Today

- Supervised learning: a direct probabilistic approach
  - Consistency, and optimality of MAP classification

- Single pixel features
  - How to choose them?

- Two pixel features
  - How to choose them?

- How many pixels should we use?
MAP classification

- Supervised learning: a direct probabilistic approach
  - Choose features
  - Estimate probabilities of those features for each class
  - Use Bayes’ rule to compute posterior probability
  - Choose class with highest posterior:
    \[ \text{maximum a posteriori (MAP) classification} \]

(What do we do in case of a tie?)
“True Distributions”

- $P(X|Y = 1)$
- $P(X|Y = 2)$
- $P(Y = 1|X)$
- $P(Y = 2|X)$
Estimated Distributions

\[ \hat{P}(X|Y = 1) \]

\[ \hat{P}(X|Y = 2) \]

\[ \hat{P}(Y = 1|X) \]

\[ \hat{P}(Y = 2|X) \]
Likelihood in 2 dimensions

$P(X|Y = 1)$
MAP Classification

- When used with the exact likelihoods and priors
  - Minimizes probability of error over ALL decision functions.
  - \textit{There exists NO BETTER CLASSIFIER} in terms of minimizing the probability of error.

- T-Maze example.
MAP Classification

- When used with the ESTIMATED likelihoods and priors
  - No guarantees for poor estimates.
  - However
    - Consistent estimators of likelihoods and priors yield consistent classifiers*
Consistent estimators

- Consistent estimator: as I gather more and more data, the difference between the true value of the estimate and the actual value goes to 0.
- Example of consistent estimator: sampling from a discrete distribution and estimating the probability of each outcome by its frequency.
- Not consistent: Estimate a probability distribution by assuming it’s Gaussian (normal) and finding the best fitting Gaussian.
Summary

- If we have enough data to estimate likelihood distributions and priors well
  - Use a consistent estimator of distributions
  - Use Bayes rule to estimate posteriors
  - Choose maximum posterior class (MAP classification)
  - Should get error close to minimum possible error.
High dimensions and lack of data

- Fundamental problem in vision:
  - don’t have enough data to estimate 10,000-dimensional probability distributions!
- Must reduce the number of things to estimate.
  Possible approaches:
  - Use a subset of pixels
  - Compute small number of features that are functions of the pixels. There are a lot of these!
  - Constrain form of estimates.
    - Gaussian
    - Only allow 3 probability levels?
    - etc.
Start with a single pixel

- Assignment 1:
  - Estimate $p(X|Y=\text{“3”})$
  - $p(X|Y=\text{“5”})$
  - Use Bayes rule to invert.

- Not all pixels are equal!
  - Which pixel to select is topic of “feature selection” methods.
Means
Code for means:

```matlab
load '~/Desktop/Teaching/Data/digits.mat';
clf;
figure(1);
subplot(1,3,1);
colormap(gray);
imagesc(mean(train_threes,3));

subplot(1,3,2);
colormap(gray);
imagesc(mean(train_fives,3));
```
Means and differences of means
What about 2 pixels?

- First question: do we have enough data to estimate 
  \( p(X_1, X_2 \mid Y=\text{“3”}) \) and 
  \( p(X_1, X_2 \mid Y=\text{“5”}) \) ?

- Which two pixels?
• The story of the late professor and the frosty windshield....
• The story of the late professor and the frosty windshield....

• Moral of the story: 
  *We want to choose features that are informative, but also features that contain independent information!*
Statistical Independence
Random variables $X$ and $Y$ are statistically independent if and only if

$$P(X, Y) = P(X)P(Y).$$
Random variables $X$ and $Y$ are **statistically independent** if and only if

$$P(X, Y) = P(X)P(Y).$$

**Mini-quiz.**
Good features

• We would like features that are NOT independent of the class we are trying to guess. That is, they should be dependent on the class.

• We would like features that are as INDEPENDENT as possible from each other.

• How do we measure the “quantity” of statistical dependence?
Mutual Information between feature and class

\[ I(X; C) = \sum_{x \in \mathcal{X}, c \in \mathcal{C}} P(X, C) \log \frac{P(X, C)}{P(X)P(C)} \]
Information Gain

- After choosing the most informative feature (highest MI with class label)
  choose feature which *adds the most information.*

\[ I(X_2; C|X_1) = I(X_1, X_2; C) - I(X_1; C). \]
Greedy versus global

- To pick the best 2 features, we would like to optimize:

\[ I(X_1, X_2; C) \]

This requires us to examine N-choose-2 feature pairs. \( O(N^2) \).

- Greedy alternative:
  - Pick best \( I(X_1; C) \)
  - Then pick best \( I(X_2; C|X_1) \)

Suboptimal, but what is complexity?