Regularization in Predictive Models

Prof. Dennis Sun

Data 401
Regularization in Linear Models

The Stein phenomenon suggests that we can achieve better MSE by shrinking our estimates towards 0.

We have already seen a technique that shrinks coefficients towards 0:

$$\hat{\beta} = \arg\min_\beta \sum_{i=1}^{n} (y_i - x_i^T \beta)^2 + \lambda \sum_{j=1}^{p} |\beta_j|.$$

(The penalty on the magnitudes of the $\beta_j$s will shrink all coefficients.)

There is a related technique called ridge regression that instead penalizes the sum of squares of the coefficients:

$$\hat{\beta} = \arg\min_\beta \sum_{i=1}^{n} (y_i - x_i^T \beta)^2 + \lambda \sum_{j=1}^{p} \beta_j^2.$$
\( \ell_1 \textit{ vs. } \ell_2 \)

The sum of the absolute values of a vector is called the \( \ell_1 \text{ norm} \):

\[
\|v\|_1 = \sum_{j=1}^{p} |v_j|. 
\]

The square root of the sum of squared values of a vector is called the \( \ell_2 \text{ norm} \):

\[
\|v\|_2 = \sqrt{\sum_{j=1}^{p} v_j^2}. 
\]

Note that this is the usual “length” of a vector.

Notice that the lasso penalizes \( \ell_1 \) norm of the coefficients, while ridge regression penalizes the \textit{square} of the \( \ell_2 \) norm of the coefficients.
Exercise

Using calculus and linear algebra, derive a closed-form expression for the ridge regression estimator, defined as

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{n} (y_i - x_i^T \beta)^2 + \lambda \sum_{j=1}^{p} \beta_j^2.$$
Regularization in General Models

We can also apply a lasso ($\ell_1$) or ridge ($\ell_2$) penalty to other predictive models, like logistic regression:

$$\arg\min_{\mathbf{w}} \sum_{i=1}^{n} -y_i \log \frac{1}{1 + e^{-\mathbf{x}_i^T \mathbf{w}}} - (1 - y_i) \log \frac{1}{1 + e^{\mathbf{x}_i^T \mathbf{w}}} + \lambda \sum_{j=1}^{p} w_j^2.$$  

Recall that the soft-margin SVM solves the problem

$$\arg\min_{\mathbf{w}} \sum_{j=1}^{p} w_j^2 + C \sum_{i=1}^{n} \max(0, 1 - y_i \mathbf{x}_i^T \mathbf{w}).$$

Letting $\lambda = 1/C$, this is equivalent to solving

$$\arg\min_{\mathbf{w}} \sum_{i=1}^{n} \max(0, 1 - y_i \mathbf{x}_i^T \mathbf{w}) + \lambda \sum_{j=1}^{p} w_j^2.$$  

So the SVM has “built-in” $\ell_2$ regularization on the weights $\mathbf{w}$. 
Logistic Regression vs. SVM

Both logistic regression and SVM make predictions based on whether $x_i^T w \leq 0$.

So the only difference between $\ell_2$-regularized logistic regression and SVM is the loss function:

<table>
<thead>
<tr>
<th>method</th>
<th>loss function</th>
</tr>
</thead>
<tbody>
<tr>
<td>logistic regression $y_i \in {0, 1}$</td>
<td>$y_i \log \left(1 + e^{-x_i^T w}\right) + (1 - y_i) \log \left(1 + e^{x_i^T w}\right)$</td>
</tr>
<tr>
<td>SVM $y_i \in {-1, 1}$</td>
<td>$\max(0, 1 - y_i x_i^T w)$</td>
</tr>
</tbody>
</table>

Let’s investigate the difference between these loss functions.
Logistic Regression vs. SVM

Sketch the two loss functions, as a function of the prediction $x_i^T w$, when $y_i = 1$. 

$$L(y_i, x_i^T w)$$