Some bad practices in data analysis and machine learning

WONG Limsoon
Plan

PCA for dimension reduction
Indiscriminately using “high” PCs
Mindlessly discarding “low” PCs

Pearson correlation measure
Mechanically trusting high correlation scores
Not seeing association behind low correlation scores

Classification accuracy as indicator of model quality
Misinterpreting accuracy w/o consideration of prevalence
Unjustifiably treating all test instances as equal
Asinine propagation of bias
Irresponsible use of badly prepared test sets
A common advice on using PCA

PCA is the process of computing the principal components and using them to perform a change of basis on the data, …

It is commonly used for dimensionality reduction by projecting each data point onto only the first few principal components to obtain lower-dimensional data while preserving as much of the data's variation as possible.

This assumes variations in the first few PCs are more meaningful/useful than the other PCs. Is this a sound assumption?
PCA, intuitively
Exercise #1

Madrid and Warsaw are at almost the same distance to Italian cities.

Are Madrid and Warsaw near each other?
PCA of distance matrix of European cities to Italian cities

PC1 accounts for >99% of variance & correlates to distance of European cities to Latium cities

PC2, PC3, … account for < 1% of variance. Are they useless?
Variance is de-convoluted into real factors by PCA

PCs that don’t correspond to real factors are thus Gaussian-like residual noise

“Pre-whitening” can therefore be used to check which PC’s are informative:

*Inject small Gaussian noise into data*

*Compare $PC_i$ pre and post noise injection*

*High correlation $\Rightarrow$ $PC_i$ carries info*

*Low correlation $\Rightarrow$ $PC_i$ carries no info*
Pre-whitening to distinguish informative vs uninformative PCs

So, PC4 and PC5 are residual noise.

Whereas, PC1, PC2, and PC3 carry real information.

PC2 & PC3 are the angular orientation of European cities centered on Latium

So, you can tell Madrid is not near Warsaw
PCs corresponding < 1% of variation can be informative
Samples from different batches are grouped together, regardless of subtypes and treatment response.

Sometimes, a gene expression study may involve batches of data collected over a long period of time.

Image credit: Difeng Dong’s PhD dissertation, 2011
Exercise #2

Batch effects are unwanted sources of variation caused by different processing date, handling personnel, reagent lots, equipment, etc.

Batch effects is a big challenge faced in biological research, especially towards translational research and precision medicine.

How do you know which PCs are dominated by batch effects?
Paired boxplots of PCs

See which PC is enriched in batch effects by showing, side by side, distribution of values of each PC stratified by class and suspected batch variables.
Remove batch effects-laden PCs

Batch effects dominate

Class-effect discrimination recovered

(Notation: A/B_D/D*_.1/2 refers to the dataset, class and batches respectively)
Top PCs can correspond to irrelevant or confounding information

A common advice on using PCA

PCA is the process of computing the principal components and using them to perform a change of basis on the data, …

It is commonly used for dimensionality reduction by projecting each data point onto only the first few principal components to obtain lower-dimensional data while preserving as much of the data's variation as possible.

This assumes variations in the first few PCs are more meaningful/useful than the other PCs. Is this a sound assumption?
A correlation value of 0.8 between X and Y tells that an increase in X will lead to an increase in Y

And this insidious converse: No or low correlation between X and Y implies no relationship between X and Y

Are these sound advices? Note that correlation value is valid only when Y varies tightly and linearly wrt X.
### Anscombe's quartet

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean of $x$</td>
<td>9</td>
<td>exact</td>
</tr>
<tr>
<td>Sample variance of $x$ : $s_x^2$</td>
<td>11</td>
<td>exact</td>
</tr>
<tr>
<td>Mean of $y$</td>
<td>7.50</td>
<td>to 2 decimal places</td>
</tr>
<tr>
<td>Sample variance of $y$ : $s_y^2$</td>
<td>4.125</td>
<td>±0.003</td>
</tr>
<tr>
<td>Correlation between $x$ and $y$</td>
<td>0.816</td>
<td>to 3 decimal places</td>
</tr>
<tr>
<td>Linear regression line</td>
<td>$y = 3.00 + 0.500x$</td>
<td>to 2 and 3 decimal places, respectively</td>
</tr>
<tr>
<td>Coefficient of determination of the linear regression : $R^2$</td>
<td>0.87</td>
<td>to 2 decimal places</td>
</tr>
</tbody>
</table>

All four sets are identical when examined using simple summary statistics, but vary considerably when graphed.
High correlation does not imply good association
Vanderbilt GRE Study

Correlate GRE scores to things like year to thesis defense, publication count, etc.

Found no or low correlation

Conclude GRE is useless for PhD student admission purpose

Really?

Moneta-Koehler et al., PLOS ONE, 12(1):e0166742, 2017
A different trend is easily grasp if we swap the axes and bin the points by yr.

This trend is reproduced in a multi-institutional STEM PhD study.

This trend is reproduced in my own university’s data!


