Last Day

SVM and Soft Margins

First let's review SVM. In blue I note the more important changes needed for soft margins, in which we'll add slack variables $\boxed{\zeta_i > 0}$. C is a regularisation term.

We had a decision rule

$$\boxed{w \cdot u + b \geqslant 0 \Rightarrow +}$$

We introduced points x_+ and x_- such that

$$w \cdot x_+ + b \geqslant 1$$
$$w \cdot x_- + b \leqslant -1$$

For convenience, we introduced

$$y_i = \begin{cases} 1 & \text{positive samples} \\ -1 & \text{negative samples} \end{cases}$$

Multiplying by y_i, we learned that

$$y_i(w \cdot x_i + b) - 1 + \zeta \geqslant 0$$

So for x_i on the margin, we had

$$\boxed{y_i(w \cdot x_i + b) - 1 = 0}$$

We want to maximise the margin, and we learned that this meant we should look at minimising

$$\frac{1}{2} w \cdot w + C \sum_i \zeta_i = \frac{1}{2} \parallel w \parallel^2 + C \sum_i \zeta_i$$

Then we used the Lagrangian to find an objective function

$$\sum \lambda_i - \frac{1}{2} \sum_i \sum_j \lambda_i \lambda_j y_i y_j x_i \cdot x_j$$

and a recognition function

$$\sum \lambda_i y_i x_i \cdot u + b \geqslant 0 \Rightarrow +$$

Popular Kernels

Linear kernel: $(u \cdot v + 1)$

Polynomial kernel: $(u \cdot v + 1)^n$

Radial basis function (RBK, RBF):

$$e^{(-\|x_i - x_j\|)/\sigma}$$

(Note: If σ is too small, we overfit.)

ANN vs SVN

There's also logistic regression, which most of the time should be the first thing you try.

SVM is based on geometrical properties of the data while logistic regression is based on statistical approaches.

Some rules of thumb:

- Generally ANNs need more data than linear classifiers.
- ANN training is more easily parallelised (dot products vs matrix multiplications)
- If you think your data might be linearly separable, you should start with a linear separator (logistic regression or maybe SVM with linear kernel).
- Lots of features, few training examples: Using logistic regression or SVM without a kernel. It's unlikely that you'll be able to learn a complex surface.
- Few features and many training examples: SVM with Gaussian kernel may work (if linear separators are inadequate).

SVN with a non-linear kernel and MLP are often similar. You should start with SVM. It is very often more efficient. It is generally also easier to explain.

The two most important points:

- A NN (or an SVM) is often the second best solution to a problem. The best solution is to understand your problem better.
- Features are king. No algorithm will overcome bad features, and time spent on features is generally well worth it.

Learning Curves

Plot training score and cross validation score against training iterations.

Discuss what it means. In particular, discuss these points:

- convergence detection
- convergence rate (and parameter tuning)
- comparing algorithms
- will more data help?

Grid Search

Discussion of hyperparameters.