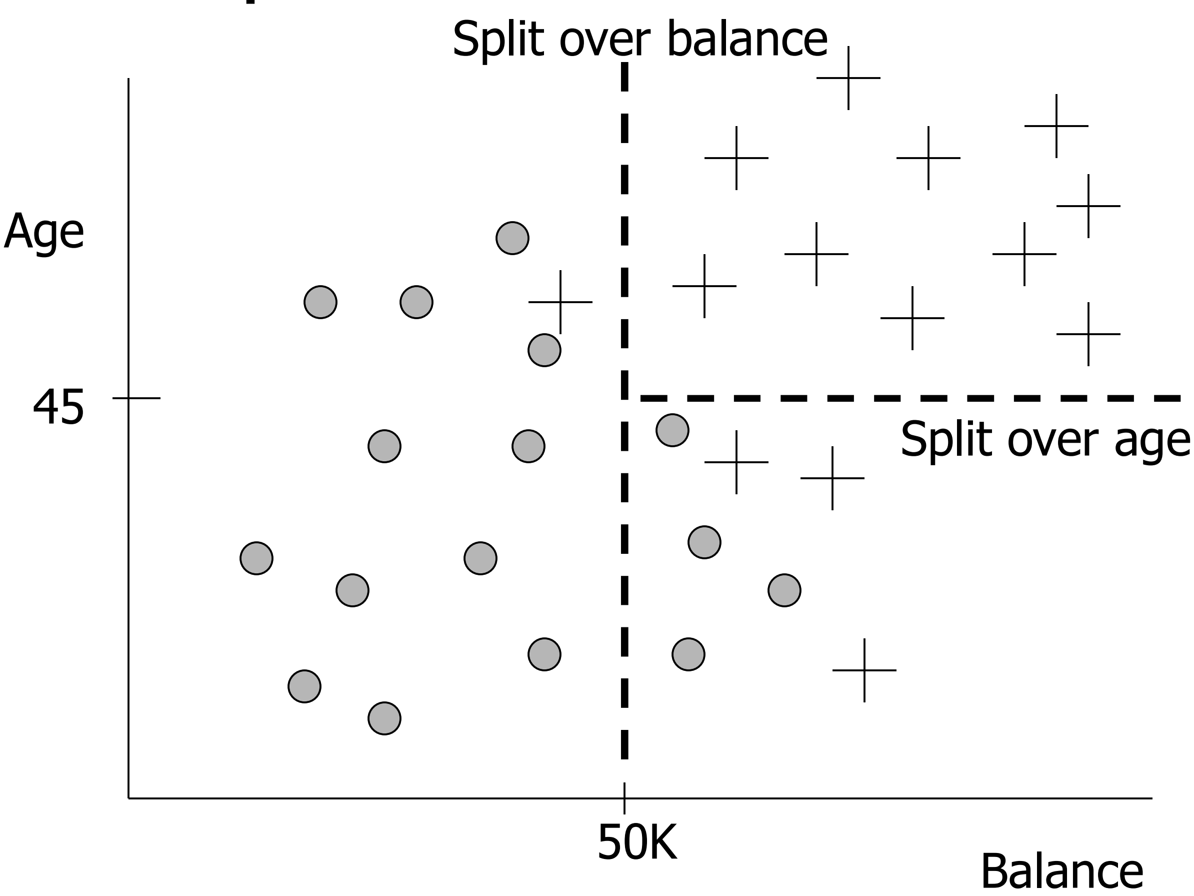
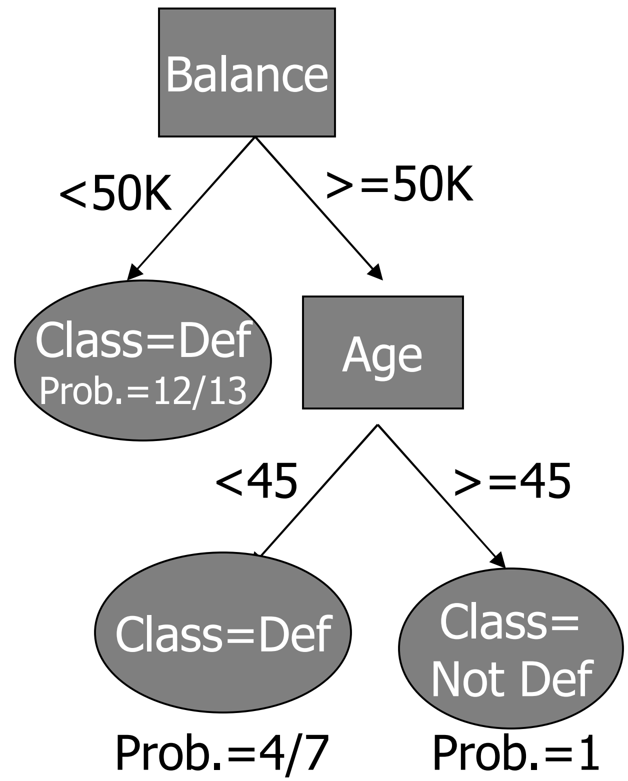
Linear Models

Source: Provost and Fawcett (2013)

Recall: Geometric interpretation of model

Classification tree partitions space of examples with axis-parallel decision boundaries



