COMP 652: Machine Learning

Lecture 7b: Generative learning

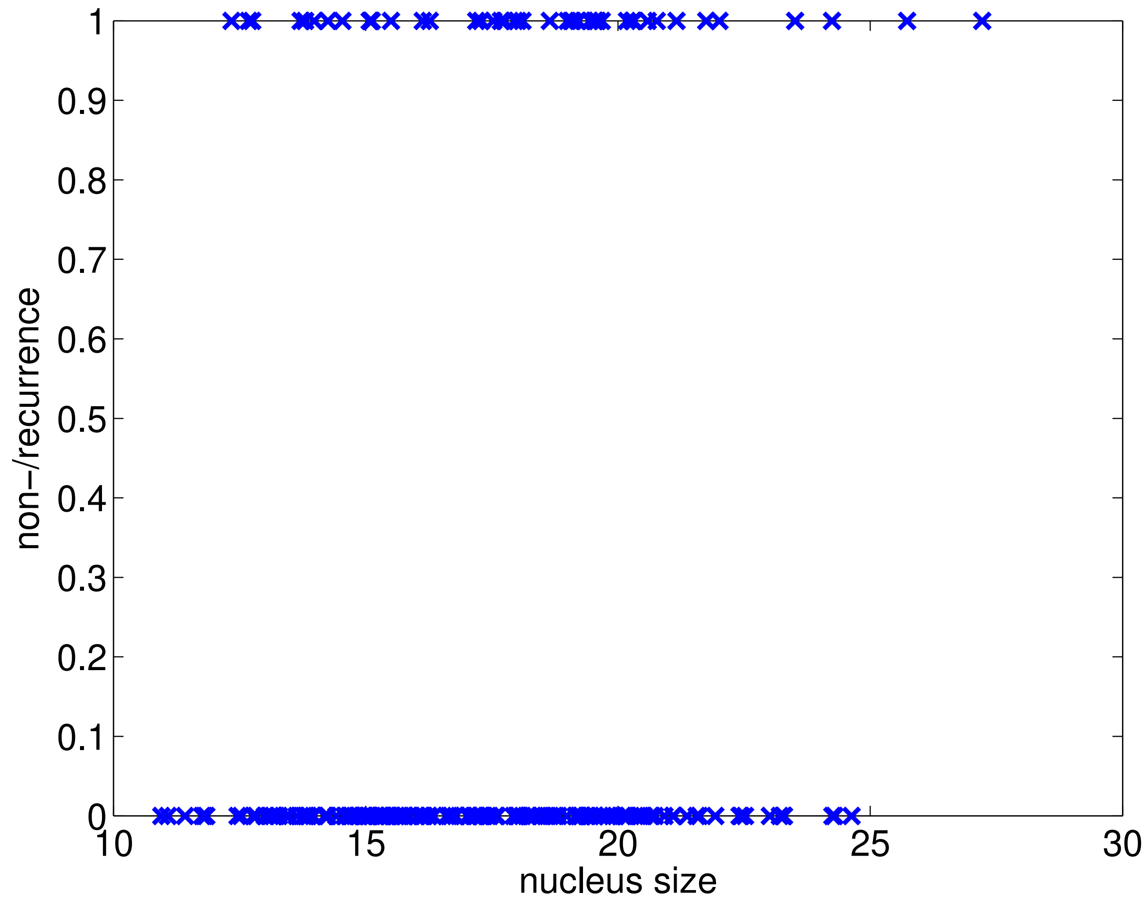
Classification so far

- Given a set of training data $\langle \mathbf{x}_i, y_i \rangle$, find a hypothesis h such that $h(\mathbf{x_i}) \approx y_i$ for as many examples as possible.
- More specifically, for binary classification, we considered that the hypothesis h could output the probability that $y_i = 1$ given $\mathbf{x_i}$.
- We could train a logistic classifier or ANN to minimize SSQ or CE.
- For multiclass classification, we could ask that h outputs the probabilities of each class. (We alluded to how this could be done with ANNs.)
- In any case, we're directly approximating $P(y|\mathbf{x})$.

