}^N$
- $r = \max_{i\in[1,m]} \|x_i\|_\infty$
- $H := \{x \mapsto w \cdot x : \|w\|_1 \leq \Lambda_1\}$

Then,

$$\hat{\mathcal{R}}_S(H) \leq \sqrt{\frac{2r_\infty^2 \Lambda_1^2 \log(2N)}{m}}$$

- Note the logarithmic dependence on dimension $N$
- Deterministic bound on empirical Rademacher complexity of a given sample
Proof of Generalization Bound

First recall two facts:

1. **Dual norms:** For $p, q \in [1, \infty]$, if $\frac{1}{p} + \frac{1}{q} = 1$, then $\forall x \in \mathbb{R}^n$,

   $$
   \|x\|_p = \left(\sum_{i=1}^{n} x_i^p\right)^{1/p} = \sup_{\|y\|_q \leq 1} y \cdot x.
   $$

2. **Massart’s Lemma:** For $A \subseteq \mathbb{R}^n$ finite, $r := \max_{x \in A} \|x\|_2$,

   $$
   \mathbb{E}_{\sigma} \left[ \frac{1}{m} \sup_{x \in A} \sum_{i=1}^{m} \sigma_i x_i \right] \leq \frac{r \sqrt{2 \log |A|}}{m}.
   $$
Proof of Generalization Bound

Since $\| \cdot \|_1$ and $\| \cdot \|_\infty$ are dual norms,

$$\hat{R}_S(H) = \frac{1}{m} \mathbb{E}_\sigma \left[ \sup_{\|w\|_1 \leq \Lambda_1} w \cdot \sum_{i=1}^{m} \sigma_i x_i \right]$$

$$= \frac{\Lambda_1}{m} \mathbb{E}_\sigma \left[ \left\| \sum_{i=1}^{m} \sigma_i x_i \right\|_\infty \right] = \frac{\Lambda_1}{m} \mathbb{E}_\sigma \left[ \sup_{z \in A} \sum_{i=1}^{m} \sigma_i z_i \right]$$

for $A = \{ s(x_{1,j}, \cdots, x_{m,j})^T : j \in [1, N], s \in \{-1, 1\} \}$. For $z \in A$, $\|z\|_2 \leq \sqrt{mr^2_\infty} = r_\infty \sqrt{m}$. Thus, by Massart’s Lemma, since $|A| \leq 2N$,

$$\hat{R}_S(H) \leq \Lambda_1 r_\infty \sqrt{m} \sqrt{\frac{2 \log(2N)}{m}} \leq \sqrt{\frac{2r_\infty^2 \Lambda_1^2 \log(2N)}{m}}.$$
Rademacher complexity of lasso (with bounded $L_1$ norm)

**Theorem 10.11:**

- $\mathcal{X} \subseteq \mathbb{R}^N$
- $r = \sup_{x \in \mathcal{X}} \|x\|_\infty$
- $H := \{x \mapsto w \cdot x : \|w\|_1 \leq \Lambda_1\}$

Then, w.p. $\geq 1 - \delta$,

$$R(h) \leq \hat{R}(h) + \frac{8r_\infty^2\Lambda_1^2}{\sqrt{m}} \left(\sqrt{2 \log(2N)} + \frac{1}{2} \sqrt{\frac{\log(1/\delta)}{2}}\right).$$

- High prob. gen. bound for bounded domain (unlike 10.10)
- Follows from previous results, including Theorem 10.10
Strengths and Weaknesses of Lasso

**Strengths:**
- Strong theoretical guarantees (gen. error and stability)
- Fast algorithms (Lars) for computing entire *regularization path*
- Efficient online version
- Feature-sparsity

**Weaknesses:**
- Not kernelizable, and hence only linear
  - Can use empirical kernel maps
- No closed form solution
If the feature space naturally partitions into subsets, we can encourage a sparse solution that selects groups of features.

