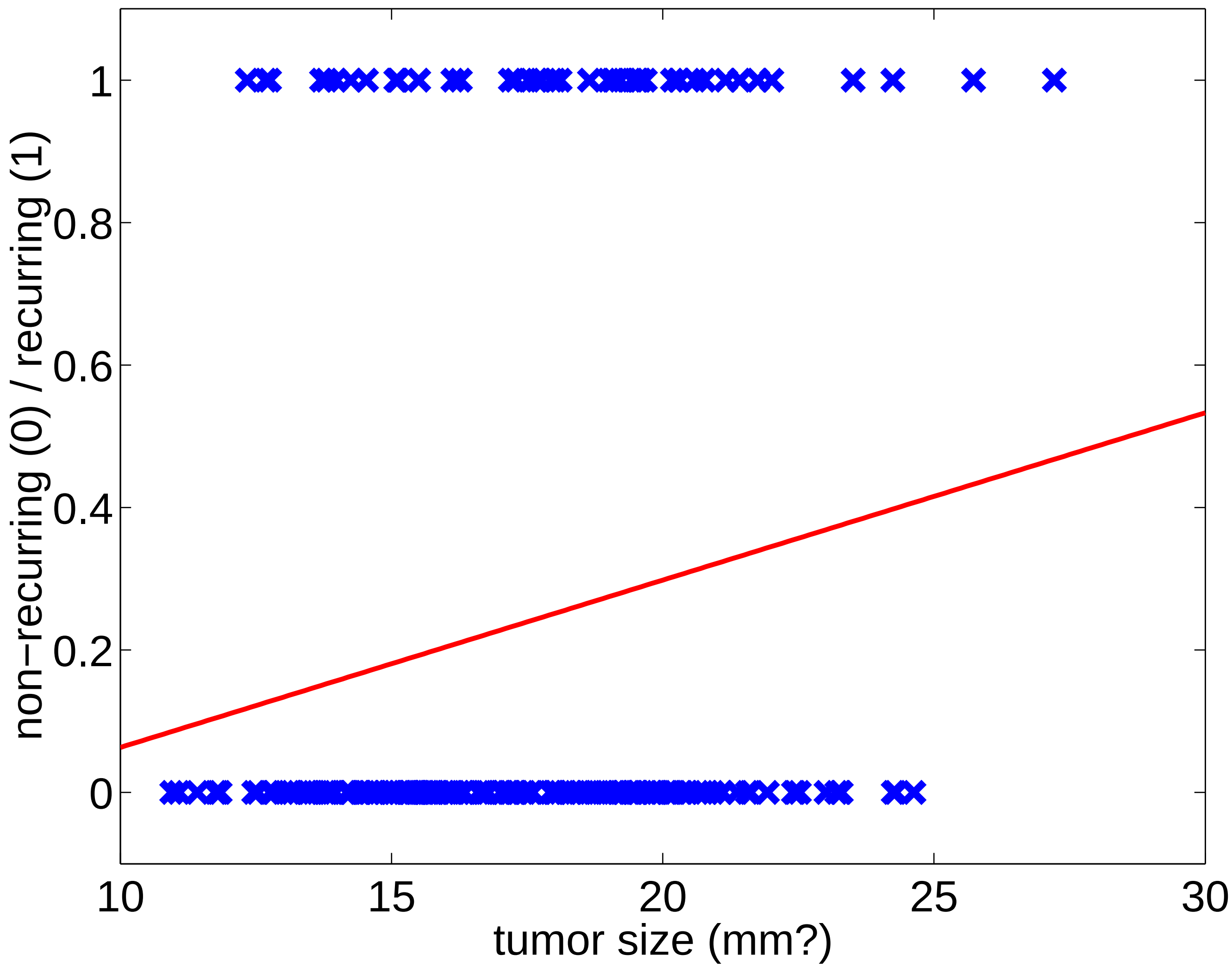
Today

- Revisit justification of sum squared error.
- $\left[\right.$Quasi-$\left.\right]$linear models for classification:
  - The perceptron
  - Logistic regression

[Quasi-]linear models for classification

- Recall: in a binary classification problem the outputs, $\mathbf{y}_i$, take one of two discrete values. (As convenient, we will assume they are -1 and $+1$, or 0 and 1.)
- Can we develop linear models for classification as we did for regression?
- What happens if we just apply linear regression as is?



1
0.8
0.6
0.4
0.2
0
10
15
20
25
30
tumor size (mm?)
non−recurring (0) / recurring (1)

Using linear regression for classification

- Sometimes it works okay. . .
- One issue: how do we interpret the output?
  - As a probability?
  - Or do we predict the most likely class?
- "Probabilities" greater than one or less than zero may be a problem.
- Another issue: what is the justification for minimizing sum squared error?

Two alternatives

- We can non-linearly transform the linear output.
- If we threshold it, typically as

$$\hat{f}(\mathbf{x}) = \text{sgn}(\mathbf{x} \cdot \mathbf{w}) = \begin{cases} +1 & \text{if } \mathbf{x} \cdot \mathbf{w} > 0 \\ -1 & \text{otherwise} \end{cases}$$

  then we have a *Perceptron*. The output is taken as the predicted class.

- In logistic regression, we use:

$$\hat{f}(\mathbf{x}) = \frac{1}{1 + e^{-\mathbf{x} \cdot \mathbf{w}}} \ ,$$

  the output of which is taken as the probability that $\mathbf{y} = 1$.

- Either way, $\mathbf{x} \cdot \mathbf{w}$ can be thought of as the “evidence for” class $+1$. (Positive=evidence for, negative=evidence against.)

Perceptrons

The Perceptron

- We seek $\mathbf{w}$ which maximize the number of correctly classified samples. ($\mathcal{E}$=number of samples misclassified.)
- Correctly classifying sample i means $\mathbf{y}_i(\mathbf{x}_i \cdot \mathbf{w}) > 0$, where $\mathbf{y}_i \in \{-1, 1\}$.
- How do we find an optimal $\mathbf{w}$?

The perceptron criterion

- Gradient descent on $\mathcal{E}$ is impossible — the gradient is zero everywhere.
- Linear programming (LP) can be used to find $\mathbf{w}$.
  - If $\mathcal{E} = 0$ for some $\mathbf{w}$, LP will find such a $\mathbf{w}$. (In this case, the data is called *linearly separable*.)
  - Otherwise, LP can find a $\mathbf{w}$ which minimizes *the perceptron criterion:*

$$\sum_{\{i:\mathbf{y}_i(\mathbf{x}_i\cdot\mathbf{w})<0\}} -\mathbf{y}_i(\mathbf{x}_i \cdot \mathbf{w})$$

- However, often gradient descent on the perceptron criterion is used.

The perceptron learning rule

- For example, stochastic gradient descent on the perceptron criterion:
  - Initialize $\mathbf{w}$ somehow.
  - While some misclassified samples remain:
    1. Choose a misclassified sample, i.
    2. $\mathbf{w} \leftarrow \mathbf{w} + \alpha \mathbf{y}_i \mathbf{x}_i$, where α is a step-size parameter.
- If the data is linearly separable, then under appropriate conditions on α this converges to a $\mathbf{w}$ with zero error.
- If the data is not linearly separable... convergence is not guaranteed?

Logistic regression

. . . will be presented later.