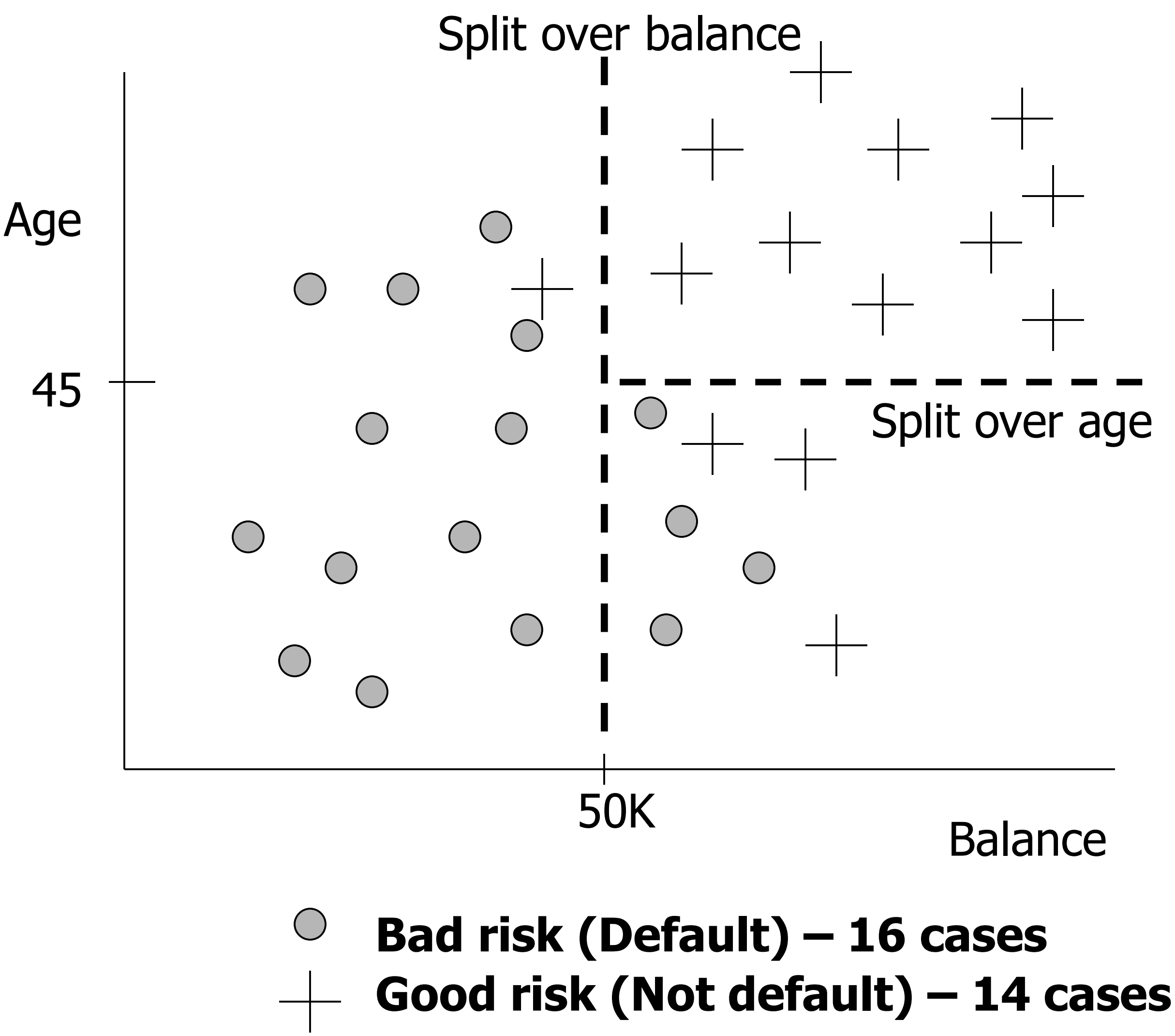


Bad risk (Default) – 16 cases

Good risk (Not default) – 14 cases

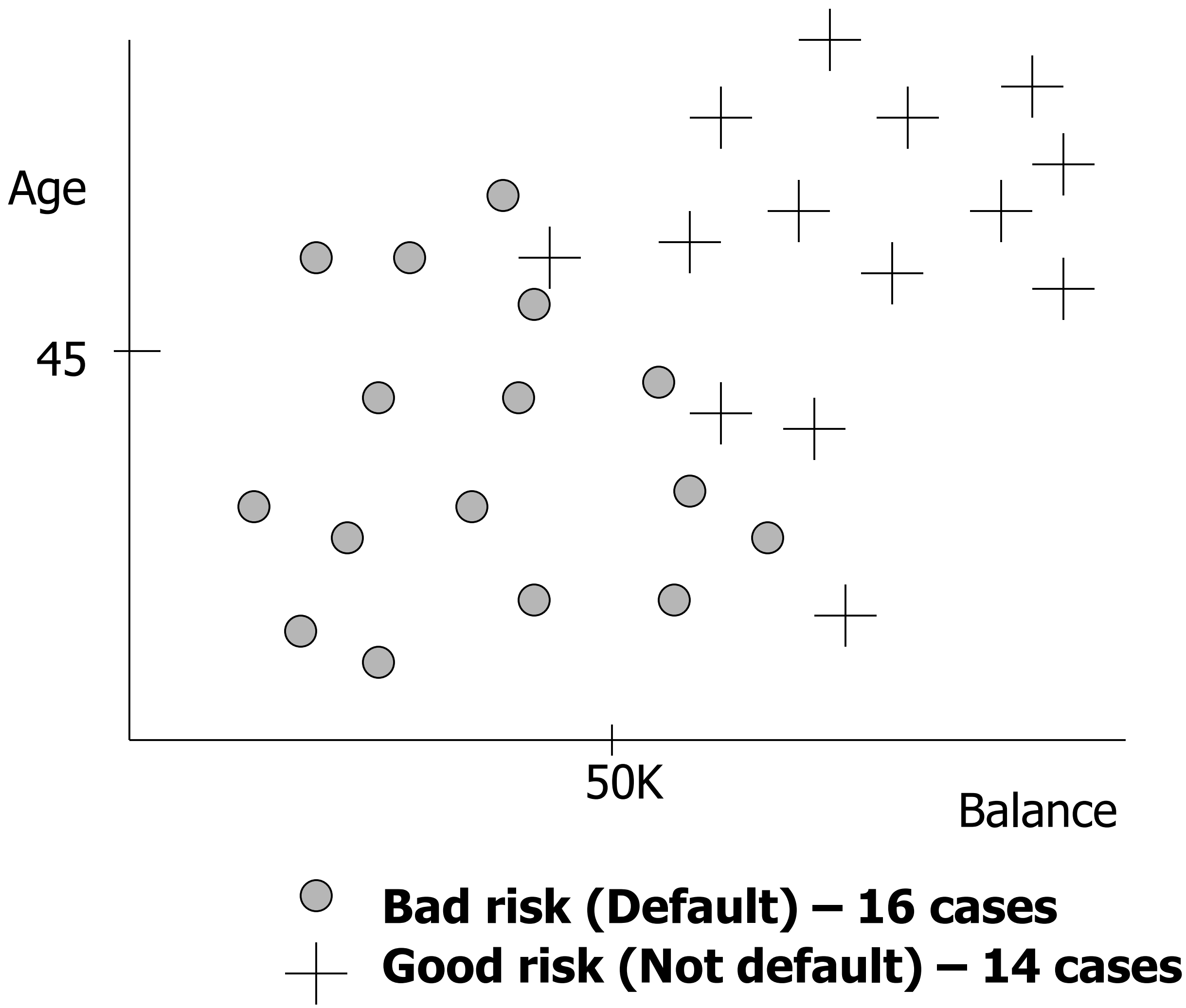
Geometric interpretation of model

What alternatives are there to partitioning <u>this way</u>?



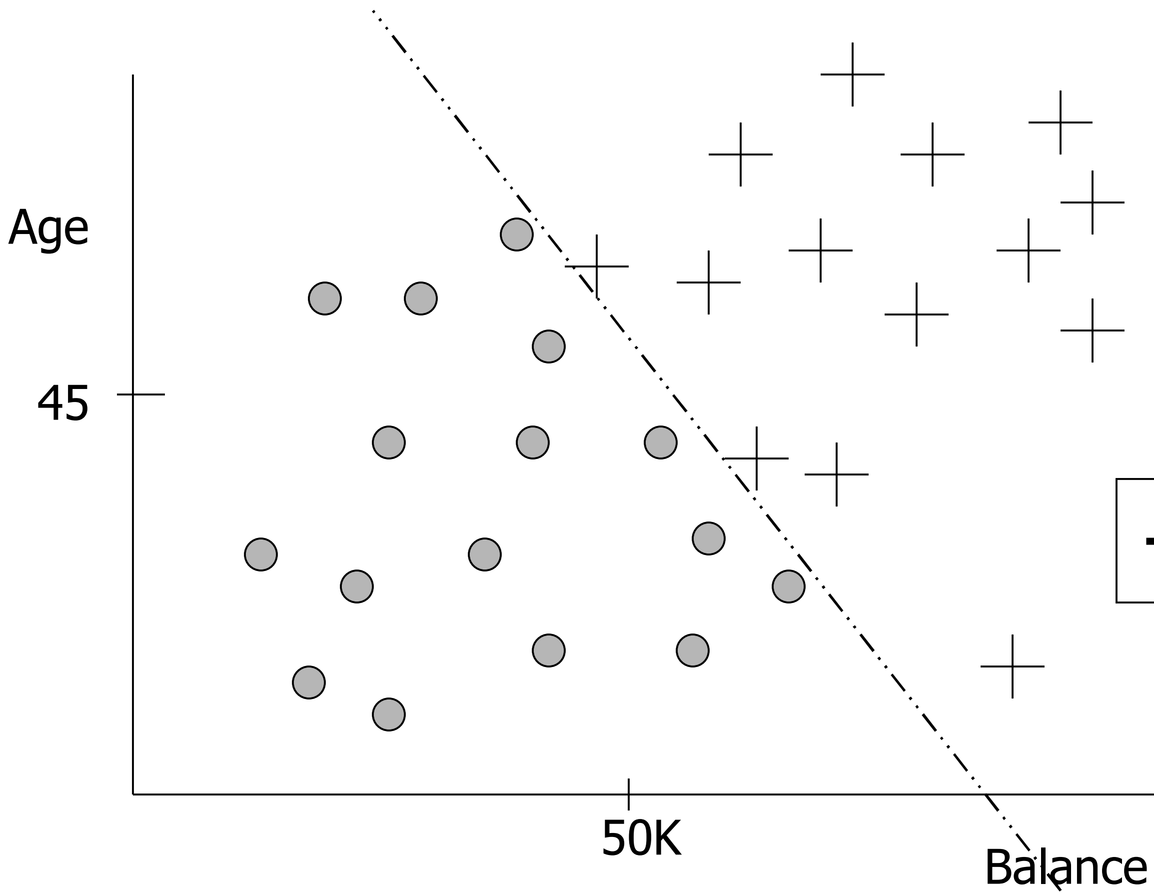
Geometric interpretation of model

What alternatives are there to DT partitioning?



Geometric interpretation

What alternatives are there to DT partitioning?



+ *if age > -2*Balance + 160*

Linear discriminant function; also basis of "support-vector machines"

Bad risk (Default) – 16 cases

Good risk (Not default) – 14 cases

Bayes Classifier

- Probability distribution P over $\mathcal{X} \times \{0,1\}$; let $(X,Y) \sim P$.
- Think of P as being comprised of two parts.
  1. Marginal distribution of X (a distribution over $\mathcal{X}$).
  2. Conditional distribution of Y given $X = x$, for each $x \in \mathcal{X}$:

$$\eta(x) \;:=\; P(Y = 1 \mid X = x)\,.$$

- The optimal classifier with smallest error rate (i.e., *Bayes classifier*) is

$$f^*(x) \;=\; \begin{cases} 0 & \text{if } \eta(x) \leq 1/2 \\ 1 & \text{if } \eta(x) > 1/2\,. \end{cases}$$

Logistic Regression

Suppose feature space is $\mathcal{X} = \mathbb{R}^d$.

Logistic regression: statistical model for $Y \mid \boldsymbol{X} = \boldsymbol{x}$ for each $\boldsymbol{x} \in \mathbb{R}^d$:

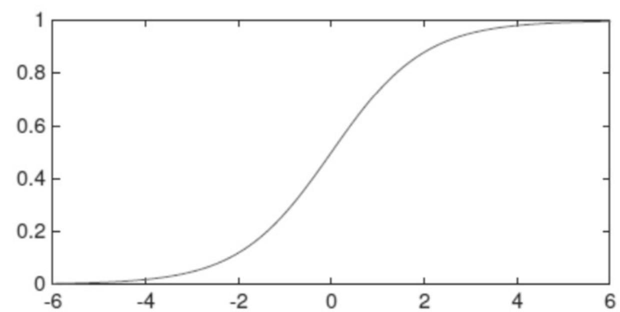
$$\mathcal{P} \;=\; \left\{ P_{(\beta_0,\boldsymbol{\beta})} : \beta_0 \in \mathbb{R},\, \boldsymbol{\beta} \in \mathbb{R}^d \right\},$$

where

$$\eta_{(\beta_0,\boldsymbol{\beta})}(\boldsymbol{x}) \;:=\; P_{(\beta_0,\boldsymbol{\beta})}(Y = 1 \mid \boldsymbol{X} = \boldsymbol{x}) \;=\; \operatorname{logistic}(\beta_0 + \langle \boldsymbol{\beta}, \boldsymbol{x} \rangle)$$

