Cross-validation

Review from last time

- Standard way to evaluate models
- Exclude a subset of the data while fitting model
- Compute predictions for the held-out subset.
- Evaluate predictions against actual held-out values
  - e.g., distance from truth, or class labels
- If you use k such sets, this is called k-fold cross-validation
- If you leave out 1 data point, it is called leave-one-out.

Cross-validation (2)

Review from last time

- Cross-validation provides
  - A way to choose models
  - A way to measure performance
  - A way to measure generalization capacity
- Held out data **must be different enough** to test the level of generality that you want
  - Consider degree of validation in a model to predict happiness
    1. How happy are you now given recent data points
    2. How happy are you now given all data points
    3. How happy are you on day X given data for other days
    4. How happy are you based on model of **other** people
    5. How happy are you based on **other** people in other experiments
    6. How happy are you based on modeling people in other cultures

Classification

- Consider that our parameters include a discrete class variable, c.
- Assume no other variables, or that they have been marginalized out.
- Use x for the data. Then the posterior over classes is

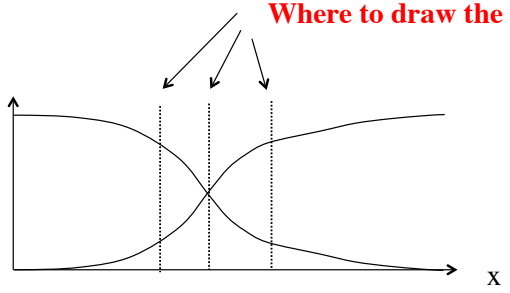
$$p(c \mid x) \propto p(c)p(x \mid c)$$

- So, given x, what is the class?

Classification

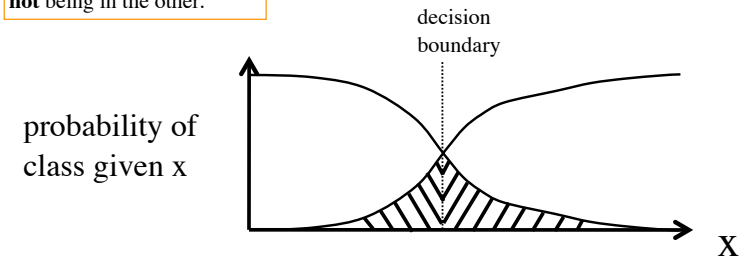
Binary case, easy to draw
Two classes, C_1 and C_2.
being in one is the same as **not** being in the other.

Where to draw the line?

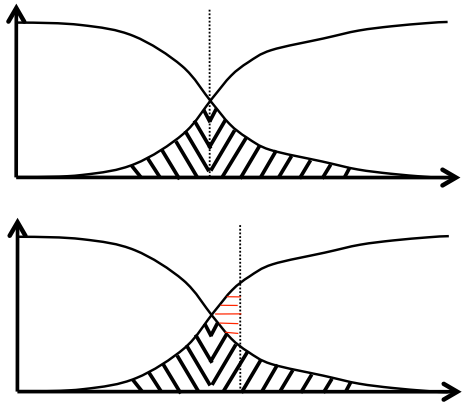


Classification

Binary case, easy to draw
Two classes, C_1 and C_2.
being in one is the same as **not** being in the other.



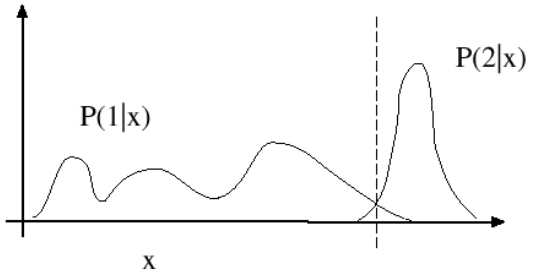
Area of intersection under curves, integrated against p(x), gives expected value of making a mistake



Red shows extra that you get wrong with different boundary

Classification

Finding a decision boundary is not the same as modeling a conditional density.



Here there are more than two classes, but only two shown. Consider all animals, but you are being force to choose between "dog" and "cat".

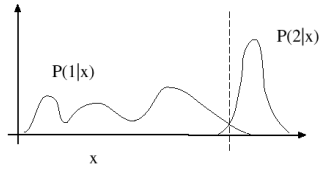
Classification

Finding a decision boundary is not the same as modeling a conditional density.

