Speaking Points - APAM Senior Seminar

Kyelee Fitts

Mathematical Background for Support Vector Regression Method

The Problem/Goal

Given training data $\{(x_1, y_1)...(x_l, y_l)\} \subset \mathcal{X} = \mathbb{R}^d \times \mathbb{R}\}$, we want to find f(x) that has at most ϵ deviation from targets and is as flat as possible.

Linear Case

We assume f has the form: $f(x) =< w, x > +b$ where $w \in \mathcal{X}f, b \in \mathbb{R}$. We want to make w as small as possible while also minimizing error. We formulate the optimization problem:

$$\begin{aligned} &\min \frac{1}{2}||w||^2 \\ &\text{subj. to } ||y_i - \langle w, x_i \rangle - b|| \leq \epsilon \end{aligned}$$

Solution may not be feasible (no such f) so introduce a problem which includes slack variables ξ_i to introduce a "soft margin."

$$\begin{aligned} &\min \frac{1}{2}||w||^2 + C\sum_{i=1}^{l} \xi_i \\ &\text{subj. to } \begin{cases} y_i - \langle w, x_i \rangle - b \leq \epsilon + \xi_i \\ -y_i + \langle w, x_i \rangle + b \leq \epsilon + \xi_i^* \\ \xi_i \geq 0 \end{cases} \end{aligned}$$

$$\begin{aligned} L = \frac{1}{2}||w||^2 + C\sum_{i=1}^{l}(\xi + \xi_i^*) - \sum_{i=1}^{l}(\eta_i \xi_i + \eta^* \xi_i^*) \\ - \sum_{i=1}^{l} \alpha_i(\epsilon + \xi_i - y_i + \langle w, x_i \rangle + b) - \sum_{i=1}^{l} \alpha_i^*(\epsilon + \xi_i^* - y_i + \langle w, x_i \rangle - b) \end{aligned}$$

Saddle points cause partial derivatives to vanish:

$$\begin{aligned} \delta_b L &= \sum_{i=1}^{l}(\alpha_i^* - \alpha_i) = 0 \\ \delta_w L &= w - \sum_{i=1}^{l}(\alpha_i^* - \alpha_i)x_i = 0 \\ \delta_{\xi_i^*} L &= C - \alpha_i^* - \eta_i^* = 0 \\ \delta_{\xi_i} L &= C - \alpha_i - \eta_i = 0 \end{aligned}$$

$$
\begin{aligned}
=&\frac{1}{2}\sum_{i,j=1}^{l}(\alpha_i-\alpha_i^*)(\alpha_j-\alpha_j^*)\langle x_i,x_j\rangle+\sum_{i=1}^{l}(\alpha_i^*+\eta_i^*)(\xi_i+\xi_i^*)-\sum_{i=1}^{l}(\eta_i\xi_i+\eta_i^*\xi_i^*)-\\
&-\sum_{i=1}^{l}\eta_i\xi_i-\sum_{i=1}^{l}\alpha_i\epsilon-\sum_{i=1}^{l}\alpha_i\xi_i+\sum_{i=1}^{l}\alpha_iy_i-\sum_{i=1}^{l}\alpha_i\langle w,x_i\rangle-\sum_{i=1}^{l}\alpha_ib-\sum_{i=1}^{l}\alpha_i^*\epsilon\\
&-\sum_{i=1}^{l}\xi_i^*\alpha_i^*-\sum_{i=1}^{l}\alpha_i^*y_i+\sum_{i=1}^{l}\alpha_i^*\langle w,x_i\rangle+\sum_{i=1}^{l}b\alpha_i^*\\
=&\frac{1}{2}\sum_{i,j=1}^{l}(\alpha_i-\alpha_i^*)(\alpha_j-\alpha_j^*)\langle x_i,x_j\rangle+\sum_{i=1}^{l}\alpha_i^*\xi_i+\sum_{i=1}^{l}\alpha_i^*\xi_i^*+\sum_{i=1}^{l}\eta_i^*\xi_i+\sum_{i=1}^{l}\eta_i^*\xi_i^*-\sum_{i=1}^{l}(\eta_i\xi_i+\eta_i^*\xi_i^*)\\
&-\sum_{i=1}^{l}\eta_i\xi_i-\sum_{i=1}^{l}\alpha_i\epsilon-\sum_{i=1}^{l}\alpha_i\xi_i+\sum_{i=1}^{l}\alpha_iy_i-\sum_{i=1}^{l}\alpha_i\langle w,x_i\rangle-\sum_{i=1}^{l}\alpha_ib-\sum_{i=1}^{l}\alpha_i^*\epsilon\\
&-\sum_{i=1}^{l}\xi_i^*\alpha_i^*-\sum_{i=1}^{l}\alpha_i^*y_i+\sum_{i=1}^{l}\alpha_i^*\langle w,x_i\rangle+\sum_{i=1}^{l}b\alpha_i^*\\
=&\frac{1}{2}\sum_{i,j=1}^{l}(\alpha_i-\alpha_i^*)(\alpha_j-\alpha_j^*)\langle x_i,x_j\rangle+\sum_{i=1}^{l}\xi_i(\alpha_i^*-\alpha_i)+\sum_{i=1}^{l}\xi_i^*(\alpha_i^*-\alpha_i)+\sum_{i=1}^{l}b(\alpha_i^*-\alpha_i)\\
&\sum_{i=1}^{l}\langle w,x_i\rangle(\alpha_i^*-\alpha_i)-\sum_{i=1}^{l}\alpha_i\epsilon+\sum_{i=1}^{l}\alpha_iy_i-\sum_{i=1}^{l}\alpha_i^*\epsilon-\sum_{i=1}^{l}\alpha_i^*y_i\\
=&\frac{1}{2}\sum_{i,j=1}^{l}(\alpha_i-\alpha_i^*)(\alpha_j-\alpha_j^*)\langle x_i,x_j\rangle-\sum_{i=1}^{l}\epsilon(\alpha_i+\alpha_i^*)+\sum_{i=1}^{l}y_i(\alpha_i-\alpha_i^*)
\end{aligned}
$$

