Support Vector Regression

Amit Bohara
April 23 2012
Advanced Machine Learning
Regression (Least Squares Sense)

To

Support Vector Regression
How has the GDP been changed year to year the last few decades?

Some things to note:

- There is a natural variability in the data i.e. the answer is statistical and not definite in comparison (n choose k) has a definite answer.

- The better answer will usually capture as much variation in the data with minimal error (but will not explain all variation perfectly).
The simplest answer ...

Mean Change: 5.16 %

If you are hired to predict average year to year changes for the next 30 years and you assumed the above model you would on average be 2.16 % off
So how did we get the 'mean' value?

This is essentially a least squares REGRESSION problem:

Model: \( \hat{y} = C \)

We are going to select 'C' so that we minimize the least squares cost function, J

Cost: \( J = \sum e_i^2 = \sum (\hat{y}_i - y_i)^2 \)

The Best 'C' will minimize the cost 'J' i.e. for the optimal C the following equation holds

\[ \frac{\partial J}{\partial C} = 0 \]

Solution: \( C = \sum (Y_i)/N \)

Which is the mean.
What if you knew the percentage change in goods?

Having more information on the 'change in expenditure of goods' should enable you to better estimate. At least you should not do any worse than when you knew nothing, as in the first case.
The approach with one 'model' variable?

This is essentially a least squares **REgression** problem:

Model: \( \hat{y} = mX + C \)

We are going to select 'C' so that we minimize the least squares cost function, J

Cost: \( J = \sum e_i^2 = \sum (\hat{y}_i - y_i)^2 \)

The Best 'C' will minimize the cost 'J' i.e. for the optimal C the following equation holds

\[ \frac{\partial J}{\partial C} = 0 \]

Solution:

\[ m = \frac{\sum y_i (x_i - \bar{x})}{\sum (x_i - \bar{x})^2} \]

\[ C = \bar{y} - m \bar{x} \]

Which is the **mean**.
One variable regression model:

This is essentially a least squares REGRESSION problem:

\[
\hat{y} = m X + C
\]

Now you would be only 0.7% incorrect as opposed to 2.16% off.
Results so far:

Single Variable model: 0.7%

Mean Value: 2.16%
General Linear Regression Model:

The previous case can then be generalized two multi-variables

Model: \[ \hat{y} = m_1 X_1 + m_2 X_2 + \ldots + m_n X_n + C \]

Or even non-linear functions (e.g polynomial)

Model: \[ \hat{y} = m_1 f_1(x_1) + m_2 f_2(x_2) + \ldots + m_n f_n(X_n) + C \]

The general problem can then be solved by selecting a standard “Least Squares” cost function, \( J \)

Cost: \[ J = \sum e_i^2 = \sum (\hat{y}_i - y_i)^2 \]

The Best 'parameter' will minimize the cost 'J' i.e. for the optimal 'm' the following equation holds

\[ \frac{\partial J}{\partial C} = 0 \]

The general solution:

\[ \hat{B} = (X' X)^{-1} X' y \]
General Features of Regression so far ...

There is an analytical “model” that describe the problem generally well (parametric)

\[ \hat{y} = m_1 X_1 + m_2 X_2 + \ldots + m_n X_n + C \]

Quadratic cost function: points further away have more influence

All Points contribute (non linearly)

The main algorithm challenge is Matrix Inversion (well understood, fast & simple)

'Normal' distribution of errors is usually assumed to make 'confidence intervals' and inference decision
Regression (Least Squares Sense)  

To  

Support Vector Regression  

(At first look SVM classification does not intuitively point to curve fitting)
The Support Vector Regression Statement

Different Cost Function:

Instead of a least squares, a 'cost function' that preserves sparseness. The e-insensitive loss function

The equation may be written as:

\[ E_e(y(x) - t) = \begin{cases} 0, & \text{if } |y(x) - t| < e \\ |y(x) - t| - e, & \text{otherwise} \end{cases} \]

The overall cost function is:

\[ J = C \sum E_e(y(x_n) - t_n) + \frac{1}{2}||w||^2 \]

't' is the estimate from <w . x> + b

But why is 'w' in the equation? (Regularization)
The Support Vector Regression Statement

Re expression of cost function using slack variables:

For each data point let's define:

- 'SquigglePos (L)' for data points that lie above 'e'
- 'SquiggleNeg (L')' for data points that lie below 'e'

And both are defined to be positive i.e. they are just measures of the distance of each point lying outside the insensitive region.

The overall cost function is:

\[ J = C \sum (L + L') + \frac{1}{2} \| w \|^2 \]
The Support Vector Regression Statement

Re expression of cost function using slack variables:

\[ J = C \sum (L + L') + \frac{1}{2} ||w||^2 \]

4 Constraints:

1. L & L’ are both greater than 0, since they simply indicate distance

2. \( y_i - t \leq e + L \)

3. \( t - y_i \leq e + L' \)
The Support Vector Regression Statement

Construct a Lagrangian to turn a constrained problem into an unconstrained one

\[ L = C \sum (L + L') + \frac{1}{2} \| w \|^2 - \sum (n_i L_i + n'_i L'_i) - \sum (a_i (e + L_i + y_i - t_i)) - \sum (a'_i (e + L'_i - y_i + t_i)) \]

\begin{align*}
\text{cost function} & \quad \text{Constraint 1} & \quad \text{Constraint 2} & \quad \text{Constraint 3} \\
\end{align*}