Working with the boundary might be easier (we don't care about the extra bumps).

But we loose any indication of whether the point is an outlier.

In this course we will not cover in detail methods for finding the boundary (discriminative method).



Following Bishop §1.5

Expected chance of misclassifications

- Assume that we have classes C_k.
- Assume that we map points in a region, R_j, to class j.

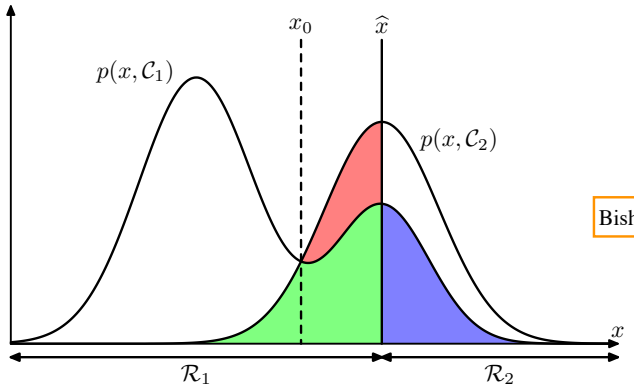
- (For now, just think about two classes)
- What is the probability of being wrong?
- Equivalently, what is 1- probability of being right?

$$p(\text{mistake}) = p(x \in R_1, C_2) + p(x \in R_2, C_1)$$
$$p(\text{correct}) = p(x \in R_1, C_1) + p(x \in R_2, C_2)$$

Expected chance of misclassifications

$$p(\text{mistake}) = p(x \in R_1, C_2) + p(x \in R_2, C_1)$$
$$= \int_{R_1} p(x, C_2)\,dx + \int_{R_2} p(x, C_1)\,dx$$

$$p(\text{correct}) = p(x \in R_1, C_1) + p(x \in R_1, C_1)$$
$$= \int_{R_1} p(x, C_1)\,dx + \int_{R_2} p(x, C_2)\,dx$$



Bishop, figure 1.24

$GREEN + RED = p(R_1, C_2)$
$BLUE = p(R_2, C_1)$
$p(\text{mistake}) = GREEN + RED + BLUE$
If we divide the regions at x_0 then $p(\text{mistake}) = GREEN$

For K classes,

$$p(\text{correct}) = \sum_{k=1}^{K} \int_{R_k} p(x, C_k)\,dx$$

To make $p(\text{correct})$ as big as possible, we choose R_k where $p(x, C_k)$ is largest. IE,

$$R_k = \left\{ x \middle| p(x, C_k) \geq p(x, C_j)\ \forall j \neq k \right\}$$

(Moving a set of points, dx, from R_j to R_k where $p(x, C_k) > p(x, C_j)$ makes p(correct) bigger.)

Bishop §1.5

Decision making

For classification accuracy, a false positive and a false negative count the same. But what if they should be treated differently?

Example: Risk of a false negative diagnosis is more than that for the risk of false positive diagnosis.

Define a loss function, $L_{k,j}$ which tells us the loss of classifying a true category k, as a category, j.

Example:

columns index "classified as"

rows index true class

	cancer	normal
cancer	0	1000
normal	1	0

Decision making

Now the classification boundaries for **x** are based on the loss, not just the probability.

Your choice of the class, j, for x, should be the one with the **lowest expected loss**.

This is found by:

$$\underset{j}{\operatorname{argmin}}\left\{\sum_k L_{k,j}\cdot p(C_k|x)\right\}$$

Decision making

The **lowest expected loss** is found by choosing class j where

$$\underset{j}{\operatorname{argmin}}\left\{\sum_k L_{k,j}\cdot p(C_k|x)\right\}$$

Penalty for calling it j when it is k

How probable is it k?

Your total expected loss is the sum of the loss over the possible true classes over all x. Given x, this above contributes the least to the expectation.

Check that we get the same answer for binary classification

$$L_{k,j} = (k \neq j)$$

$$\underset{j}{\operatorname{argmin}}\left\{\sum_k L_{k,j}\cdot p(C_k|x)\right\} = \underset{j}{\operatorname{argmin}}\begin{Bmatrix} j=1 \,\&\; p(C_2|x) \\ j=2 \,\&\; p(C_1|x)\end{Bmatrix}$$

Most right == least wrong.