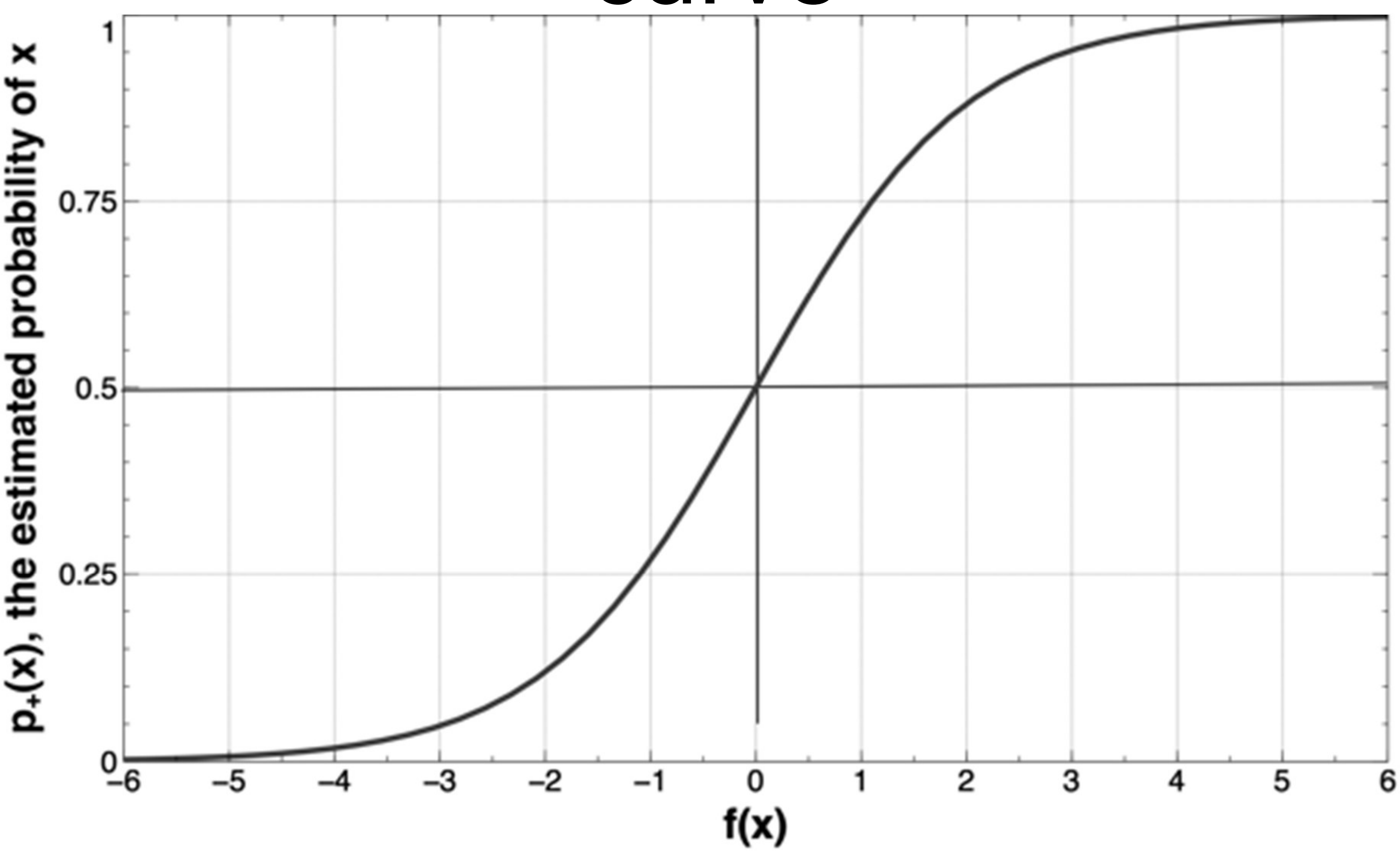
and

$$\operatorname{logistic}(z) \;:=\; \frac{1}{1 + e^{-z}} \;=\; \frac{e^z}{1 + e^z}\,.$$



(Note: Logistic regression does not specify marginal distribution for $\boldsymbol{X}$.)

Logistic regression ("sigmoid") curve



Logistic Regression

Log-odds function of $P_{(\beta_0,\boldsymbol{\beta})}$ is

$$\boldsymbol{x} \;\mapsto\; \log \frac{\eta_{(\beta_0,\boldsymbol{\beta})}(\boldsymbol{x})}{1-\eta_{(\beta_0,\boldsymbol{\beta})}(\boldsymbol{x})} \;=\; \log \frac{\frac{\exp(\beta_0+\langle\boldsymbol{\beta},\boldsymbol{x}\rangle)}{1+\exp(\beta_0+\langle\boldsymbol{\beta},\boldsymbol{x}\rangle)}}{\frac{1}{1+\exp(\beta_0+\langle\boldsymbol{\beta},\boldsymbol{x}\rangle)}} \;=\; \beta_0 + \langle \boldsymbol{\beta}, \boldsymbol{x}\rangle\,,$$

which is an **affine function**.

Bayes classifier for $P_{(\beta_0,\boldsymbol{\beta})}$ is

$$\boldsymbol{x} \;\mapsto\; \begin{cases} 0 & \text{if } \beta_0 + \langle \boldsymbol{\beta}, \boldsymbol{x}\rangle \le 0\,, \\ 1 & \text{if } \beta_0 + \langle \boldsymbol{\beta}, \boldsymbol{x}\rangle > 0\,. \end{cases}$$

Such classifiers are called **linear classifiers**.

The decision boundary separating the two predicted classes is the solution of β_0 + x β = 0, which is a point if x is one dimensional, a line if it is two dimensional, etc.

The distance from the decision boundary is $\beta_0/\|\beta\|$ + x $\beta_0/\|\beta\|$.

Logistic regression indicates that the class probabilities depend on distance from the boundary and that they go towards 0 and 1 more rapidly when $\|\beta\|$ is larger.

Logistic Regression: parameter estimation with Maximum Likelihood Estimation (MLE)

Given data $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ (regarded as an iid sample), MLE for $(\beta_0, \boldsymbol{\beta})$ is

$$
\begin{aligned}
(\hat{\beta}_0, \hat{\boldsymbol{\beta}}) \quad &= \quad \underset{\beta_0 \in \mathbb{R},\, \boldsymbol{\beta} \in \mathbb{R}^d}{\arg\max} \log \prod_{i=1}^n \eta_{(\beta_0, \boldsymbol{\beta})}(\boldsymbol{x}_i)^{y_i} \left(1 - \eta_{(\beta_0, \boldsymbol{\beta})}(\boldsymbol{x}_i)\right)^{1-y_i} \\
&\vdots \\
&= \quad \underset{\beta_0 \in \mathbb{R},\, \boldsymbol{\beta} \in \mathbb{R}^d}{\arg\max} \sum_{i=1}^n y_i(\beta_0 + \langle \boldsymbol{\beta}, \boldsymbol{x}_i \rangle) - \log(1 + \exp(\beta_0 + \langle \boldsymbol{\beta}, \boldsymbol{x}_i \rangle)) .
\end{aligned}
$$

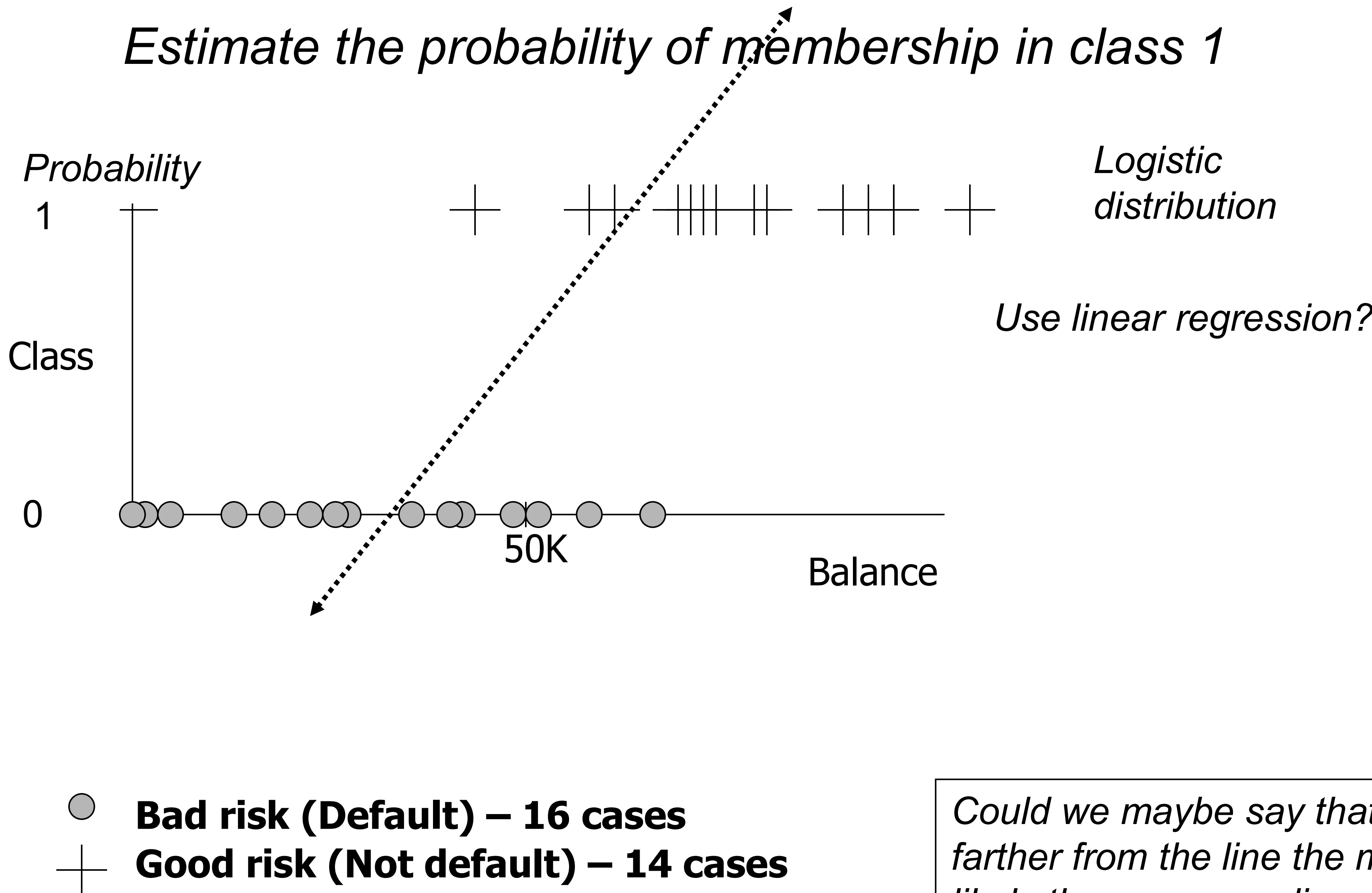
- No closed-form solution for MLE.
- But can apply convex optimization algorithms to get approximate maximizer of the MLE objective function (which is a function of $(\beta_0, \boldsymbol{\beta})$).
  - The negative of the above expression is the logistic loss or the cross-entropy loss which should be minimized.

