1 Problem

Given two sets of data: images and weather reports, the problem is to train a model to accurately guess the weather from the images. The five weather categories used are "Clear", "Somewhat Cloudy", "Very Cloudy", "Rain", and "Snow".

2 Data

I used the given data. roughly ten thousand weather reports, and five thousand images. First I collect the weather data, which is needed to be in a precise format: csv files with data starting on 15th line of text. Right away I clean it, then from the Date/Time column I choose which image files to read from. I read with 64x48 pixel resolution to speed along the process.

In cleaning I make the following decisions.

- Use only good "Data Quality", (which is not missing or estimating data)
- Exclude any weather that is "NA", since it is not useful
- Exclude thunderstorm data since the sample is too small
- Ignore "Drizzle" weather since it confuses me, (18 rows only)

I end up with 2200 matched weather-image pairs. Classifications are extracted from weather, while features are extracted from the image pixels. From weather I extract two categories of information: The sky categorized as "Clear", "Somewhat Cloudy", "Very Cloudy", "Rain", and "Snow". And fog, categorized as "Fog" or "No Fog". At this point I have only used the first 5 class sky category. As for features, available options are the overall Brightness of the entire image, the overall red green and blue, and a list of all pixels as grayscale. I used each of these three

3 Analysis

Because there were so many features I started with decomposition. The techniques I used for each model were

- Scaling (with resolutions of 256*192, 128*96, 64*48)
- PCA (with feature counts within the range of 10 and 250)

Then I used the following models.

- Naive Bayes
- K-Nearest Neighbours (slight performance boost with n=10)
- Support Vector Machine (kernel radial based function)

I also tried to use ONLY the overall brightness and colours, which did surprisingly well with K-Nearest-Neighbours.

I found that my best results were with Support Vector Machine, C=2, gamma=1e-6. Resolution and PCA did not make a big difference when I used high values, so I stuck with 64*48 and 100 features.

4 Results

I get an average result of 66% accuracy for my 5 categories. This definitely means that my model is doing something, but whether it is a good result is another story. The classifier is working very poorly for "Clear" weather, almost always claiming it is somewhat cloudy, but it does surprisingly well for snow, considering the small sample size for snowy days.

One interesting outcome was how easily it was to get a decent K-Nearest-Neighbours result. I could get about 60% doing almost nothing and applying K-Nearest Neighbours. And the performance was about the same when looking for 1 neighbour or 10. This got me thinking. Because the weather is from consecutive hours from the same location, I expect that some of the sky images look almost identical. Instead of learning what a cloud looks like it may be getting effective results just by looking for near identical images. For this reason I see my 66% score as being fairly trivial.

5 Visualizations

Placeholder (I hope):

Totals

	Clear	Somewhat Cloudy	Very Cloudy	Rain	Snow	total
Clear	4	23	2	0	0	29
Somewhat Cloudy	4	114	25	7	0	150
Very Cloudy	0	30	36	28	2	96
Rain	0	8	21	120	1	150
Snow	0	0	0	4	13	17

Percents

	Clear	Somewhat Cloudy	Very Cloudy	Rain	Snow	correct
Clear	0.14	0.15	0.02	0.00	0.00	0.14
Somewhat Cloudy	0.14	0.76	0.26	0.05	0.00	0.76
Very Cloudy	0.00	0.20	0.38	0.19	0.12	0.38
Rain	0.00	0.05	0.22	0.80	0.06	0.80
Snow	0.00	0.00	0.00	0.03	0.76	0.76

6 Limitations

Here are some of the major limitations:

- I didn't address that the locations of weather and images do not match.
- I run PCA on all features, but I would rather run PCA only on my pixels, and leave some features such as the overall hue and brightness alone.

If I spent more time on the project I would look into solving the above, but also:

- Try my classifier for categories like fog vs no-fog
- Try many other ways to represent the image (grayscale/colour/edges only/etc...)

7 Project Experience Summary

In my resume I would include this experience something like this:

Projects

Image Classification Project: https://github.com/cbabnik/cmpt318-project

- Used Machine Learning strategies to predict weather reports from images.
- Extracted useful manageable features from a raw array of pixel data.
- Cleaned and organized data into a useable state.