We get the following dual optimization problem:

$$
\begin{aligned}
&\max\frac{1}{2}\sum_{i,j=1}^{l}(\alpha_i-\alpha_i^*)(\alpha_j-\alpha_j^*)\langle x_i,x_j\rangle-\epsilon\sum_{i=1}^{l}(\alpha_i+\alpha_i^*)+\sum_{i=1}^{l}y_i(\alpha_i-\alpha_i^*)\\
&\text{subj. to }\begin{cases}\sum_{i=1}^{l}\alpha_i=0\\ \alpha_i\in[0,C]\end{cases}
\end{aligned}
$$

Where $w=\sum_{i=1}^{l}\alpha_ix_i$ and $f(x)=\sum_{i=1}^{l}\alpha_i\langle x_i,x\rangle+b$. Note that in this formulation, $f(x)$ does not depend on w explicitly but on the inner product of the features. The dual problem intuitively means we want to maximize our model's slack while minimizing error.

$$
w=\sum_{i,j=1}^{l}(\alpha_i-\alpha_i^*)x_i
$$

$$
f(x)=\sum_{i,j=1}^{l}(\alpha_i-\alpha_i^*)\langle x_i,x\rangle+b
$$

Nonlinear Case

The above method depends on the fact that our feature space can be fit by a linear model. We can elegantly generalize SVR for a non-linear feature space by first transforming the data into a feature space, say, with some transformation $T(x) = \Phi(x)$ which makes it "linear" and then applying the dual problem. However, because of the above observation that our model $f(x)$ depends only on *inner products* we actually don't need to find $\Phi(x)$ explicitly.

We can use what are called "kernel functions" which are functions $k(x_i, x_j) : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. If our kernel function fulfills certain conditions (Mercer's conditions) which specify that it is among other things positive definite (this is the generalization of the concept of positive definiteness for matrix operators- we take the convolution of k with $L^2(\mathcal{X})$ functions) then k is the inner product of some feature space $\mathcal{F}$. So our new optimization problem is:

$$\max \frac{1}{2} \sum_{i,j=1}^{l} (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) k(x_i, x_j) - \epsilon \sum_{i=1}^{l} (\alpha_i + \alpha_i^*) + \sum_{i=1}^{l} y_i (\alpha_i - \alpha_i^*)$$

$$\text{subj. to } \begin{cases} \sum_{i=1}^{l} \alpha_i = 0 \\ \alpha_i \in [0, C] \end{cases}$$

Now we have $w = \sum_{i=1}^{l} \alpha_i \Phi(x_i)$ and $f(x) = \sum_{i=1}^{l} \alpha_i k(x_i, x) + b$.

The Actual Algorithm - Sketch

1. Remap the feature space using $\Phi(x)$. This corresponds to applying the kernel function to the feature vectors.
2. Solve the optimization problem using as weights α_i and the constant term b.

Why RBF Kernel?

$$k(x_1, x_2) = e^{-\gamma ||x_n - x_m||_2^2}$$

- Generally a good first choice of kernel if linear kernel doesn't work well and we're not creating our own kernel for the data.
- Infinitely smooth
- Translation invariant $K(x, y) = K(x + c, y + c)$ (why is this good?)
- Gives us another parameter, γ to optimize for overfitting. The two free parameters we have are C and γ. C is how much we control for using slack variables. A lower C means we don't penalize the use of slack variables and our error margins can be bigger. γ measure the amount of influence a single training vector has on the model. A large γ means that the training vectors only influence a small region around themselves - increase overfitting and complexity of model. A small γ means each training vector's sphere of influence is the entire dataset, which constrains the flexibility of the model. Balancing C and γ means we can hope for a sufficiently complex but not overfitted model.