Logistic regression is a misnomer

- The distinction between classification and regression is whether the value for the **target variable is categorical or numeric**
- For logistic regression, the model produces a numeric estimate
- However, **the values of the target variable in the data are categorical**
- Logistic regression is estimating the probability of class membership (a numeric quantity) over a **categorical class**
- Logistic regression is a **class probability estimation model** and not a regression model

A simpler case (one independent variable)

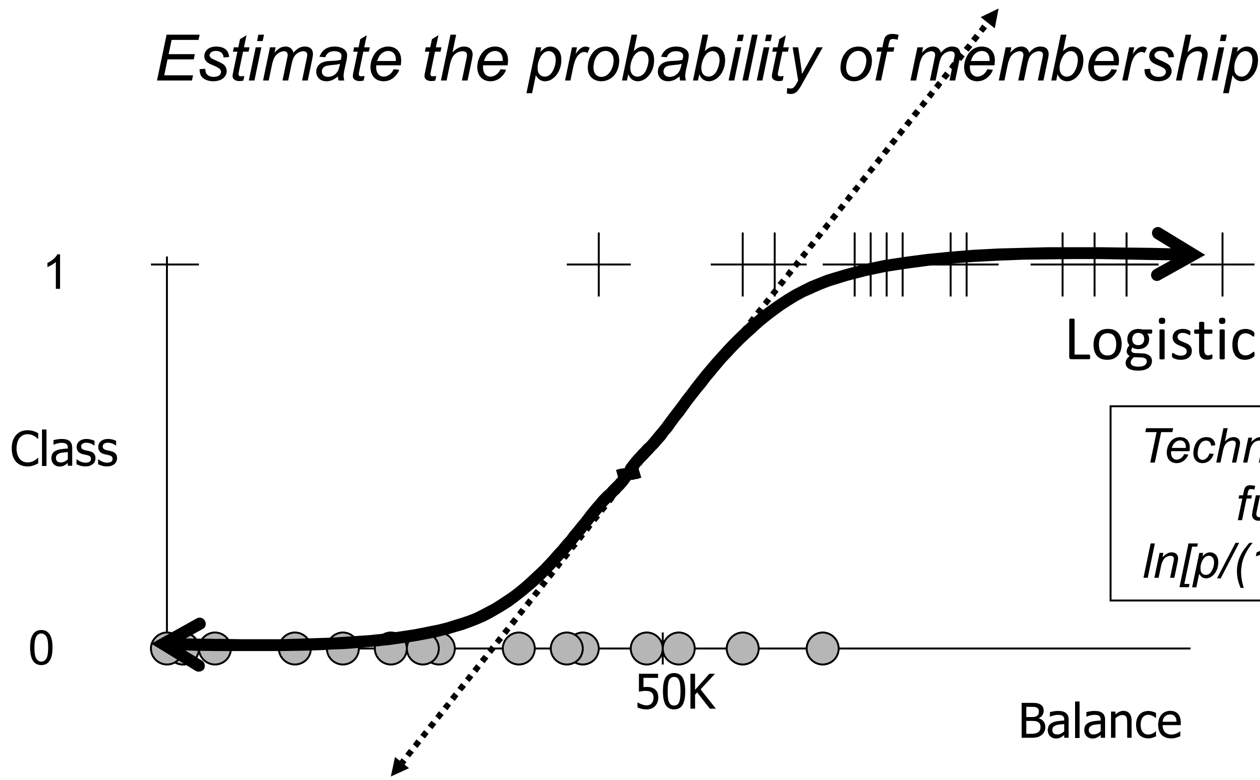
Estimate the probability of membership in class 1



Could we maybe say that the farther from the line the more likely the corresponding class?

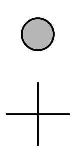
A simpler case (one independent variable)

Estimate the probability of membership in class 1



Technically: "log odds" is a linear function of the features:
$\ln[p/(1-p)] = b_0 + b_1x_1 + b_2x_2 + \dots$

*FYI: Functions such as logistic regression are the basic building blocks of a "**neural network**"*



Bad risk (Default) – 16 cases

Good risk (Not default) – 14 cases

Linear Discriminant Function or Linear Classifier

- Linear discriminant function or linear classifier: specified by a weight vector $w \in R^d$ and threshold $t \in R$:

$$f_{\boldsymbol{w},t}(\boldsymbol{x}) \;:=\; \begin{cases} 0 & \text{if } \langle \boldsymbol{w}, \boldsymbol{x} \rangle \leq t\,, \\ 1 & \text{if } \langle \boldsymbol{w}, \boldsymbol{x} \rangle > t\,. \end{cases}$$

Interpretation: does a linear combination of input features exceed a threshold?

$$\langle \boldsymbol{w}, \boldsymbol{x} \rangle \;=\; \sum_{i=1}^{d} w_i x_i \;\stackrel{?}{>}\; t\,.$$

Translation from logistic regression parameters $(\beta_0, \boldsymbol{\beta})$:

$$\boldsymbol{w} \;=\; \boldsymbol{\beta}\,, \qquad t \;=\; -\beta_0\,.$$

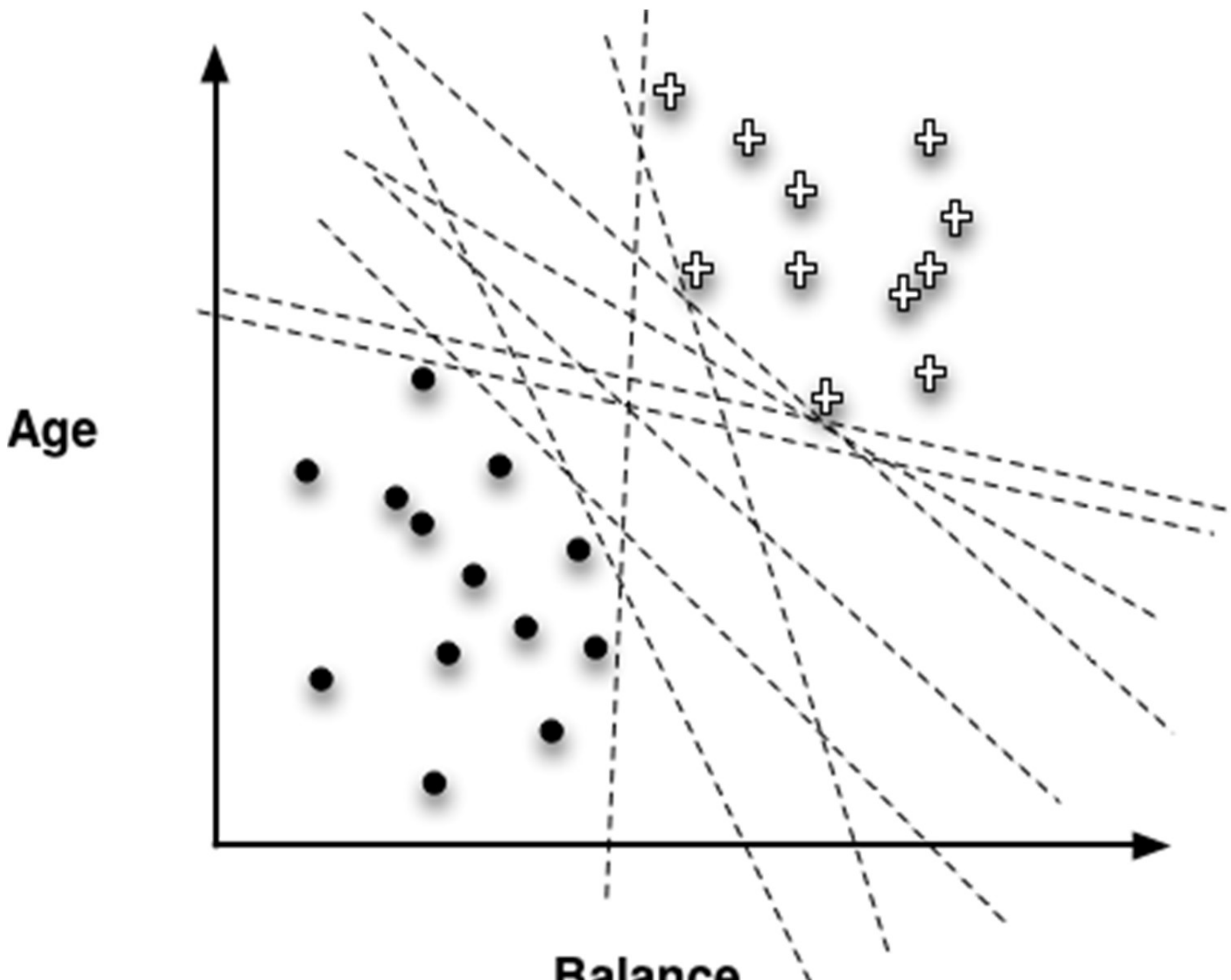
Example of Classification Function

- Linear discriminant function:

$$class(x) = \begin{cases} + \text{ if } 1.0 \times Age \ - 1.5 \times Balance + 60 > 0 \\ \bullet \text{ if } 1.0 \times Age \ - 1.5 \times Balance + 60 \leq 0 \end{cases}$$

