Today

- More on nearest neighbor (for classification and regression)
- Cross-validation
- Linear least-squares fitting, polynomial lest-square fitting

Recall – Wisconsin breast cancer data set

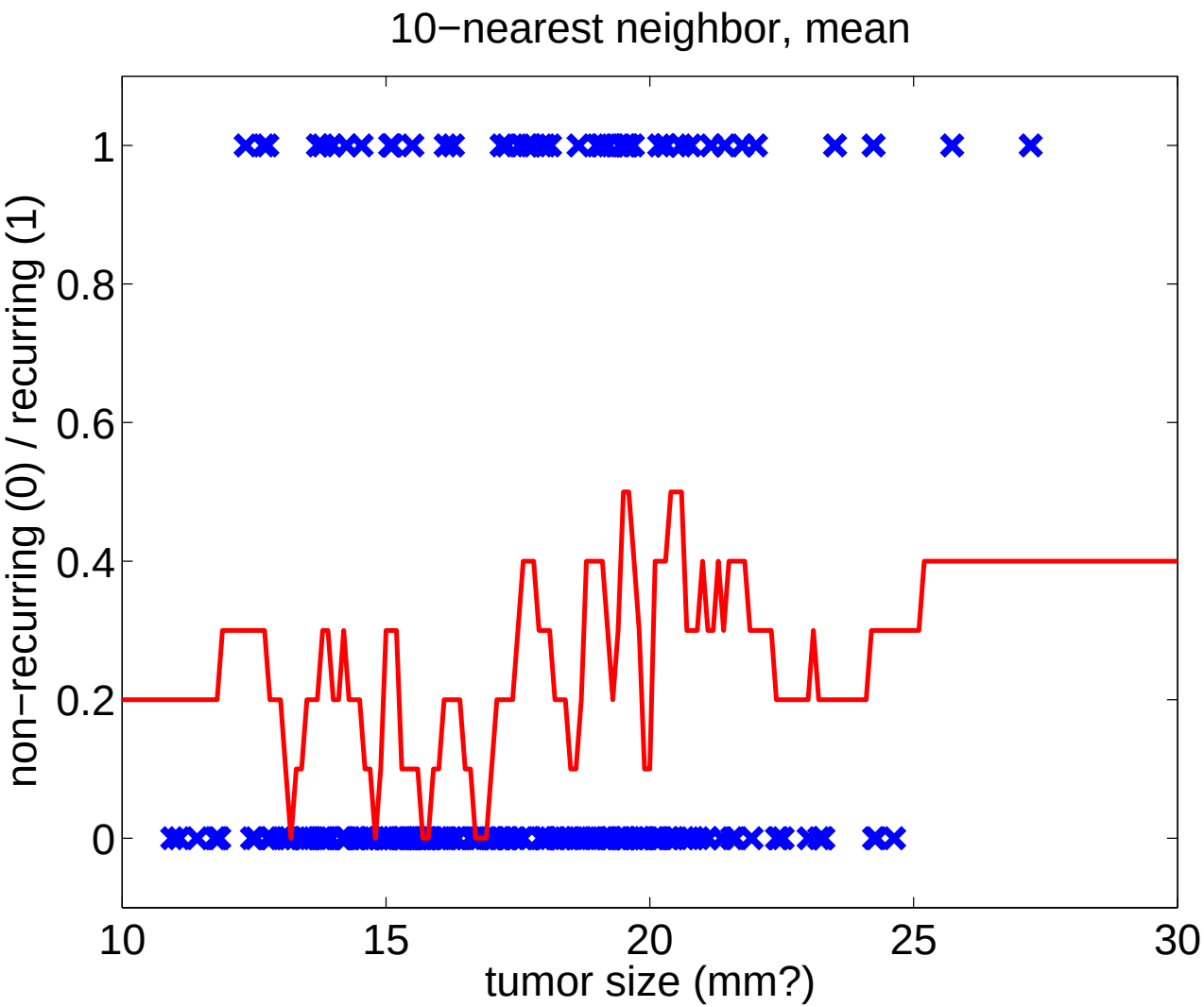
- Thirty real-valued variables per tumor that can be used for prediction.
- Two variables that can be predicted:
  - Outcome (R=recurrence, N=non-recurrence)
  - Time (until recurrence, for R, time healthy, for N).

tumor size	texture	perimeter	...	outcome	time
18.02	27.6	117.5		N	31
17.99	10.38	122.8		N	61
20.29	14.34	135.1		R	27
...					