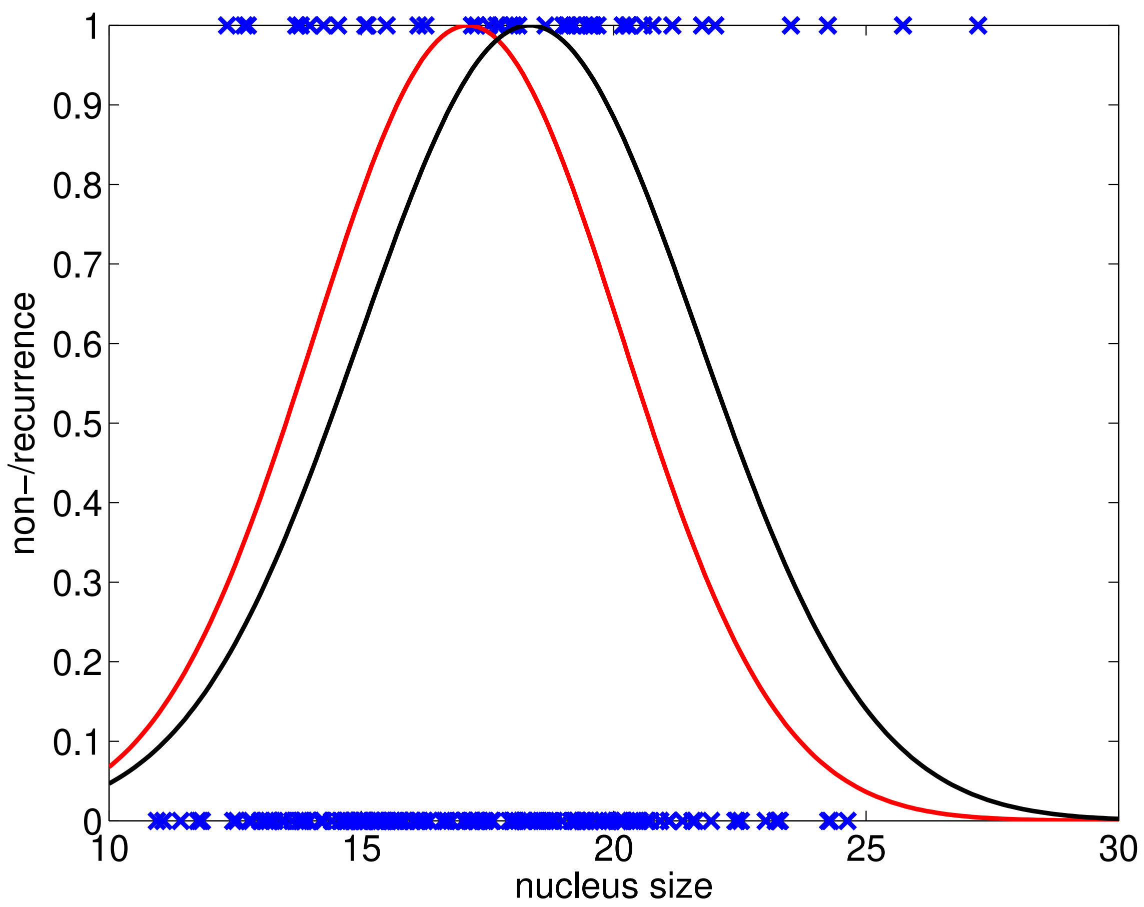
Generative Learning

- A different idea: use the data to build a *model* of the classes, in which we view the class label as "causing" a certain pattern of inputs.
- To classify a new example, we match it against the models for the two classes, then decide which model it resembles more
- This is called *generative learning*
- Usually you could use the model to *generate* (simulate) new data