- We now have a **parameterized model**: the weights of the linear function are the parameters
- The weights are often *loosely* interpreted as **importance indicators** of the features
- A different sort of multivariate supervised segmentation
    - The difference from DTs is that the method for taking multiple attributes into account is to create a mathematical function of them

Choosing the “best” line



Objective Functions

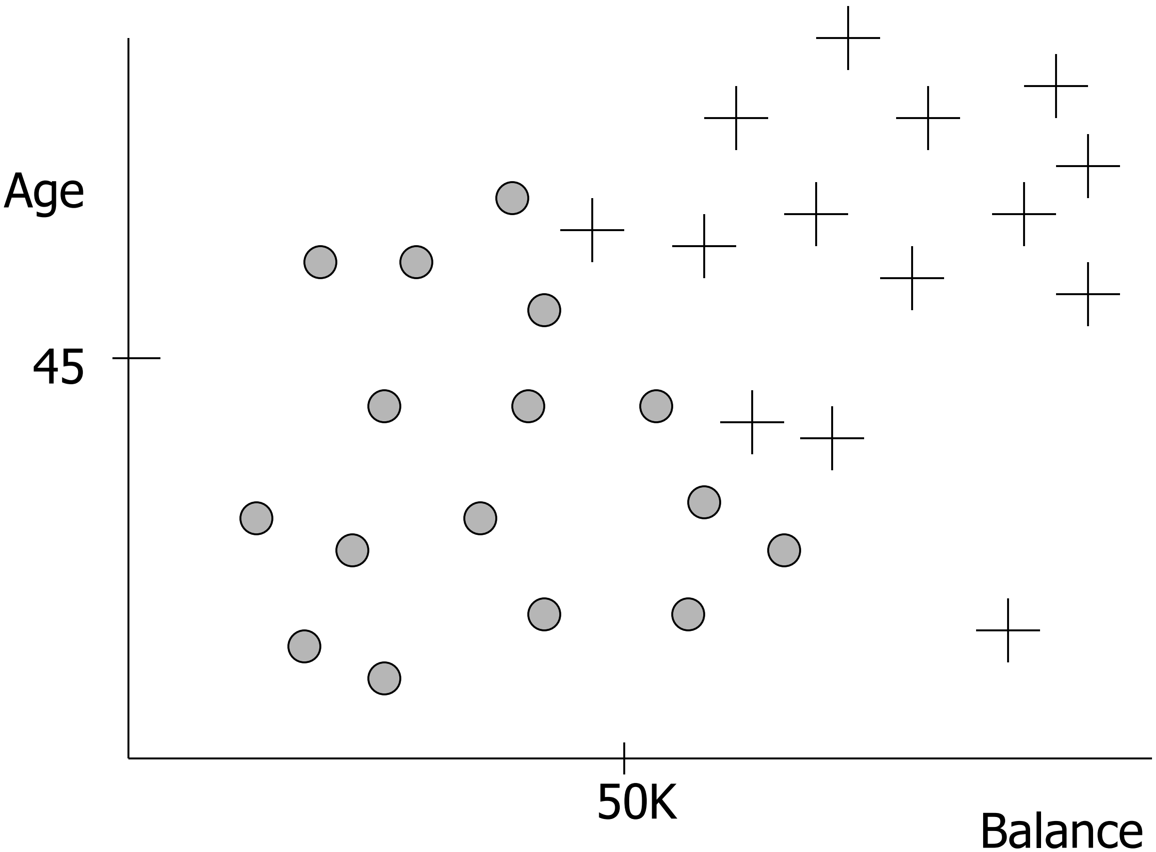
- "Best" line depends on the **objective (loss) function**
  - Objective function should represent our goal
- A loss function determines how much penalty should be assigned to an instance based on the error in the model's predicted value
- Examples of objective (or loss) functions:
  - $L(y;x) = |y - f(x)|$ Absolute loss
  - $\mathrm{L}(y;x) = \left(y - f(x)\right)^2$ Squared error loss: used for linear regression
  - $\mathrm{L}(y;x) = I(y \neq f(x))$ Zero-one loss
  - $\mathrm{L}(y;x) = \max(0, 1 - yf(x))$ Hinge loss: used for SVM
- **Linear regression**, **logistic regression**, and **support vector machines** are all very similar instances of our basic fundamental technique:
  - The key difference is that each uses **a different objective function**

Loss Functions

- **Zero-one loss** assigns a loss of zero for a correct decision and one for an incorrect decision
- **Squared error** specifies a loss proportional to the square of the distance from the boundary
  - Squared error loss usually is used for numeric value prediction (regression), rather than classification
  - The squaring of the error has the effect of greatly penalizing predictions that are grossly wrong

Geometric interpretation

What alternatives are there to DT partitioning?



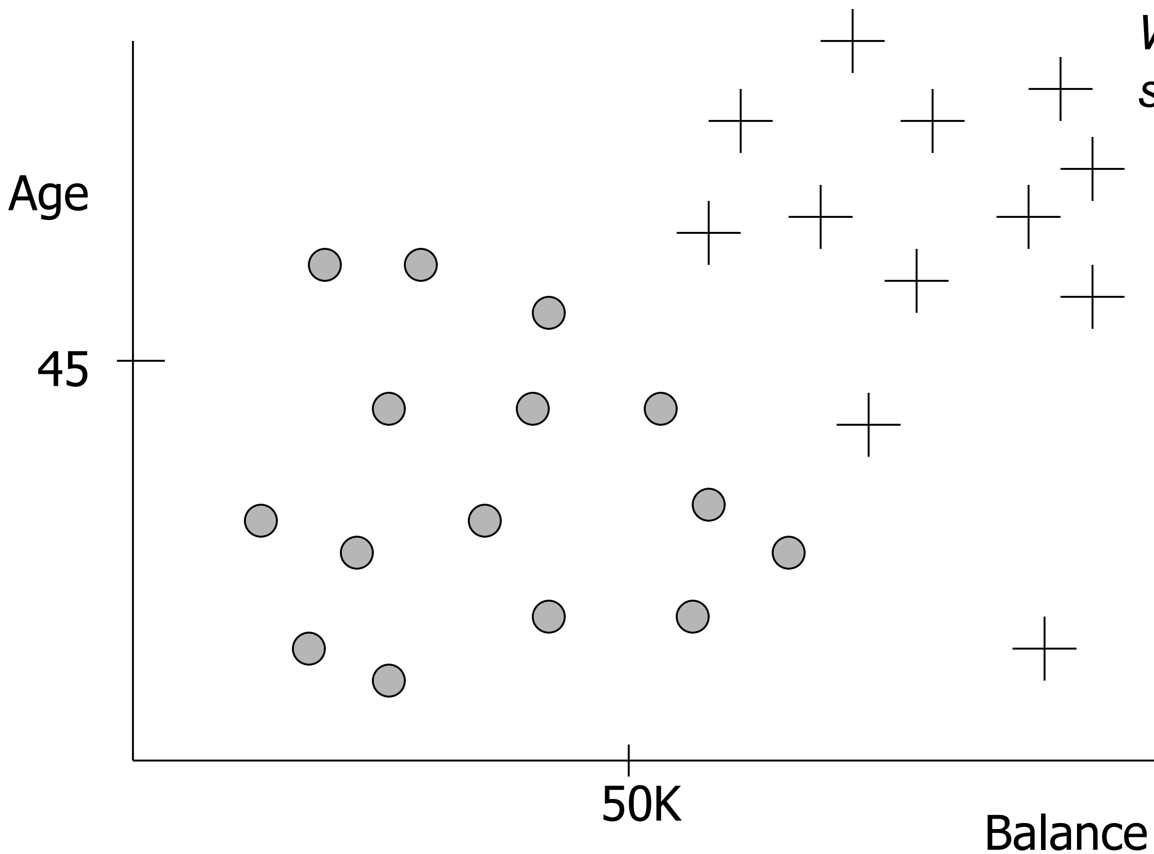
Bad risk (Default) – 16 cases

Good risk (Not default) – 14 cases

Brief digression on SVMs

Support-vector machines (SVMs) are linear discriminant functions

Which linear discriminant function should we choose?