Recall k-nearest neighbor

- Given: Training data $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^m$, distance metric d on $\mathcal{X}$.
- Learning: Nothing to do!
- Prediction: for $\mathbf{x} \in \mathcal{X}$
  - Find the k nearest training samples to $\mathbf{x}$.
    Let their indeces be $i_1, i_2, \ldots, i_k$.
  - Predict $\mathbf{y}$ =mean/median/mode of $\{\mathbf{y}_{i_1}, \mathbf{y}_{i_2}, \ldots, \mathbf{y}_{i_k}\}$.

Recall – predicting N/R based on tumor size



Problems

- The curve is jagged – piecewise constant.
- Zero probability is attached to some outcomes.
- What can we do?

Distance-weighted nearest neighbor

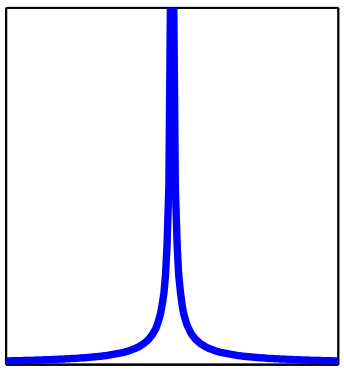
- Inputs: Training data $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^m$, distance metric d on $\mathcal{X}$, weighting function $w : \Re \mapsto \Re$.
- Learning: Nothing to do!
- Prediction: On input $\mathbf{x}$,
  - For each i compute $w_i = w(d(\mathbf{x}_i, \mathbf{x}))$.
  - Predict weighted majority or mean. For example,

$$\mathbf{y} = \frac{\sum_i w_i \mathbf{y}_i}{\sum_i w_i}$$

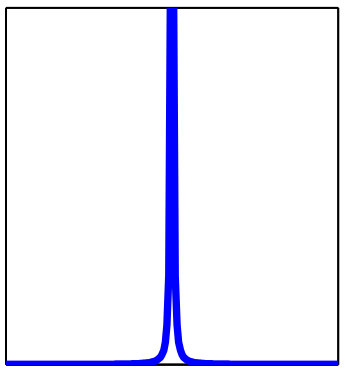
How to weight distances?

Some weighting functions

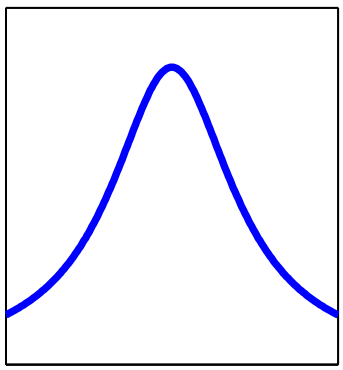
$$\frac{1}{d(\mathbf{x}_i, \mathbf{x})}$$



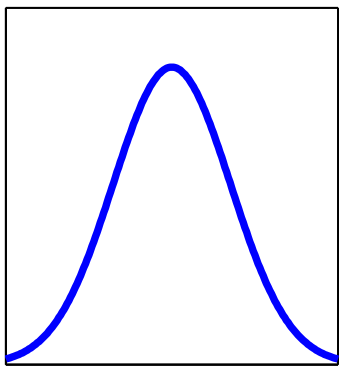
$$\frac{1}{d(\mathbf{x}_i, \mathbf{x})^2}$$

