Support Vector Machine

Tianqi Chen
Nov. 5 2014
The Linear SVM Objective

- Maximizing the margin

\[
\arg\max_{w, w_0} \gamma \\
\text{subject to } \frac{1}{\|w\|} y^{(j)} (w^T x^{(j)} + w_0) \geq \gamma, j \in \{1, 2, \cdots, N\}
\]

Distance between x and hyper-plane, why?

- The objective that is usually used in

\[
\arg\min \|w\|^2 \\
\text{subject to } y^{(j)} (w^T x^{(j)} + w_0) \geq 1, j \in \{1, 2, \cdots, N\}
\]

- This is the objective used when the data is linearly separable
Constraint Violation and Slack Variables

Original SVM

\[
\begin{align*}
\text{argmin} & \|w\|^2 \\
\text{subject to} & \quad y^{(j)}(w^T x^{(j)} + w_0) \geq 1, j \in \{1, 2, \cdots, N\}
\end{align*}
\]

The soft constraint version

\[
\begin{align*}
\text{argmin} & \|w\|^2 + C \sum_{j=1}^{N} \xi^{(j)} \\
\text{subject to} & \quad y^{(j)}(w^T x^{(j)} + w_0) \geq 1 - \xi^{(j)}, \quad \xi^{(j)} \geq 0, j \in \{1, 2, \cdots, N\}
\end{align*}
\]

Slack variable: how much violation instance j have on the constraint

- This allows the constraint to be violated for some (outlier) j
- We add a linear penalty to the violations of constraint
Soft Constraint and Hinge Loss

- The soft constraint version

\[
\arg\min \|w\|^2 + C \sum_{j=1}^{N} \xi^{(j)}
\]
subject to
\[
y^{(j)}(w^T x^{(j)} + w_0) \geq 1 - \xi^{(j)}, \quad \xi^{(j)} \geq 0, \quad j \in \{1, 2, \ldots, N\}
\]

- This means \( \xi^{(j)} \geq 1 - y^{(j)}(w^T x^{(j)} + w_0) \) also note \( \xi^{(j)} \geq 0 \)

- The equivalent form

\[
\arg\min \|w\|^2 + C \sum_{j=1}^{N} \max \left(1 - y^{(j)}(w^T x^{(j)} + w_0), 0\right)
\]
Soft Constraint and Hinge Loss (cont’)

• Think of following new problem

• Assume we have set of pairs \( \{(x_1, z_1), (x_2, z_2), \ldots (x_N, z_N)\} \)
  - We know that for each pair, \( x \) is better than \( z \)
  - How can we learn the rank of the items from these pairs?
  - Objective will look like

\[
\begin{align*}
\text{argmin} & \|w\|^2 + C \sum_{j=1}^{N} \xi^{(j)} \\
\text{subject to} & \quad (w^T x^{(j)} + w_0) \geq (w^T z^{(j)} + w_0) + 1 - \xi^{(j)}, j \in \{1, 2, \ldots, N\}
\end{align*}
\]

- What is the corresponding hinge loss form?
SGD for Linear Model

- Think of how can you implement SGD for both logistic regression, linear regression and linear SVM

- General loss function

\[
L(w, w_0) = \frac{1}{N} \|w\|^2 + \frac{1}{N} \sum_{j=1}^{N} l(\hat{y}(j), y(j)), \quad \hat{y}(j) = w^T x^{(j)} + w_0
\]

- SGD update rule (derived using chain rule)

\[
w_i^{(t+1)} \leftarrow w_i^{(t)} - \eta \left( 2 \frac{1}{N} w_i^{(t)} + x_i^{(j)} \partial_{\hat{y}(j)} l(w^T x^{(j)} + w_0, y^{(j)}) \right)
\]

  - SVM hinge loss

  \[
l(\hat{y}, y) = \max(1 - \hat{y}y, 0), \quad \partial_{\hat{y}} l(\hat{y}, y) = \begin{cases} -y & \hat{y}y < 1 \\ 0 & \hat{y}y \geq 1 \end{cases}
\]

  - Ridge regression, square loss

\[
l(\hat{y}, y) = \frac{1}{2} (\hat{y} - y)^2, \quad \partial_{\hat{y}} l(\hat{y}, y) = \hat{y} - y
\]
• Again, think of separation between model and objective function (loss and regularization)

• Think of this question: How can you implement a SGD solver for logistic/linear regression and linear SVM, with L1 or L2 regularization supported.
  - I would encourage you to try, and see how much code you can reuse
  - Same thing applies beyond linear models (e.g. Matrix Factorization, Neural Nets)
One thing you need to know about Kernel

• Many machine learning models accepts kernel as input instead of explicit feature mapping.

\[ K(x^{(i)}, x^{(j)}) = \phi^T(x^{(i)}) \phi(x^{(j)}) \]

- Kernel
- Feature mapping

• When is kernel more helpful than explicit feature mapping?
  - Sometimes it is easier to specify inner product (distance) than explicit feature map
  - String kernels
  - Graph kernels
  - Image matching kernels
Midterm

• The grades has been posted

• When you have time, try to take a look at all the questions, including the one you did not manage to answer

• Try to learn from the questions 😊