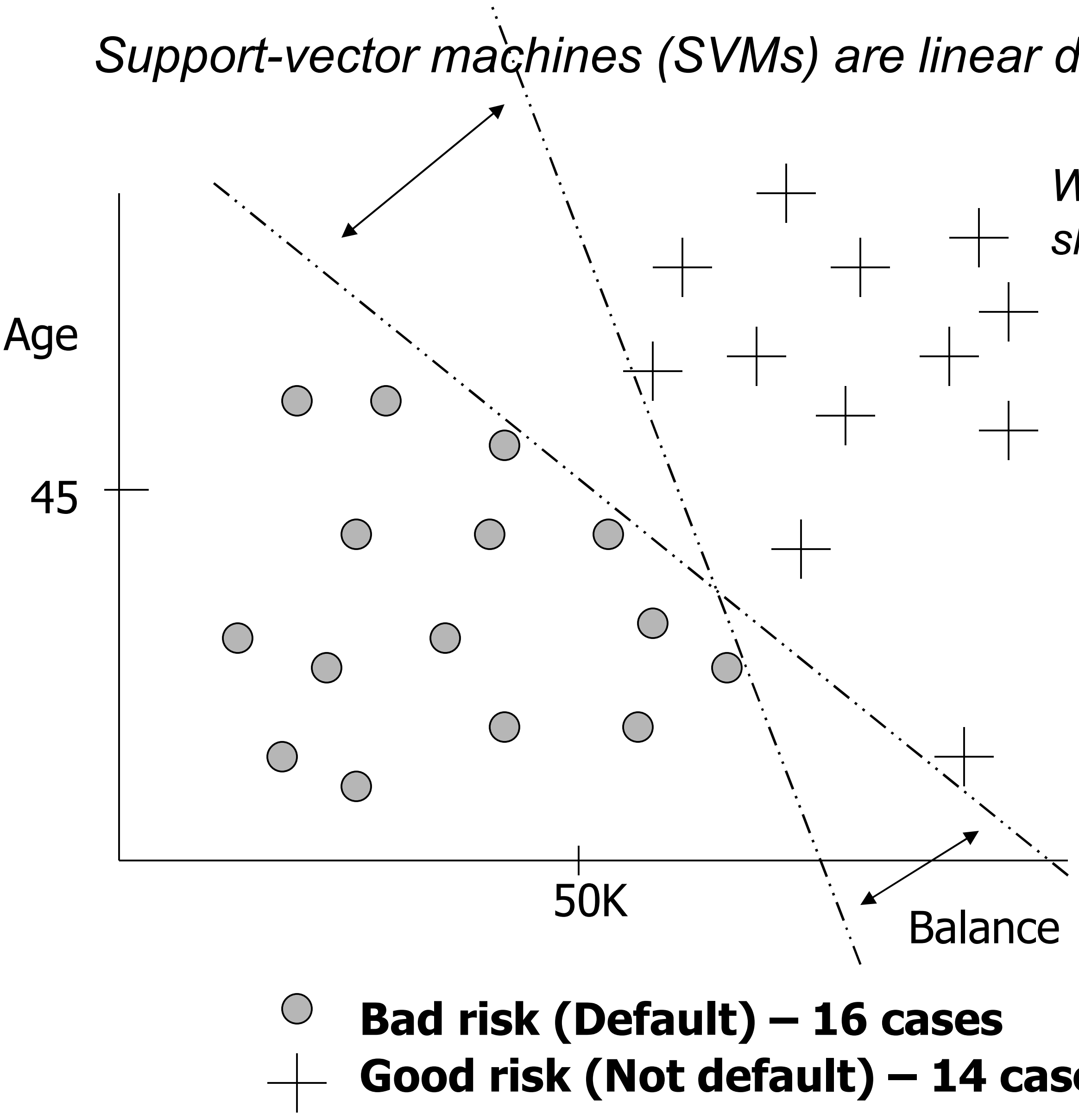


Bad risk (Default) – 16 cases

Good risk (Not default) – 14 cases

Brief digression on SVMs

Support-vector machines (SVMs) are linear discriminant functions

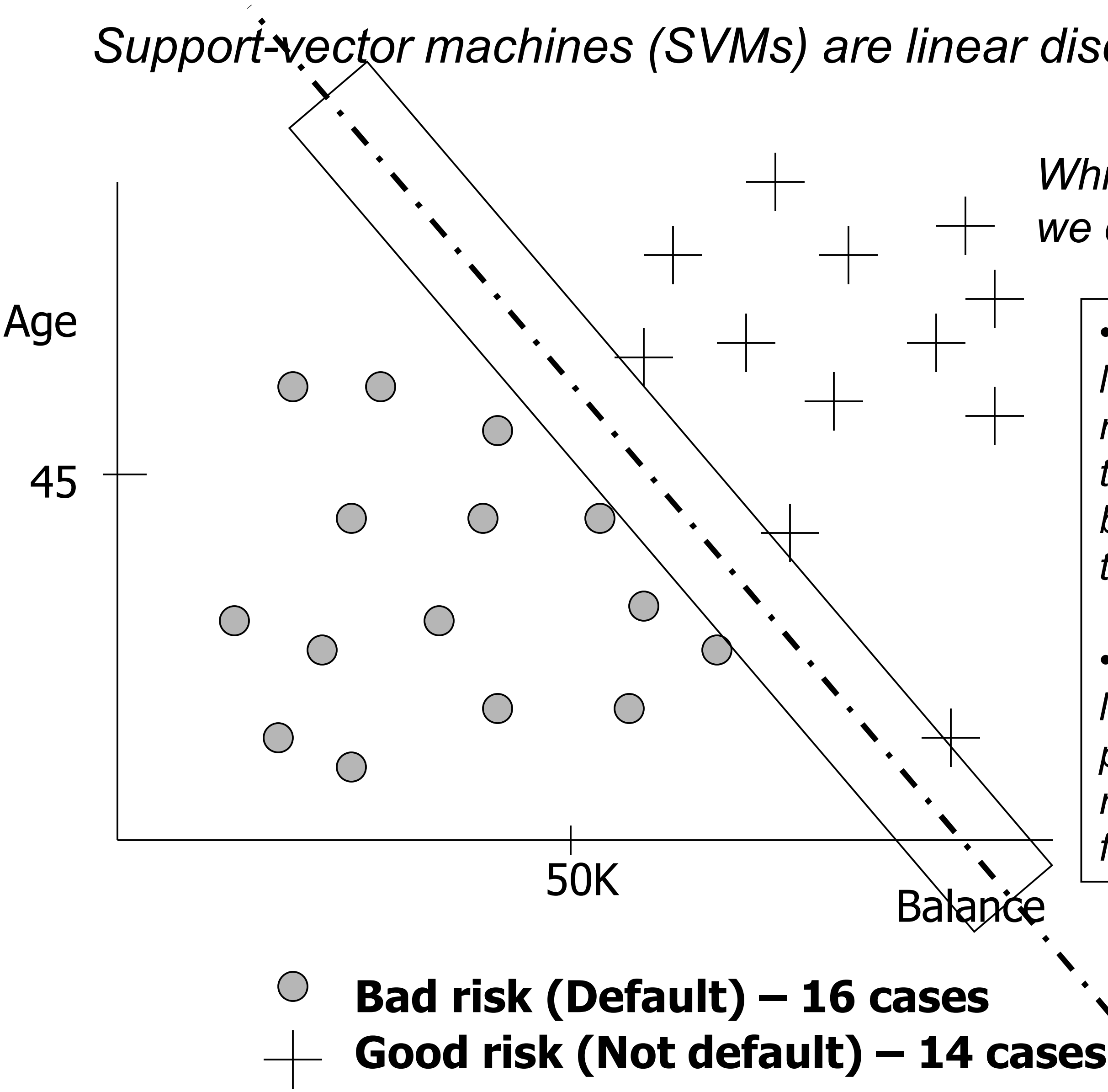


Which linear discriminant function should we choose?

Bad risk (Default) – 16 cases
Good risk (Not default) – 14 cases

Brief digression on SVMs

Support-vector machines (SVMs) are linear discriminant functions



Which linear discriminant should we choose?

- *SVMs choose the discriminant line as the center of the bar maximizes the margin between the classes (i.e.,the width of the bar that can be placed between the classes).*

- *SVMs also can handle non-linearly separable data by penalizing errors or by using a more complicated similarity function ("kernel").*

Support Vector Machines (SVMs)

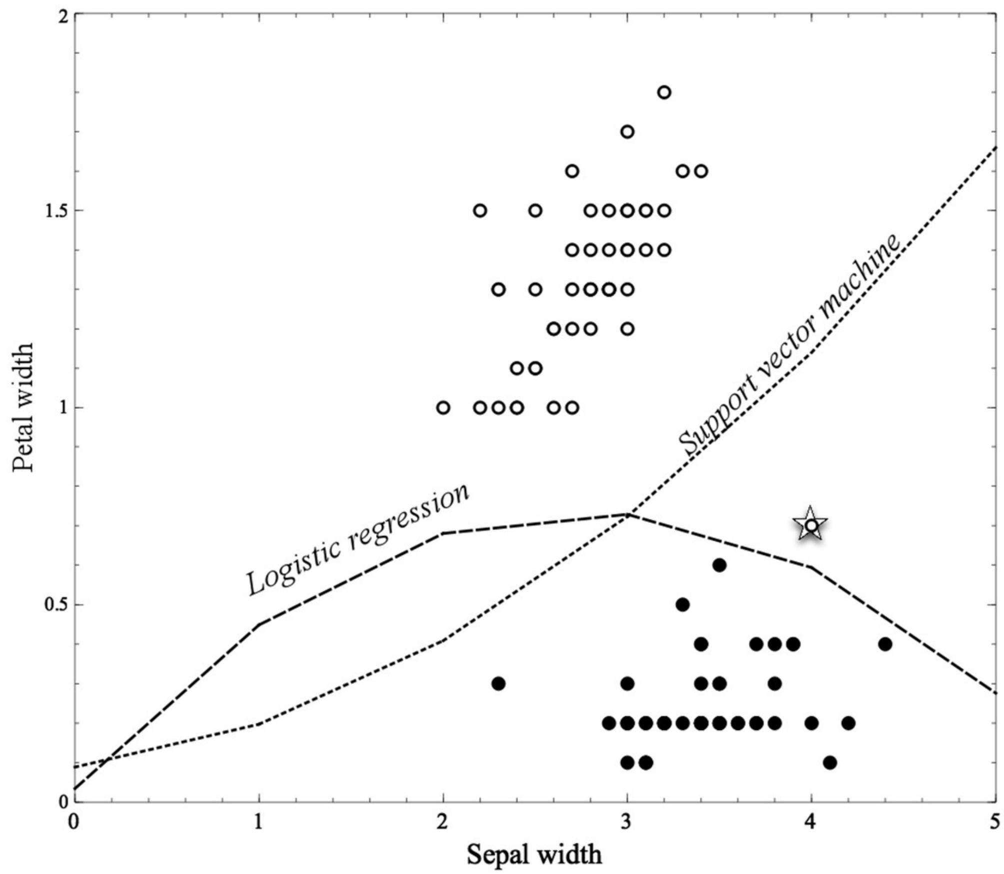
- Linear Discriminants
- Effective
- Use “hinge loss”
- Also, non-linear SVMs