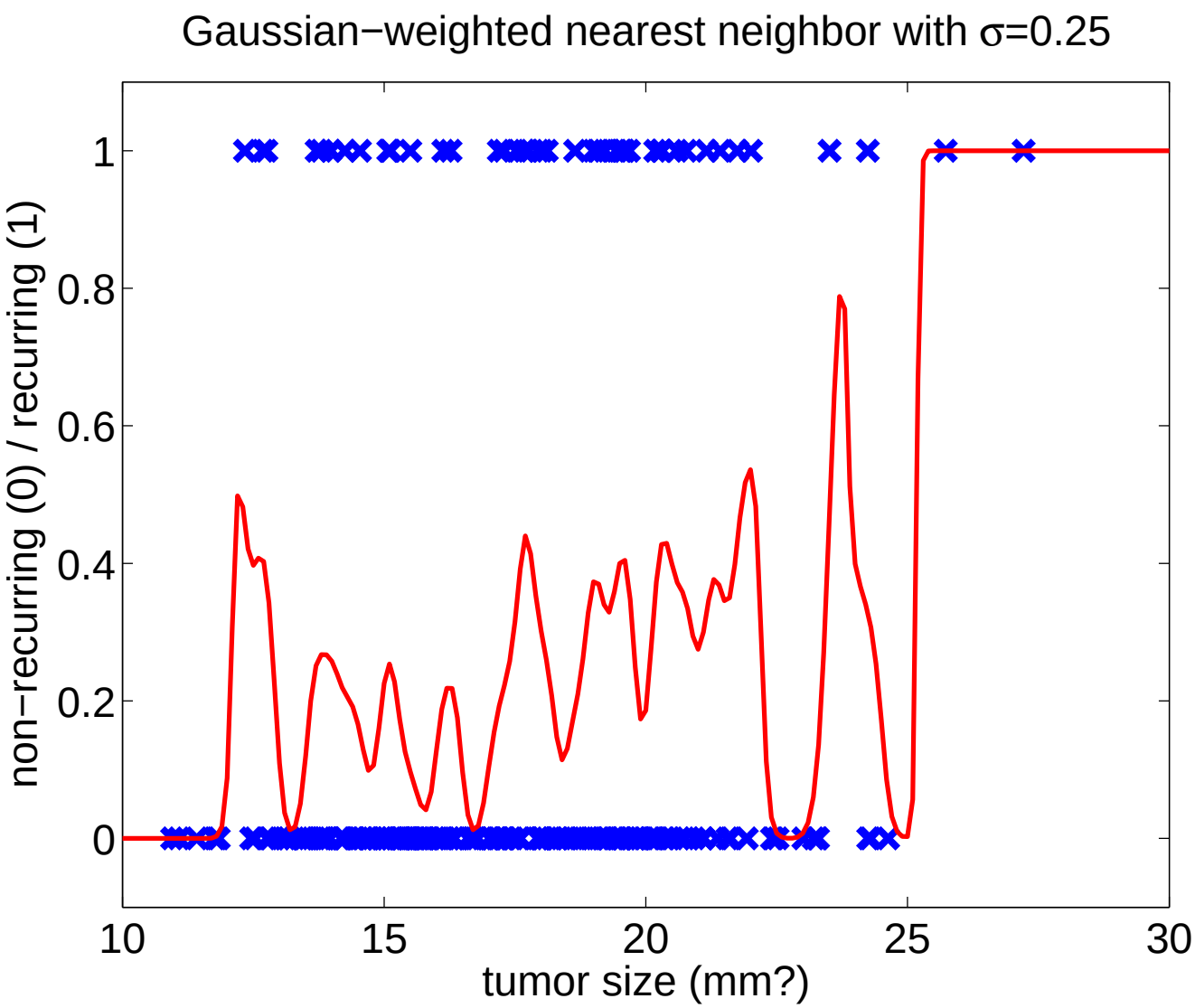


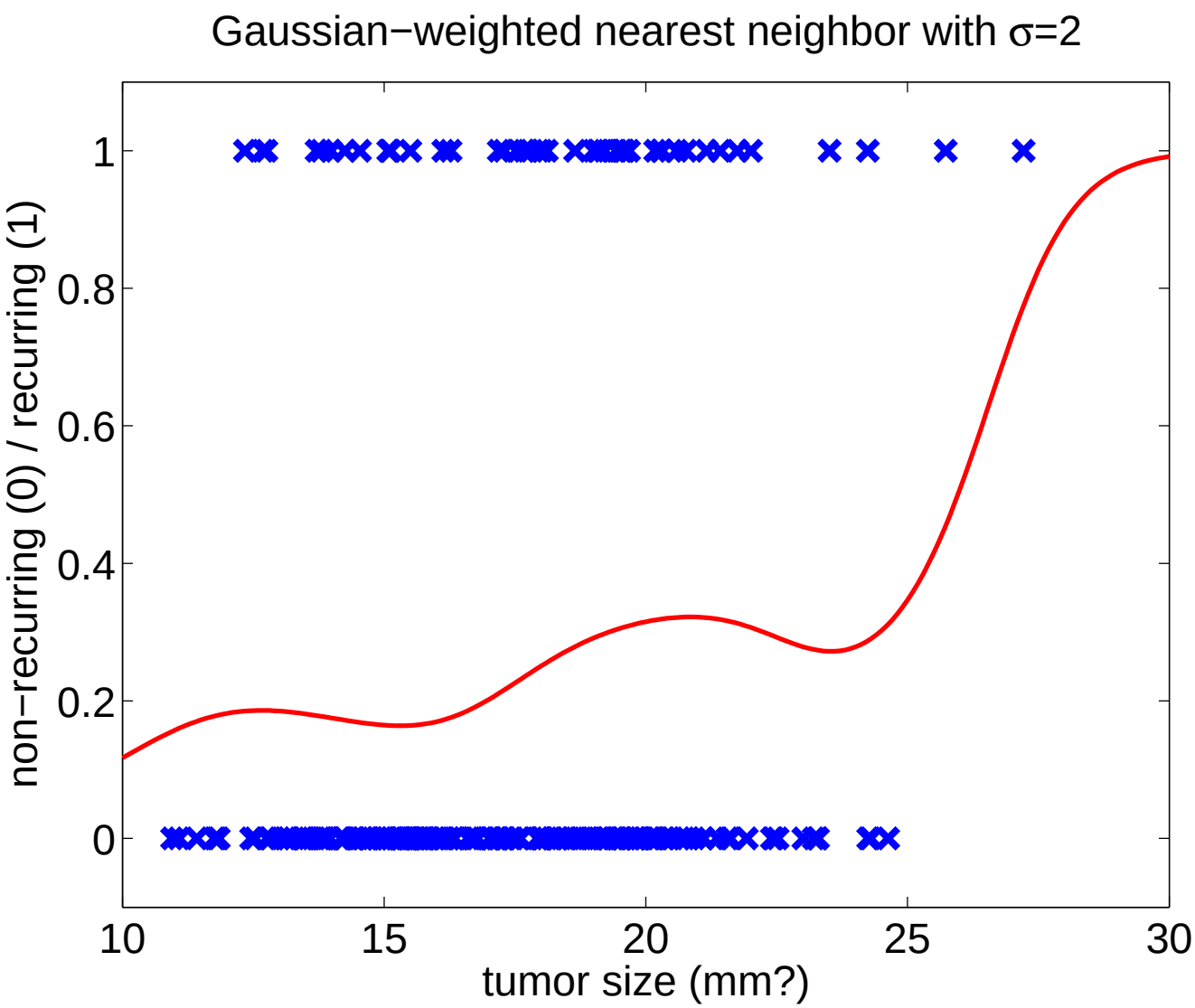
$$\frac{1}{c + d(\mathbf{x}_i, \mathbf{x})^2}$$