Alternative view of data of Petersen et al., *PLOS ONE*, 13(10):e0206570, 2018
The top 20% GRE scorers are different from everyone else

In fact, the top 20% vs bottom 20% comparison tells us more…

The bottom 20% was likely accepted to Vanderbilt based on undergrad grades, reference letters, research statements, interviews, etc.

So, these other considerations are less informative than GRE
**A simple model that explains GRE scores distribution**

“A” students master 7-10 topics  
“B” students master 6-9 topics  
“C” students master 5-8 topics

Exam covers 5 out of 10 topics, each topic is worth 20 marks. Then,  
“A” students get 40-100 marks, $P(80+|A) = 81\%$  
“B” students get 20-100 marks, $P(80+|B) = 63\%$  
“C” students get 0-100 marks, $P(80+|C) = 39\%$

Exercise: Derive $P(A|80+), P(B|80+), P(C|80+)$

50 “A” students, 150 “B” students, 150 “C” students
No correlation does not imply no association
A common practice in machine learning

Optimize and evaluate based on cross-validation accuracy

Is this a sound advice? Note that a vast majority of AI/machine learning projects fail on deployment
Exercise #3

You have a classifier. On a test set having 20% +ve and 80% -ve cases, the classifier’s recall and precision are both 80%.

Suppose you test it on a new test set having 80% +ve and 20% -ve cases. What do you expect its accuracy to be?

You may assume that the +ve (resp. –ve) cases in both test sets are equally sufficiently representative of the +ve (resp. –ve) real-world population.
Class proportion of test set is not always fidel to real life; accuracy determined from test set may not give the right picture of real-life performance.

Test set:
20% +ve, 80% -ve
\[
\text{recall} = 80\%, \text{precision} = 80\%
\]
\[
\therefore \text{specificity} = 95\%, \text{accuracy} = 92\%
\]

New test set “real life”:
80% +ve, 20% -ve

By “representativeness”,
\[
\text{recall} = 80\%, \text{specificity} = 95\%
\]
\[
\therefore \text{accuracy} = 83\%, \text{precision} = 98\%
\]
Accuracy measured from a test set must be calibrated for interpretability

Probably better to optimize wrt recall & specificity, as these are independent of class proportion
Protein function assignment

A protein is a large complex molecule made up of one or more chains of amino acids. Usually, only the sequence of amino acid is known.

Proteins perform a wide variety of activities in the cell. How do we predict the function of a protein?
A standard postulate based on evolution

In the course of evolution...

Two proteins (not) inheriting their function from a common ancestor (do not) have similar amino acid sequences.
Guilt by association

Compare $T$ with seqs of known function in a db

Poor Sequence Alignment

- Poor seq alignment shows few matched positions
  ⇒ The two proteins are not likely to be homologous

Good Sequence Alignment

- Good alignment usually has clusters of extensive matched positions
  ⇒ The two proteins are likely to be homologous

Assign to $T$ same function as homologs

Discard this function as a candidate

Confirm with suitable wet experiments
Twilight zone: Limit of sequence similarity-based protein function assignment

So, need clever methods for the twilight zone
DeepFam, deep learning for protein family prediction

This looks good

Really?

Table 2. Prediction accuracy (%) comparison of COG dataset

<table>
<thead>
<tr>
<th>Dataset</th>
<th>COG-500-1074</th>
<th>COG-250-1796</th>
<th>COG-100-2892</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeepFam</td>
<td>95.40</td>
<td>94.08</td>
<td>91.40</td>
</tr>
<tr>
<td>pHMM</td>
<td>91.75</td>
<td>91.78</td>
<td>91.67</td>
</tr>
<tr>
<td>3-mer LR</td>
<td>85.59</td>
<td>81.15</td>
<td>75.44</td>
</tr>
<tr>
<td>Protvec LR</td>
<td>47.34</td>
<td>41.76</td>
<td>37.05</td>
</tr>
</tbody>
</table>

Bold indicates the best performance for each dataset.
DeepFam’s good accuracy is largely due to “easy” proteins… it doesn’t advance the field.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Method</th>
<th>predCount = 1</th>
<th>predCount = 2</th>
<th>predCount = 3</th>
<th>predCount = 4</th>
<th>predCount = 5</th>
<th>predCount &gt; 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>COG-500-1074</td>
<td>EnsembleFam</td>
<td>72.07</td>
<td>81.00</td>
<td>82.82</td>
<td>84.96</td>
<td>85.33</td>
<td>85.27</td>
</tr>
<tr>
<td></td>
<td>pHMM</td>
<td>69.54</td>
<td>73.75</td>
<td>55.51</td>
<td>70.62</td>
<td>70.85</td>
<td>73.55</td>
</tr>
<tr>
<td></td>
<td>DeepFam</td>
<td>57.14</td>
<td>54.52</td>
<td>49.90</td>
<td>46.92</td>
<td>43.64</td>
<td>35.94</td>
</tr>
<tr>
<td>COG-250-1796</td>
<td>EnsembleFam</td>
<td>72.84</td>
<td>77.07</td>
<td>81.02</td>
<td>82.14</td>
<td>84.66</td>
<td>86.45</td>
</tr>
<tr>
<td></td>
<td>pHMM</td>
<td>75.39</td>
<td>73.82</td>
<td>73.84</td>
<td>71.02</td>
<td>67.44</td>
<td>72.43</td>
</tr>
<tr>
<td></td>
<td>DeepFam</td>
<td>32.44</td>
<td>32.54</td>
<td>30.24</td>
<td>29.53</td>
<td>30.02</td>
<td>28.68</td>
</tr>
<tr>
<td>COG-100-2892</td>
<td>EnsembleFam</td>
<td>75.24</td>
<td>79.55</td>
<td>81.21</td>
<td>80.63</td>
<td>82.05</td>
<td>88.95</td>
</tr>
<tr>
<td></td>
<td>pHMM</td>
<td>63.44</td>
<td>59.69</td>
<td>53.45</td>
<td>48.16</td>
<td>47.42</td>
<td>57.57</td>
</tr>
<tr>
<td></td>
<td>DeepFam</td>
<td>27.30</td>
<td>26.13</td>
<td>25.54</td>
<td>27.62</td>
<td>24.83</td>
<td>25.36</td>
</tr>
</tbody>
</table>
If there are few twilight zone proteins in real life, maybe DeepFam’s poor twilight zone performance is ok?