Hinge Loss function: $\mathrm{L}(y; x) = \max(0, 1 - yf(x))$

- Support vector machines use **hinge loss**
- Hinge loss incurs no penalty for an example that is <u>not</u> on the wrong side of the margin
- The hinge loss only becomes positive when an example is on the wrong side of the boundary and beyond the margin
  - Loss then increases linearly with the example's distance from the margin
  - Penalizes points more the farther they are from the separating boundary

Non-linear Functions

- **Linear functions can actually represent nonlinear models**, if we include more complex features in the functions

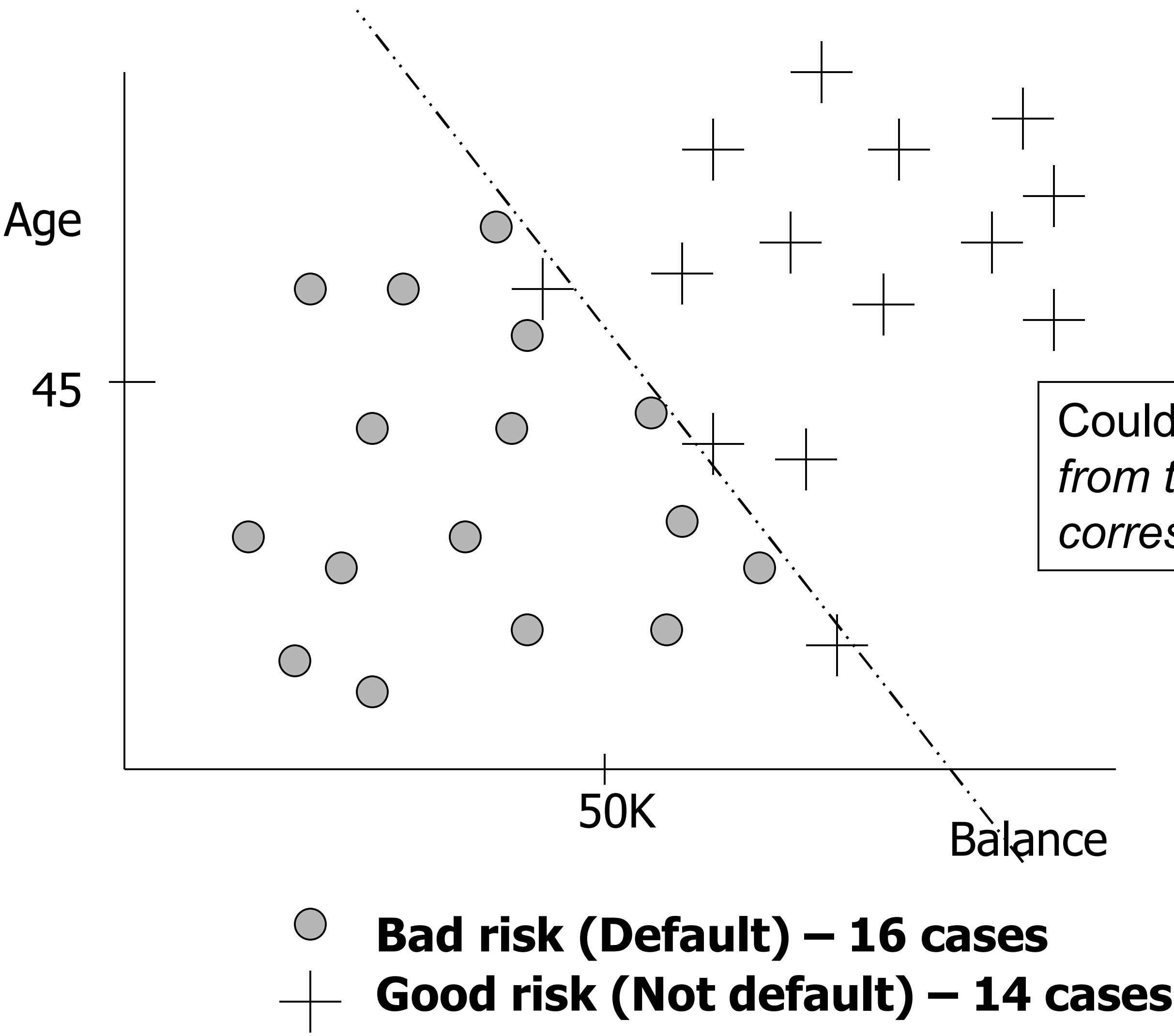


Non-linear Functions

- Using “higher order” features is just a “trick”
- Common techniques based on fitting the parameters of complex, nonlinear functions:
    - Non-linear support vector machines and neural networks
- **Nonlinear support vector machine** with a “polynomial kernel” consider “higher-order” combinations of the original features
    - Squared features, products of features, etc.
- Think of a **neural network** as a “stack” of models
    - On the bottom of the stack are the original features
    - Each layer in the stack applies a simple model to the outputs of the previous layer
- Might fit data *too* well (..to be continued)

Geometric interpretation

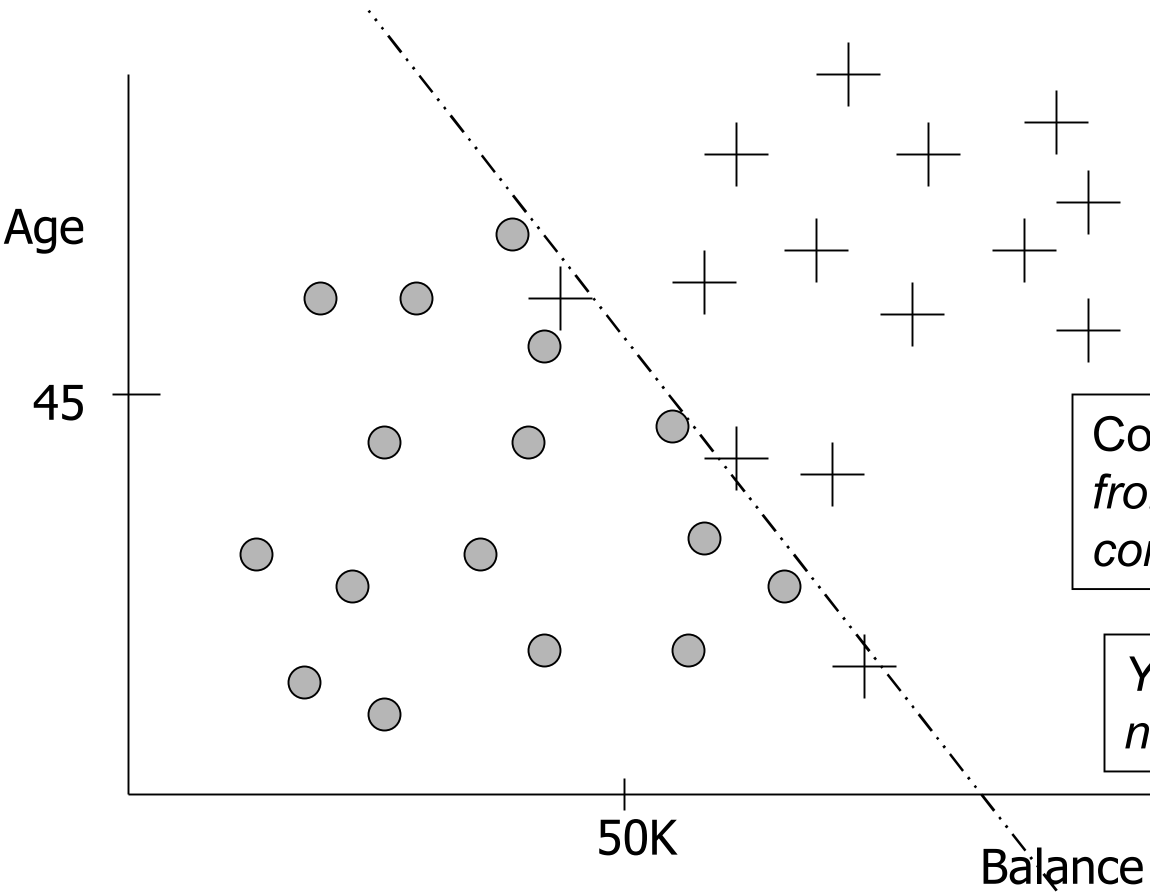
What if we want estimates of class membership probability?



Could we maybe say: *the farther from the line, the more likely the corresponding class?*

Geometric interpretation

What if we want estimates of class membership probability?



Could we maybe say: *the farther from the line, the more likely the corresponding class?*

Yes! – and relationship need not be linear

Logistic Regression

Technically: "log odds" is a linear function of the features:

$$ln[p/(1-p)] = b_0 + b_1x_1 + b_2x_2 + \ldots$$

Ranking Instances and Probability Class Estimation

- In many applications, we don't simply want a yes or no prediction of whether an instance belongs to the class, but we want some notion of **which examples are more or less likely to belong to the class**
    - Which consumers are most likely to respond to this offer?
    - Which customers are most likely to leave when their contracts expire?
- Ranking
    - Tree induction
    - Linear discriminant functions (e.g., linear regressions, logistic regressions, SVMs)
        - Ranking is free
- Class Probability Estimation
    - Tree induction
    - Logistic regression

The many faces of classification:
Classification / Probability Estimation / Ranking

Increasing difficulty

→

Classification *Ranking* *Probability*

- Ranking:
  - – Business context determines the number of actions ("how far down the list")
- Probability:
  - – You can always rank / classify if you have probabilities!