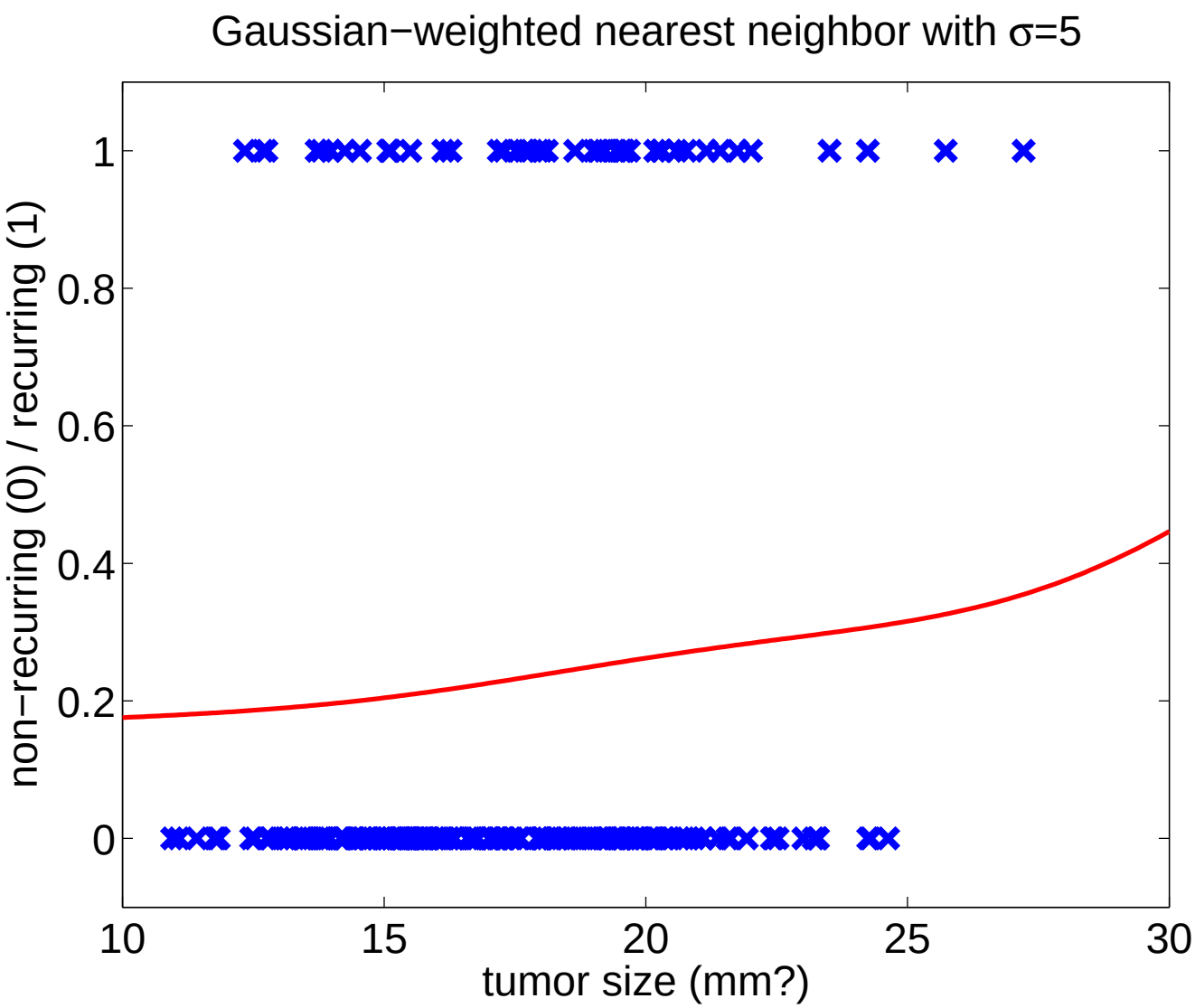


$$e^{-\frac{d(\mathbf{x}_i, \mathbf{x})^2}{\sigma^2}}$$




Gaussian−weighted nearest neighbor with σ=0.25
non−recurring (0) / recurring (1)
1
0.8
0.6
0.4
0.2
0
10
15
20
25
30
tumor size (mm?)