If \( w = (w_1, \cdots, w_k) \in \mathbb{R}^N \), where each \( w_i \in \mathbb{R}^{N_i} \), then the \( L_{2,1} \) norm of \( w \) is

\[
\| w \|_{2,1} = \sum_{i=1}^{k} \| w_i \|_2.
\]
Widrow-Hoff Algorithm

\[
\text{WIDROWHOFF}(w_0)
\]

1. \( w_1 \leftarrow w_0 \quad \triangleright \text{typically } w_0 = 0 \)
2. \( \text{for } t \leftarrow 1 \text{ to } T \text{ do} \)
3. \( \quad \text{RECEIVE}(x_t) \)
4. \( \quad \hat{y}_t \leftarrow w_t \cdot x_t \)
5. \( \quad \text{RECEIVE}(y_t) \)
6. \( \quad w_{t+1} \leftarrow w_t + 2\eta(w_t \cdot x_t - y_t)x_t \quad \triangleright \text{learning rate } \eta > 0. \)
7. \( \text{return } w_{T+1} \)

- Applies online gradient descent to least squares objective
- Can do the same with ridge regression and lasso
Online Dual SVR

\begin{algorithm}
\caption{OnlineDualSVR()}
\begin{algorithmic}[1]
\State $\alpha \leftarrow 0$
\State $\alpha' \leftarrow 0$
\For{$t \leftarrow 1$ to $T$}
\State $\text{receive}(x_t)$
\State $\hat{y}_t \leftarrow \sum_{s=1}^{T}(\alpha'_s - \alpha_s)K(x_s, x_t)$
\State $\text{receive}(y_t)$
\State $\alpha'_{t+1} \leftarrow \alpha'_t + \min(\max(\eta(y_t - \hat{y}_t - \epsilon), -\alpha'_t), C - \alpha'_t)$
\State $\alpha_{t+1} \leftarrow \alpha_t + \min(\max(\eta(\hat{y}_t - y_t - \epsilon), -\alpha_t), C - \alpha_t)$
\EndFor
\State return $\sum_{t=1}^{T} \alpha_t K(x_t, \cdot)$
\end{algorithmic}
\end{algorithm}

- Applies online gradient descent to dual SVR objective
Nonparametric Regression

- Want to fit
  
  \[ y = f(x) + \varepsilon, \]

  where \( f \) is smooth, but need not be linear

- Can still use penalized risk minimization approaches (e.g., splines)

- Here, we’ll mention a different method
Let $w_i(x)$ denote the weight of the data point $(x_i, y_i)$ on the prediction at $x$.

Usually, $w_i(x)$ is based on a smoothing kernel, e.g.,

$$w_i(x) = \frac{1}{\sqrt{2\pi h^2}} \exp \left( -\frac{(X_i - x)^2}{2h^2} \right),$$

where $h > 0$ is a bandwidth parameter.

Let $P_x(X; a)$ denote the order $p$ polynomial with coefficients $(a_1, \cdots, a_p)$, evaluated at $X - x$:

$$P_x(X; a) = \sum_{j=1}^{p} \frac{a_p}{p!} (X - x)^p$$
Local Polynomial Regression

- For weights $w_i(x)$ (e.g., from a smoothing kernel), minimize

$$\min_{a \in \mathbb{R}^p} \sum_{n=1}^{n} w_i(x)(Y_i - P_x(X_i; a))^2$$

- If

$$X_x = \begin{bmatrix} 1 & X_1 - x & \cdots & \frac{(X_1-x)^p}{p!} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & X_n - x & \cdots & \frac{(X_n-x)^p}{p!} \end{bmatrix}, \quad W = \text{diag}(w_1(x), \cdots, w_n(x)),$$

then

$$\sum_{n=1}^{n} w_i(x)(Y_i - P_x(X_i; a))^2 = (Y - X_x a)^T W_x (Y - X_x a),$$

and so $\hat{a}(x) = (X_x^T W_x X_x)^{-1} X_x^T W_x Y$ (closed form solution)
Nonparametric Theory

- Rademacher complexity is usually infinite for nonparametric hypothesis classes
- Rather than generalization bounds, results are usually bounds on bias and variance of $\hat{f}$, either at a point or integrated
- Bias bounds depend crucially on smoothness assumptions on true $f$, e.g.

$$L = \int_{\mathbb{R}} (f^{(\beta)}(x))^2 \, dx < \infty,$$

where $\beta \in \mathbb{N}$ and $f^{(\beta)}$ the $\beta$-order derivative of $f$