Training, test, and validation sets

Wassily Kandinsky, “Farbstudie Quadrate”

CSC320: Introduction to Visual Computing
Michael Guerzhoy
The General Scenario in Supervised Machine Learning

• We have a lot of labelled data
  • For example, we have 10,000 images, and we know which of the images are faces, and which aren’t
• We would like to be able to predict the label for new data
  • For example, for a new image, we would like to decide whether it is a face or not
• General approach: we would like to come up with a function $F(x, \alpha)$ which takes in an input $x$ (e.g. an image) and some alpha (e.g., a database of faces), and outputs the predicted label
An example of (a bad) $F(x, \alpha)$

- Set $\alpha$ to be all the faces in our database
- Now, $F(x, \alpha)$ will be “face” if $x$ is identical to one of the faces stores in $\alpha$, and “not a face” if $x$ is different from all the faces in $\alpha$
  - Will work well for some faces (the ones in $\alpha$), but not for faces in general
- (Note: the performance of $F$ on test $S$ is the proportion faces in $S$ that $F$ labels correctly)
Test Set

• To know how our F performs, will split our the face database into a **training set** and a **test set**
• We set alpha to the **training set**
• Now, we can test how $F(x, \alpha)$ performs on the **test set**, and see that it doesn’t do well
  – Because presumably, faces in the test set won’t be identical to any of the faces in alpha
  – Note: on the **training set**, $F(x, \alpha)$ will get things 100% right
Reminder: Face Detection with Eigenfaces

• For each (centered) window $x$ and for a set of principal components $V$, compute the Euclidean distance $|VV^T x - x|

• That is the distance between the reconstruction of $x$ and $x$. The reconstruction of $x$ is similar to $x$ if $x$ lies in the face subspace

• But how do we know how many eigenvectors in $V$ to use, and how small $|VV^T x - x|$ should be?
  • Could try different settings and see how good the performance is on the test set?
Don’t fit parameters using the Test Set

• Could try different settings and see how good the performance is on the test set
  • Bad idea! You would be picking the best settings for one particular test set, and then measuring the performance on the same set.
  • Biases the estimate for the performance on new data upwards
Solution: Use a Validation Set

• Set aside a **validation set**, on which the different settings (reminder: number of eigenvectors, how small is a small distances....) can be tried.

• Find the best settings on the **validation set**, and then use these settings to obtain the performance on the **test set**.

• This way, we obtain the best estimate for how our algorithm will perform on new data.
Summary

• Training set
  – The set used to construct the classifier
    • The set used to construct the eigenfaces
    • The set of faces with known names to which we compare new faces for face recognition
    • ...
  – Generally, larger than the test and validation sets

• Validation set
  – The set used to tune the settings of the classifier
  – Try different settings, and keep the ones for which the performance on the validation set is the best

• Test set
  – The set the performance on which is hopefully similar to performance on new data