Gaussian−weighted nearest neighbor with σ=2
non−recurring (0) / recurring (1)
1
0.8
0.6
0.4
0.2
0
10
15
20
25
30
tumor size (mm?)


Gaussian−weighted nearest neighbor with σ=5
non−recurring (0) / recurring (1)
1
0.8
0.6
0.4
0.2
0
10
15
20
25
30
tumor size (mm?)

Cross-validation

Suppose we want to estimate the performance of a learning algorithm L, on a given data set $D = \{(\mathbf{x}_i, \mathbf{y}_i)\}$, with respect to expected prediction error $\mathcal{E}$.

$$\mathcal{E}(\hat{f}) = \int_{\mathbf{x}} \mathcal{E}_0(\mathbf{x}, \hat{f}(\mathbf{x}), f(\mathbf{x}))P(\mathbf{x})d\mathbf{x}$$

- We can divide D into a training set D_{train} and a validation set D_{valid}.
- Suppose L can be viewed as a function that maps a data set D to a function $L(D) = \hat{f} : \mathcal{X} \mapsto \mathcal{Y}$. Let $\hat{f} = L(D_{train})$.

Then:

$$\mathcal{E}(L(D)) \approx \frac{1}{|D_{valid}|} \sum_{(\mathbf{x},\mathbf{y}) \in D_{valid}} \mathcal{E}_0(\mathbf{x}, \hat{f}(\mathbf{x}), \mathbf{y})$$

Cross-validation

- Leave-one-out cross validation averages m iterations of the previous procedure (where m is number of samples in data set), using for the ith iteration $D_{valid} = \{(\mathbf{x}_i, \mathbf{y}_i)\}$ and $D_{train} = D - D_{valid}$.

- k-fold cross-validation divides D into k roughly-equal sized sets $D_1, \ldots, D_k$, and performs k iterations where $D_{valid} = D_i$ and $D_{train} = D - D_i$ for the ith iteration.

- What if L is stochastic, so that it doesn't always produce the same $\hat{f}$ for a given data set D?

Linear and polynomial least-squares fits

Assumptions

- We assume that $\mathcal{X} = \Re^n$ and $\mathcal{Y} = \Re$.
- The data can be organized into a $m \times n$ matrix X and $m \times 1$ vector Y as