4 Constraints:

1. \( L \) & \( L' \) are both greater than 0, since they simply indicate distance

2. \( y_i - t \leq e + L \)

3. \( t - y_i \leq e + L' \)

We have introduced the Lagrangian and 4 multipliers, \( n_i, n'_i, a_i, a'_i \)

The 'negative' sign of the constraints is due to the inequality in the constraints.
The Support Vector Regression Statement

Construct a Lagrangian to turn a constrained problem into an unconstrained one

\[
L = C \sum (L + L') + \frac{1}{2} \|w\|^2 - \sum (n_i L_i + n_i' L'_i) - \sum (a_i (e + Li + y_i - t_i)) - \sum (a_i' (e + L'_i - y_i + t_i))
\]

- **cost function**
- **Constraint 1**
- **Constraint 2**
- **Constraint 3**

At the optimal parameters, the partials of the primals vanish:

1. \( \partial_b L = \sum (a_i' - a_i) = 0 \)
2. \( \partial_w L = w - \sum (a_i - a_i') x_i = 0 \)
3. \( \partial_L L = C - a_i' - n_i = 0 \)
4. \( \partial_L' L = C - a_i' - n_i' = 0 \)

Substituting the results of these back into the Lagrangian, we get the dual representation

\[
L'(a, a') = -\frac{1}{2} \sum_i \sum_j (a_i - a_i')(a_j - a_j') \langle x_i, x_j \rangle - e \sum_n (a_n + a_n') + \sum_n (a_n - a_n') y_i
\]
The Support Vector Regression Dual

The problem is thus specified in the dual form \((a, a')\). All other variables have vanished from the Lagrangian cost function

\[
L'(a, a') = \frac{-1}{2} \sum_i \sum_j (a_i - a'_i)(a_j - a'_j)\langle x_i, x_j \rangle - e \sum_n (a_n + a'_n) + \sum_n (a_n - a'_n) y_i
\]

This is still a constrained maximization problem:

1. The Lagrange multipliers \( a_i, a'_i \geq 0 \)

2. From constraints, (previously stated in 3 and 4: \( \partial_L L = C - a'_i - n_i = 0 \))
   
   Since, \( n_i, n'_i \geq 0 \), because it is a Lagrange multiplier,

   This implies: \( a_n \leq C, a'_n \leq C \)

3. And the conditions previously stated:

   \( \partial_w L = w - \sum (a_i - a'_i) x_i = 0 \)

**The above is a standard quadratic programming problem** and we can solve for \( a (*) \):

From the last constraint:

\[
w = \sum (a_i - a'_i) x_i = 0
\]

Which can be substituted to get:

\[
y(x) = \sum (a_i - a'_i) \langle x_i, x'_i \rangle + b
\]
The Support Vector Regression: Solving 'b'

We have 'a' from the Lagrangian dual formulation

\[ y(x) = \sum (a_i - a'_i) \langle x_i, x'_i \rangle + b \]

For, 'b', we use the KKT conditions which state that solution of the product of the dual variables (a) and the constraints must vanish:

1. \[ a_n(e + L_i + y_i - t_i) = 0 \]
2. \[ a'_n(e + L'_i - y_i + t_i) = 0 \]
3. \[ (C - a_i) L_i = 0 \]
4. \[ (C - a'_i) L'_i = 0 \]

Some observations

A) Either a or a' has to be zero, since one of the slack variables (L or L') has to be zero

B) a or a' can only be non-zero if it lies outside the tube

C) From (3), if we consider a point, where \( a_i < C \), then \( L_i = 0 \) to satisfy the constraint.

Then from (1) \[ e + y_n - t_n = 0 \]

Which combined into the top equation gives:

\[ b = t_n - e - \sum (a_m - a'_m) \langle x_n, x_m \rangle \]
The Support Vector Regression Summary

Define a lost function
(e.g. Lost function)

Cost function using slack variables:

\[ J = C \sum \left( L + L' \right) + \frac{1}{2} \| w \|^2 \]

Construct a Lagrangian to turn a constrained problem into an unconstrained one

\[ L = C \sum \left( L + L' \right) + \frac{1}{2} \| w \|^2 - \sum \left( n_i L_i + n'_i L'_i \right) - \sum \left( a_i (e + L_i + y_i - t_i) \right) - \sum \left( a'_i (e + L'_i - y_i + t_i) \right) \]

Cast the problem into dual

\[ L'(a, a') = \frac{-1}{2} \sum_i \sum_j \left( a_i - a'_i \right) \left( a_j - a'_j \right) \langle x_i, x_j \rangle - e \sum_n \left( a_n + a'_n \right) + \sum_n \left( a_n - a'_n \right) y_i \]

The above solves for the dual coefficients \( a \) which can then be used in the following form:

\[ y(x) = \sum \left( a_i - a'_i \right) \langle x, x'_i \rangle + b \]
The Support Vector Regression Summary

1. Sparse... only points outside 'e' influence the results
2. Linear cost function (there are other functions)
3. Quadratic cost in computation
4. More complicated relative to least squares
5. Regularized (similar to maximum margin)