Probabilistic Outputs for SVMs and Comparisons to Regularized Likelihood Methods

John Platt\textsuperscript{1}

January 31st 2007

\textsuperscript{1}Presented by Nikos Karampatziakis
Outline

- Background.
- Related Work.
- Platt’s Method.
- Results and Conclusions.
In many settings we want to give an input to a classifier and we are more interested in the degree of its belief that the output should be +1.

Typical examples include combining individual predictions and the “reject” option.

In such cases it is useful to produce a probability $P(y = 1|x)$.

However SVMs don’t do that.
Tradiotional SVM training:

\[
\min \frac{1}{2}||w||^2 + C \sum_i \xi_i
\]

s.t. \( y_i(w \cdot x_i) \geq 1 - \xi_i, \xi_i \geq 0 \)

Classification of a point \( x \): \( f(x) = w \cdot x \).

Focus on accuracy. Zero/one loss.

When the loss function is not symmetric probabilities can help.
Wahba’s Approach: Train an SVM to minimize the negative log likelihood

$$\min \sum_i -y_i f(x_i) + \log(1 + e^{f(x_i)})$$

Can add a regularization term to control the complexity of $f$ versus fit to the data.

In this case $P(y = 1|x) = \frac{1}{1+e^{-f(x)}}$.

This formulation gives solutions with many support vectors.
Vapnik’s Approach: Fit a function $P(y = 1|t, u)$ on the data.

Vapnik uses a linear combination of basis functions (cosines) to fit $P(y = 1|t, u)$.

- Need to solve a linear system for every new input $x$.
- Issues with monotonicity and outputs outside $[0, 1]$?
Hastie & Tibshirani: Fit gaussians(?) to $p(f|y = \pm 1)$.

\[
P(y = 1|f) = \frac{p(f|y = 1)P(y = 1)}{\sum_{i=\pm 1} P(y = i)p(f|y = i)}
\]

\[
= \frac{q \exp\left(\frac{-(f-\mu_1)^2}{\sigma_1^2}\right)}{q \exp\left(\frac{-(f-\mu_1)^2}{\sigma_1^2}\right) + (1-q) \exp\left(\frac{-(f-\mu_2)^2}{\sigma_2^2}\right)}
\]

\[
= \frac{1}{1 + \frac{1-q}{q} \exp\left(\frac{-(f-\mu_2)^2}{\sigma_2^2} + \frac{(f-\mu_1)^2}{\sigma_1^2}\right)}
\]

\[
= \frac{1}{1 + \exp\left(2\frac{\mu_2-\mu_1}{\sigma_2^2}f + \frac{\mu_1^2-\mu_2^2}{\sigma_1^2\sigma_2^2} + \ln \frac{1-q}{q}\right)}
\]

With unequal variances we get $P(y = 1|f) = \frac{1}{1+e^{af^2+bf+c}}$ where

\[
a = \frac{\sigma_2^2-\sigma_1^2}{\sigma_2^2\sigma_1^2}
\]

Non monotonic.
Platt’s Method (1)

- Idea: Look at the data! (but not Vapnik’s $u$)
Idea: Look at the data! (but not Vapnik’s $u$)

- $p(f|y = i)$ seems to be exponentially distributed when $f$ is in the wrong side of the margin. E.g. $p(f|y = 1) = r_1 e^{-r_1(1-f)}$, $f \leq 1$
- If we use Bayes’ rule we get

$$p(y = 1|f) = \frac{1}{1 + \exp(Af + B)}$$

where $A = -(r_1 + r_2)$ and $B = r_1 - r_2 + \ln \frac{P(y=-1)}{P(y=1)}$
Platt’s Method (2)

- Platt’s sigmoid vs. Hastie & Tibshirani’s sigmoid. (number of parameters, training procedure)
- Using the previous histogram we can compute estimates of $p(f \in \text{bin}_i)$ and $p(f \in \text{bin}_i|y=1)$. Then by Bayes’ rule:

$$P(y = 1|f \in \text{bin}_i) = \frac{p(f \in \text{bin}_i|y = 1)P(y = 1)}{p(f \in \text{bin}_i)}$$

- Plotting these probabilities and the fitted sigmoid we see that Platt’s sigmoid is doing well in practice.
- The reliability diagrams we saw are also sigmoid shaped.
Platt’s Method (3)

- Training data: \((p_i, t_i)\). \(p_i\) sigmoid’s response to \(f_i\), \(t_i = \frac{y_i + 1}{2}\).
- Parameters are fit by minimizing:
  \[
  - \log \prod p_i^{t_i} (1 - p_i)^{1 - t_i}
  \]
  \[
  \min - \sum_{i} t_i \log(p_i) + (1 - t_i) \log(1 - p_i)
  \]

- Issue: How to choose the training set?
- Using the output of the SVM for the training set.
- Biased estimate both for linear and non linear SVMs.
- (Re)using a hold out set. Using cross-validation.
Another issue: How to avoid overfitting?

Overfitting occurs when there are very few examples from one class which are separable from the other class.

Then the learned sigmoid is essentially a step function.

We are back to bad probabilities.

Add some regularization by changing \( t_i \ (0 \rightarrow \epsilon_-, \ 1 \rightarrow 1 - \epsilon_+) \).

Minimizing the same function is still valid.

Similar trick is used in neural net training when there is no error propagated back if the difference between the target and the output is small.
Platt’s Method (5)

MAP estimate

\[ t_i = \begin{cases} 
\frac{N_+ + 1}{N_+ + 2} & \text{if } y_i = 1 \\
\frac{1}{N_- + 2} & \text{if } y_i = -1 
\end{cases} \]
Platt’s Method (5)

- MAP estimate

\[
t_i = \begin{cases} 
\frac{N_+ + 1}{N_+ + 2} & \text{if } y_i = 1 \\
\frac{1}{N_- + 2} & \text{if } y_i = -1 
\end{cases}
\]

- These are derived in the same way as class probabilities in the leaves of a decision tree when we use Laplacian smoothing.

- We start with two examples one positive and one negative. Then we get \(N_+\) positives (\(N_-\) negatives).
Experiments

- Three models: raw SVM, SVM+sigmoid and SVM trained for maximizing log likelihood.
- Reuters, Adult and Web data sets.
- Linear and quadratic kernels.
- Accuracy of Raw SVM \( f(x) = 0 \) vs. SVM+sigmoid \( P(y = 1) = 0.5 \).
- Quality of probabilities of log likelihood SVM vs. SVM+sigmoid.
Zero threshold is not always optimal (we knew that from 578). Sigmoid threshold is significantly better for unbalanced problems.

Produced probabilities are not worse than those of regularized likelihood SVM. Solution is sparser and fitting the sigmoid is much simpler than implementing a kernel machine.

SVM with sigmoid and regularized likelihood SVM are trained to optimize one measure but they preform similarly for both accuracy and log likelihood. For a particular set of hypotheses (e.g. SVMs with quadratic kernels) it is hard to know in advance which training method will perform better.
Beware of the pseudocode! A recent paper by Chih-Jen Lin points out bugs and numerical difficulties.

Platt’s method is not specific to SVMs.

Any model that predicts poor probabilities should be calibrated but already well calibrated models such as neural nets cannot benefit from any type of calibration.

Reliability diagrams in the latter case are very close to straight lines and a sigmoid is not a good model for fitting straight lines.