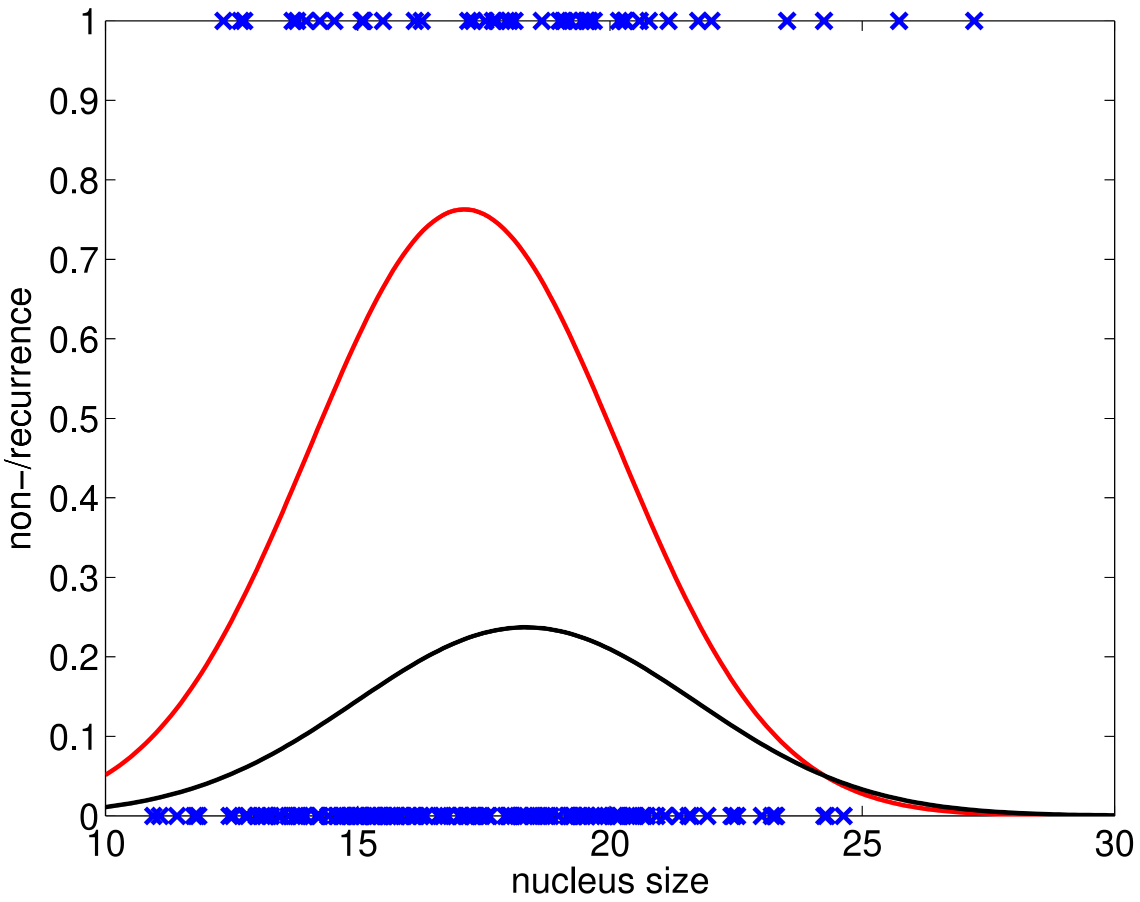
Recall our Wisconsin problem



Recall our Wisconsin problem



Recall our Wisconsin problem



More formally

- Suppose we have data of the form $\langle \mathbf{x}, y \rangle$, where $\mathbf{x}$ is a feature vector and y is a class label.
- Suppose that we want to predict the probability of the different classes
- Discriminative algorithms will try to model directly $P(y|\mathbf{x})$.
- Generative learning algorithms will try to model $P(y)$ and $P(\mathbf{x}|y)$ based on the data.
- For binary classification, $P(\mathbf{x}|y=0)$ and $P(\mathbf{x}|y=1)$ give the models of the two classes.
- $P(y)$ are called **class priors**