So, if $p(C_2|x)$ is smaller than $p(C_1|x)$, to have minimal mistakes, we need to choose j=1.

(Same answer as when we focused on maximizing p(correct))

Decision making

Additional example to illustrate that the formula is sensible.

Suppose that at a given x*, we have

$p(C_1 \mid x^*) = 0.3 \quad p(C_2 \mid x^*) = 0.2 \quad p(C_3 \mid x^*) = 0.5$

Evaluate the assignment of x* under loss functions

$$L_A = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix} \qquad L_B = \begin{pmatrix} 0 & 1 & 1 \\ 10 & 0 & 10 \\ 1 & 1 & 0 \end{pmatrix}$$

$p(C_1 \mid x^*) = 0.3 \quad p(C_2 \mid x^*) = 0.2 \quad p(C_3 \mid x^*) = 0.5$

For the first example (loss is misclassifcation rate)

$$L_B = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}$$

Note that loss is defined for misclassifying the column item as the row item.

Declaring that x at x* is C_1 has expected loss: (0.3)*0 +(0.2)*1+ (0.5)*1=0.7

Declaring that x at x* is C_2 has expected loss: (0.3)*1 +(0.2)*0+ (0.5)*1=0.8

Declaring that x at x* is C_3 has expected loss: (0.3)*1 +(0.2)*1+ (0.5)*0=0.5

As expected, the minimum loss is for the likeliest class.

$p(C_1 \mid x^*) = 0.3 \quad p(C_2 \mid x^*) = 0.2 \quad p(C_3 \mid x^*) = 0.5$

For the second example

$$L_B = \begin{pmatrix} 0 & 1 & 1 \\ 10 & 0 & 10 \\ 1 & 1 & 0 \end{pmatrix}$$

Note that loss is defined for misclassifying the column item as the row item.

Declaring that x at x* is C_1 has expected loss: (0.3)*0 +(0.2)*10+ (0.5)*1 = 2.5

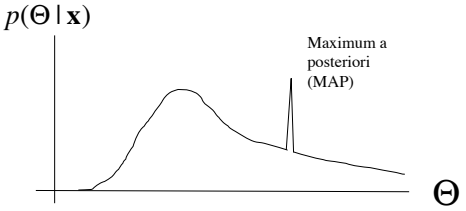
Declaring that x at x* is C_2 has expected loss: (0.3)*1 +(0.2)*0+ (0.5)*1 = 0.8

Declaring that x at x* is C_3 has expected loss: (0.3)*1 +(0.2)*10+ (0.5)*0 = 2.3

Now the heavy penalty for missing C_2 leads to C_2 being the best answer. (Note that C_2 was the worst answer with the previous loss).

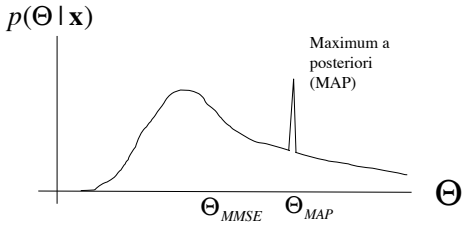
More on estimation

- If the goal is to provide the model, then we often estimate the MAP value for the parameters
- This assumes that the posterior is nicely behaved
- An alternative is to average some or all (MMSE) of the posterior.



Minimum Squared Error (MMSE) Estimate

- MMSE estimate is the expected value of the parameters with respect to the posterior.
- Average of the parameter values if sampled from $p(\Theta|\mathbf{x})$
- Weighted average, where where weight is $p(\Theta|\mathbf{x})$



$$\Theta_{MMSE} = \int \Theta \; p(\Theta|\mathbf{x}) d\Theta$$

Loss functions for continuous variables

$L(\Theta,\hat{\Theta})$ tells us the loss of each Θ, given an estimate, $\hat{\Theta}$.

The expected loss is $\int L(\Theta,\hat{\Theta}) p(\Theta|X) d\Theta$

A loss function drives the estimate because we want to minimize the expected loss.

A common choice is $L(\Theta,\hat{\Theta}) = (\Theta - \hat{\Theta})^2$ (squared loss)

The minimizer here is $\hat{\Theta} = \int \hat{\Theta}\, p(\Theta|X) d\Theta$

(IE, the MMSE estimate we just looked at)