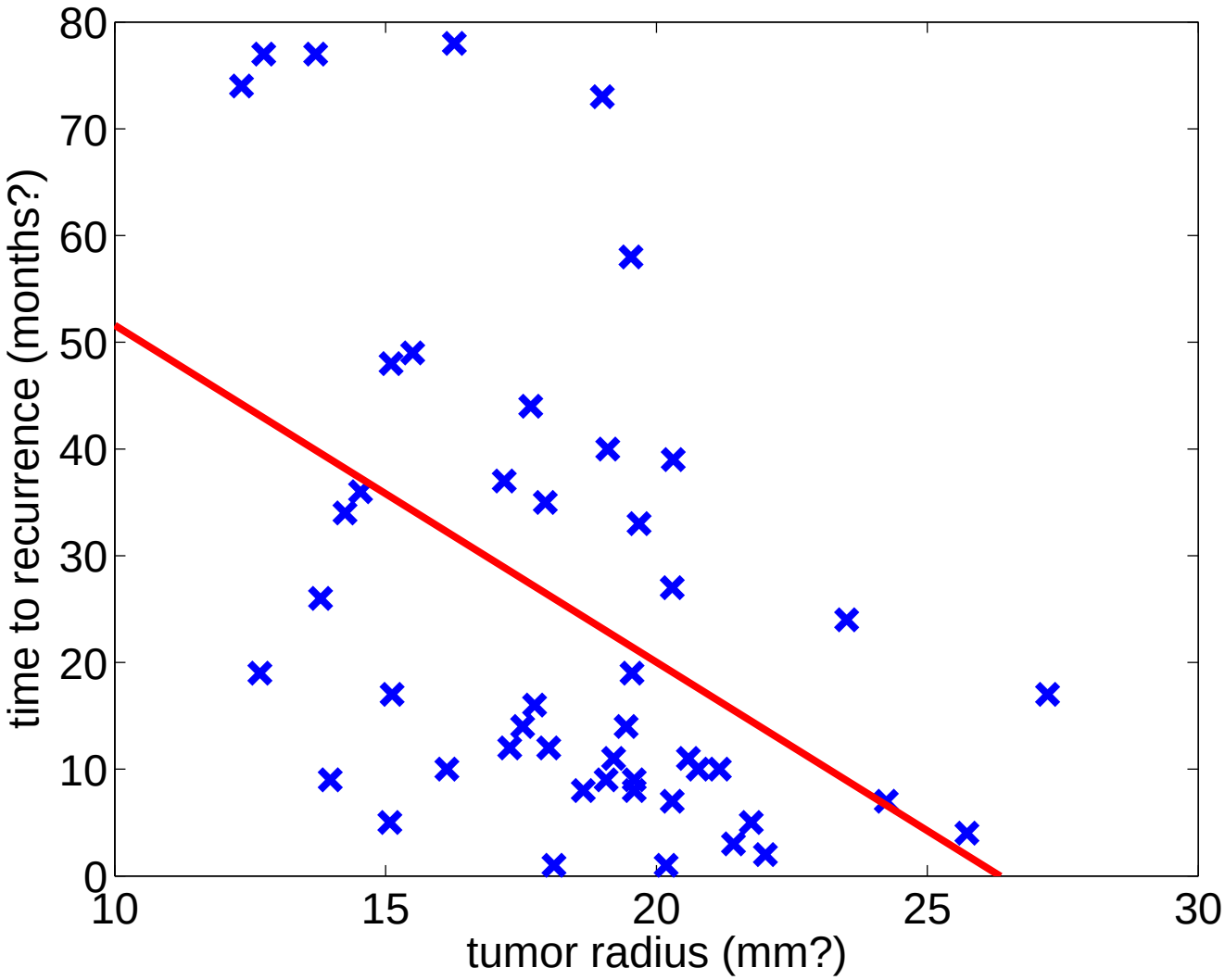
$$X = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_m \end{bmatrix} \qquad Y = \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \vdots \\ \mathbf{y}_m \end{bmatrix}$$

- We want to find a linear (affine, really) function of the $\mathbf{x}$'s that predicts the $\mathbf{y}$'s. Informally, find a $n \times 1$ vector $\mathbf{w}$ of "feature weights" such that

$$X\mathbf{w} + \mathbf{w}_0 \approx Y$$

- Can be written $X\mathbf{w} \approx Y$ by appending a column of 1's to X.

Example: predicting recurrence time from tumor size



Least-squares criterion

- Specifically, $\mathbf{w}$ should minimize the least-squares criterion

$$SSQ = \sum_{i=1}^{m} (\mathbf{x}_i \mathbf{w} - \mathbf{y}_i)^2 \ ,$$

  which can also be written

$$SSQ = (X\mathbf{w} - Y)^T (X\mathbf{w} - Y)$$

- Why least-squares?
- How do we find $\mathbf{w}$?

Differentiate w.r.t. w

- What does the partial derivative look like?

$$\frac{\partial}{\partial \mathbf{w}_i} SSQ$$

Differentiate w.r.t. w

- What does the partial derivative look like?

$$\frac{\partial}{\partial \mathbf{w}_i} SSQ$$

- Answer: it is linear in the $\mathbf{w}_j$.
- We could solve by gradient descent, but . . .
- Because $\frac{\partial}{\partial \mathbf{w}_i} SSQ$ is linear in $\mathbf{w}$, we can set $\frac{\partial}{\partial \mathbf{w}_i} SSQ = 0$ for all i.
- This gives us $(n+1)$ linear equations and $(n+1)$ unknowns.