I. Experiment

Based on "/work/arun/ybian/Gas_project/training_data/compressed_data_10_Half/Non_Probability_Sampling/X_data.txt"

This chunk of data has a dimension of (631, 80), thus 631 samples and each sample have 80 features. In the previous experiment, classifying this chunk of data using logistic regression can achieve an average accuracy of 99.5%. However, the static logistic coefficient array (w) has a dimension of (80, 4). Thus, 80 input features and 4 possible outputs (as there are 4 gas types). Now the goal is to reduce the size of static array to (8, 4). In essence, to reduce the number of features. We propose the following three options as dimension reduction measures: 1. Averaging 10 data segments (proposed by Dr. Somani) 2. Averaging 10 data points captured by the same sensor 3. PCA. In the following subsections, "FTC" stands for fail to converge. The converged model is marked in red. The test accuracies are demonstrated in a python dictionary. The keys are the proportion of the test data, for example, 0.6 means 40% data are selected for training and the rest 60% are for testing. The values are the accuracy number in each train-test scenario.

Option 1. Averaging 10 segments

100 iterations (FTC):

{0.6: 0.7757, 0.5: 0.7342, 0.4: 0.7273, 0.3: 0.6842, 0.2: 0.7638, 0.1: 0.7031}

200 iterations (FTC):

{0.6: 0.7995, 0.5: 0.807, 0.4: 0.8103, 0.3: 0.8, 0.2: 0.8504, 0.1: 0.8281}

300 iterations (FTC):

{0.6: 0.8021, 0.5: 0.8165, 0.4: 0.7905, 0.3: 0.8105, 0.2: 0.8268, 0.1: 0.7969}

1000 iterations (FTC):

{0.6: 0.8074, 0.5: 0.8449, 0.4: 0.83, 0.3: 0.8526, 0.2: 0.8504, 0.1: 0.875}

2000 iterations (FTC):

{0.6: 0.8417, 0.5: 0.8291, 0.4: 0.8419, 0.3: 0.8368, 0.2: 0.8819, 0.1: 0.875}

3000 iterations (FTC):

{0.6: 0.8522, 0.5: 0.8386, 0.4: 0.8458, 0.3: 0.8579, 0.2: 0.8898, 0.1: 0.8594}

Option 2. Averaging by sensor

100 iterations (FTC):

{0.6: 0.8522, 0.5: 0.8386, 0.4: 0.8775, 0.3: 0.8368, 0.2: 0.8583, 0.1: 0.8594}

200 iterations (FTC):

{0.6: 0.8549, 0.5: 0.8576, 0.4: 0.8854, 0.3: 0.8947, 0.2: 0.8583, 0.1: 0.7812}

300 iterations (FTC):

{0.6: 0.847, 0.5: 0.8987, 0.4: 0.8933, 0.3: 0.8789, 0.2: 0.8425, 0.1: 0.9219}

1000 iterations (FTC):

{0.6: 0.8734, 0.5: 0.8766, 0.4: 0.8735, 0.3: 0.8316, 0.2: 0.8898, 0.1: 0.9062}

2000 iterations (FTC):

{0.6: 0.8945, 0.5: 0.8797, 0.4: 0.913, 0.3: 0.8632, 0.2: 0.8583, 0.1: 0.7812}

3000 iterations:

{0.6: 0.8707, 0.5: 0.8924, 0.4: 0.913, 0.3: 0.8526, 0.2: 0.8976, 0.1: 0.9375}

Option 3. PCA (assume all features are independent)

100 iterations (FTC):

{0.6: 0.9551, 0.5: 0.9462, 0.4: 0.9486, 0.3: 0.9842, 0.2: 0.9685, 0.1: 0.9531}

200 iterations:

{0.6: 0.942, 0.5: 0.9589, 0.4: 0.9447, 0.3: 0.9684, 0.2: 0.9764, 0.1: 0.9688}

II. Results

Using option 2:

90% training + 10% testing. 3000 iterations. Accuracy = 93.75%

$$w = \begin{pmatrix} -0.61318024 & -0.47519703 & -0.11495193 & 1.20332919 \\ -1.51951389 & 0.36867674 & 0.22239161 & 0.92844554 \\ 1.07570403 & 0.74342827 & -0.63922379 & -1.17990851 \\ 0.70287702 & -2.65763186 & 0.46923062 & 1.48552423 \\ -0.30401349 & 3.34037128 & -0.97027463 & -2.06608316 \\ 0.62616087 & -1.03775693 & 0.05125071 & 0.36034535 \\ -0.38251275 & -0.91347073 & 0.89578473 & 0.40019875 \\ 0.45868306 & 0.61147558 & 0.02525933 & -1.09541797 \end{pmatrix}, b = \begin{pmatrix} -0.99829749 \\ -1.06392822 \\ -0.08098097 \\ 2.14320667 \end{pmatrix}$$

Using option 3:

90% training + 10% testing. 200 iterations. Accuracy = 96.88%

$$w = \begin{pmatrix} 0.95214888 & 0.58015171 & -0.42451341 & -1.10778719 \\ -0.4001824 & -1.69110466 & 0.66325837 & 1.42802869 \\ -1.28542757 & -0.77068368 & 0.62916873 & 1.42694252 \\ -0.89447982 & 0.73178558 & 0.3884338 & -0.22573957 \\ -0.06309417 & -0.08011296 & 0.09836984 & 0.04483729 \\ -0.98290776 & 4.17844246 & 0.17671509 & -3.3722498 \\ 0.24860825 & -1.54597575 & -0.21623488 & 1.51360237 \\ -1.12810469 & 0.25419034 & 0.45996738 & 0.41394697 \end{pmatrix}, b = \begin{pmatrix} -4.50098815 \\ -3.08026881 \\ 5.45359694 \\ 2.12766002 \end{pmatrix}$$

III. Conclusions

A concise comparison among three proposed options are shown as follows.

Option	Iterations for convergence	Best accuracy
1	> 3000	≈ 88%
2	3000	93.75%
3	200	96.88%

Option 3, which adopts PCA to reduce the data dimension, enables the logistic regression classifier converges very quickly (200 iterations) and has the best accuracy (96.88%). It is the best option if the hardware supports all the required operations including co-variance matrix calculation etc.

The model finally converges on the reduced data by option 2 after 3000 iterations. Its accuracy is not the best but still comparable (93.75%). This dimension reduction method can be of interest due to its simplicity, as it only averages the 10 data points captured by each sensor.

To conclude the overall data processing steps, given the original data (60000, 8), we first drop the second half as that part tends to be flat (30000, 8). Then we down-sampling by keeping the first number of every 3000 points (10, 8). With 80 data points for each gas sample, if we apply option 2, we average 10 data points for each sensor (1, 8). If we apply option 3, we apply PCA to reduce the dimension to a specified one. In the above case, it is 8. Then we fit a logistic regression classifier to the reduced data.

As we can specify what size of dimension the data can be reduced to using PCA, below we did a small experiment to explore the minimum dimension size while the performance is still descent. According to our results, the dimensions can be reduced from (10, 8) to (1, 6) while the accuracies are better than 95%. It reaches 100% in some randomized processes.

PCA component	Iterations for convergence	Best accuracy
7	200	0.9685
6	200	1
5	100	0.7795
4	100	0.8291
3	200 (100 converge, 200 best)	0.7656
2	100	0.625

Based on the observations of the original, some sensors do not capture variant data points. We can select those sensors which capture highly fluctuated data that distinguish gas types. Therefore, other dimension reduction ideas that reduced the number of features are: 1. PCA (1 feature for each sensor); 2. PCA (2 features for selected 4 sensors); 3. PCA (4 features for selected 2 sensors).