The reference database comprises proteins with known function.

If no function is predicted for a protein, or a wrong function is predicted, there won’t be any validated result for the protein.

**: Few twilight zone proteins can get into the reference database.

Catch-22!
Don’t be fooled by high accuracy on easy test sets

Need to stratify accuracy wrt easy and hard test instances
Exercise #4

How did EnsembleFam achieve its superior performance in the twilight zone?

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Method</th>
<th>predCount = 1</th>
<th>predCount = 2</th>
<th>predCount = 3</th>
<th>predCount = 4</th>
<th>predCount = 5</th>
<th>predCount &gt; 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>COG-500-1074</td>
<td>EnsembleFam</td>
<td>72.07</td>
<td>81.00</td>
<td>82.82</td>
<td>84.96</td>
<td>85.33</td>
<td>85.27</td>
</tr>
<tr>
<td></td>
<td>pHMM</td>
<td>69.54</td>
<td>73.75</td>
<td>55.51</td>
<td>70.62</td>
<td>70.85</td>
<td>73.55</td>
</tr>
<tr>
<td></td>
<td>DeepFam</td>
<td>57.14</td>
<td>54.52</td>
<td>49.90</td>
<td>46.92</td>
<td>43.64</td>
<td>35.94</td>
</tr>
<tr>
<td>COG-250-1796</td>
<td>EnsembleFam</td>
<td>72.84</td>
<td>77.07</td>
<td>81.02</td>
<td>82.14</td>
<td>84.66</td>
<td>86.45</td>
</tr>
<tr>
<td></td>
<td>pHMM</td>
<td>75.39</td>
<td>73.82</td>
<td>73.84</td>
<td>71.02</td>
<td>67.44</td>
<td>72.43</td>
</tr>
<tr>
<td></td>
<td>DeepFam</td>
<td>32.44</td>
<td>32.54</td>
<td>30.24</td>
<td>29.53</td>
<td>30.02</td>
<td>28.68</td>
</tr>
<tr>
<td>COG-100-2892</td>
<td>EnsembleFam</td>
<td>75.24</td>
<td>79.55</td>
<td>81.21</td>
<td>80.63</td>
<td>82.05</td>
<td>88.95</td>
</tr>
<tr>
<td></td>
<td>pHMM</td>
<td>63.44</td>
<td>59.69</td>
<td>53.45</td>
<td>48.16</td>
<td>47.42</td>
<td>57.57</td>
</tr>
<tr>
<td></td>
<td>DeepFam</td>
<td>27.30</td>
<td>26.13</td>
<td>25.54</td>
<td>27.62</td>
<td>24.83</td>
<td>25.36</td>
</tr>
</tbody>
</table>
EnsembleFam uses low-/dis-similarity information discarded by other methods!

Inspired by SVM-pairwise
Exercise #5

What qualities should a test set have?

Representativeness
Absence of doppelgangers
Etc.

Do you / your students / your professors ever check test sets for these qualities?
Coudray et al. report that common mutations in lung cancers can be predicted from histopath images using deep learning.

Is this claim sound based purely on their results?
Doppelgangers

Image credit: Mustafa Umit Oner
PCA is not about dimensionality reduction
It concerns deconvolution of variations
Top PCs may be irrelevant or confounding
Bottom PCs can be relevant and informative

High correlation may not imply association
No correlation does not imply no association
Accuracy measured from a test set must be calibrated wrt prevalence for interpretability

Accuracy should be “stratified” wrt easy and hard test cases

Discarded information can be very useful; cf. the informativeness of low PCs

Beware that test sets may not meet quality requirements or may not be used properly
References


Goh & Wong, “Protein complex-based analysis is resistant to the obfuscating consequences of batch effects---a case study in clinical proteomics”, *BMC Genomics*, 18(Suppl 2):142, 2017


Goh et al., “What can scatterplots teach us about doing data science better?”, *International Journal of Data Science and Analysis*, in press