Classifiers in Practice

D.A. Forsyth
Rough draft of assignment

• Make an estimate of density of swimming pools per square kilometre for a suburb
• Check this estimate
• Use risk to modify your estimate
General procedure

• Get labelled data
  • pairs \((x_i, y_i)\), where \(x\) is feature vector, \(y\) label

• Split into 3 groups
  • Training (big)
  • Validation (smaller)
  • Test (small)

• Use software to train on training
  • for different values of theta
  • evaluate on validation; choose best theta

• Now evaluate on test
Evaluation

- Rough numbers
  - good for validation
  - Total error rate
    - % of classification attempts that get wrong answer (ideally, small)
  - Performance
    - % of classification attempts that get right answer (ideally, big)

- More detailed statistics
  - broader picture of performance
  - Recall
    - \( \frac{\text{number of true positives labelled true}}{\text{total number of true positives}} \)
  - Precision
    - \( \frac{\text{number of true positives labelled true}}{\text{total number labelled true}} \)
Turning a classifier into a detector

- **Procedure**
  - Sweep boxes over the image
    - compute features
    - present to classifier

- **Questions**
  - How big a step between boxes?
    - experiment
  - Blurred response
    - non-maximum suppression
Many good codes available

- **LIBSVM**
  - this implements a linear classifier
  - you can call from Matlab
  - easy script and examples on web page

- **SVMLight**
  - tends to be aimed at sophisticated users
  - complex interface
  - extremely accurate, and will do anything
  - [http://svmlight.joachims.org/](http://svmlight.joachims.org/)

- **VLFeat**
  - has a solver, VL_PEGASOS, which implements what I described in class
  - [http://www.vlfeat.org/](http://www.vlfeat.org/)