Ranking: Examples

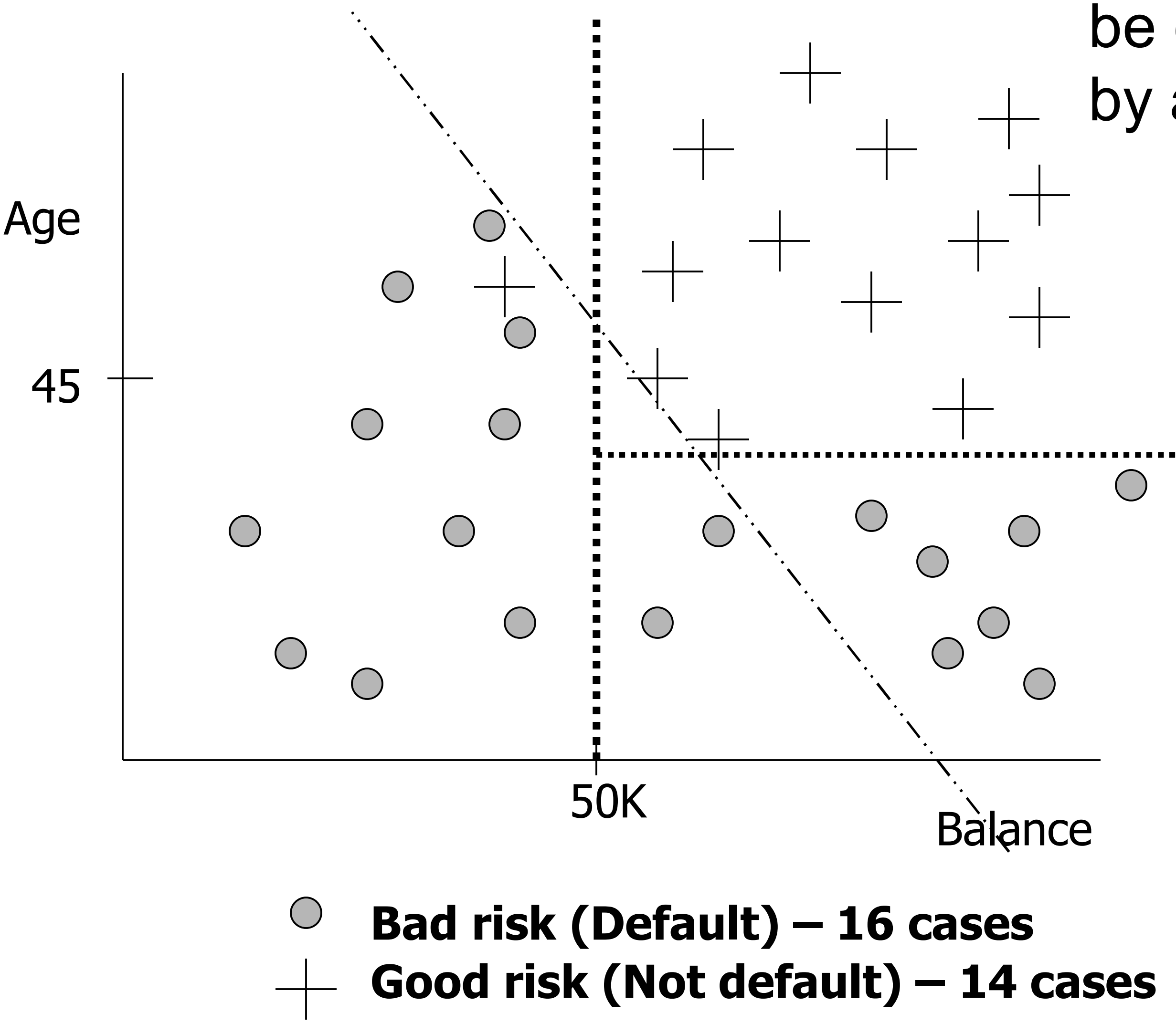
- Search engines
  - Whether a document is relevant to a topic / query

Class Probability Estimation: Examples

- MegaTelCo
  - Ranking vs. Class Probability Estimation
- Identify accounts or transactions as likely to have been defrauded
  - The director of the fraud control operation may want the analysts to focus not simply on the cases most likely to be fraud, but on accounts where the **expected monetary loss** is higher
    - We need to estimate the actual probability of fraud

Geometric interpretation

“True” boundary may not be closely approximated by a linear boundary



Tree induction versus linear model
(e.g., logistic regression)
Factors to consider

- What is more comprehensible to the stakeholders?
  - rules?
  - a numeric function?
- How “smooth” is the underlying phenomenon being modeled? (trees need a lot of data to approx. curved boundaries)
- How “non-linear” is the underlying phenomenon being modeled? (if “very”, much “data engineering” needed to apply linear models)
- How much data do you have?!
  - → there is a key tradeoff between the complexity that can be modeled and the amount of training data available
- What are the characteristics of the data: missing values, types of variables (numeric, categorical), relationships between them, how many are irrelevant, etc.
  - trees fairly robust to these complications

Tree Induction vs Logistic Regression

- For smaller training-set sizes, logistic regression yields better generalization accuracy than tree induction
    - For smaller data, tree induction will tend to over-fit more
- Classification trees are a more flexible model representation than linear logistic regression
- Flexibility of tree induction can be an advantage with larger training sets:
    - Trees can represent substantially nonlinear relationships between the features and the target

Avoiding Over-fitting

Tree Induction:

- Post-pruning
  - takes a fully-grown decision tree and discards unreliable parts
- Pre-pruning
  - stops growing a branch when information becomes unreliable

Linear Models:

- Feature Selection
- Regularization
  - Optimize some combination of fit and simplicity

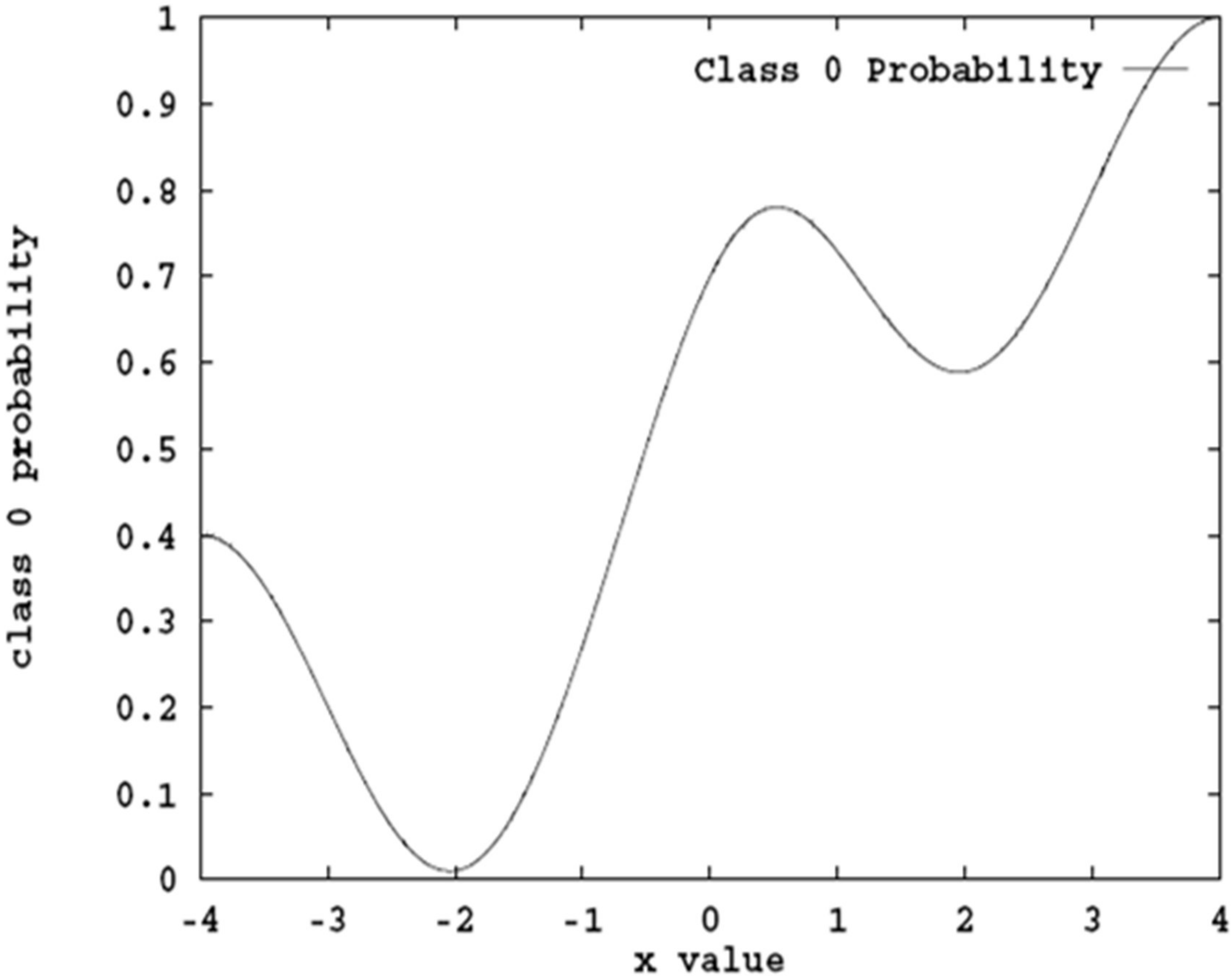
Regularization

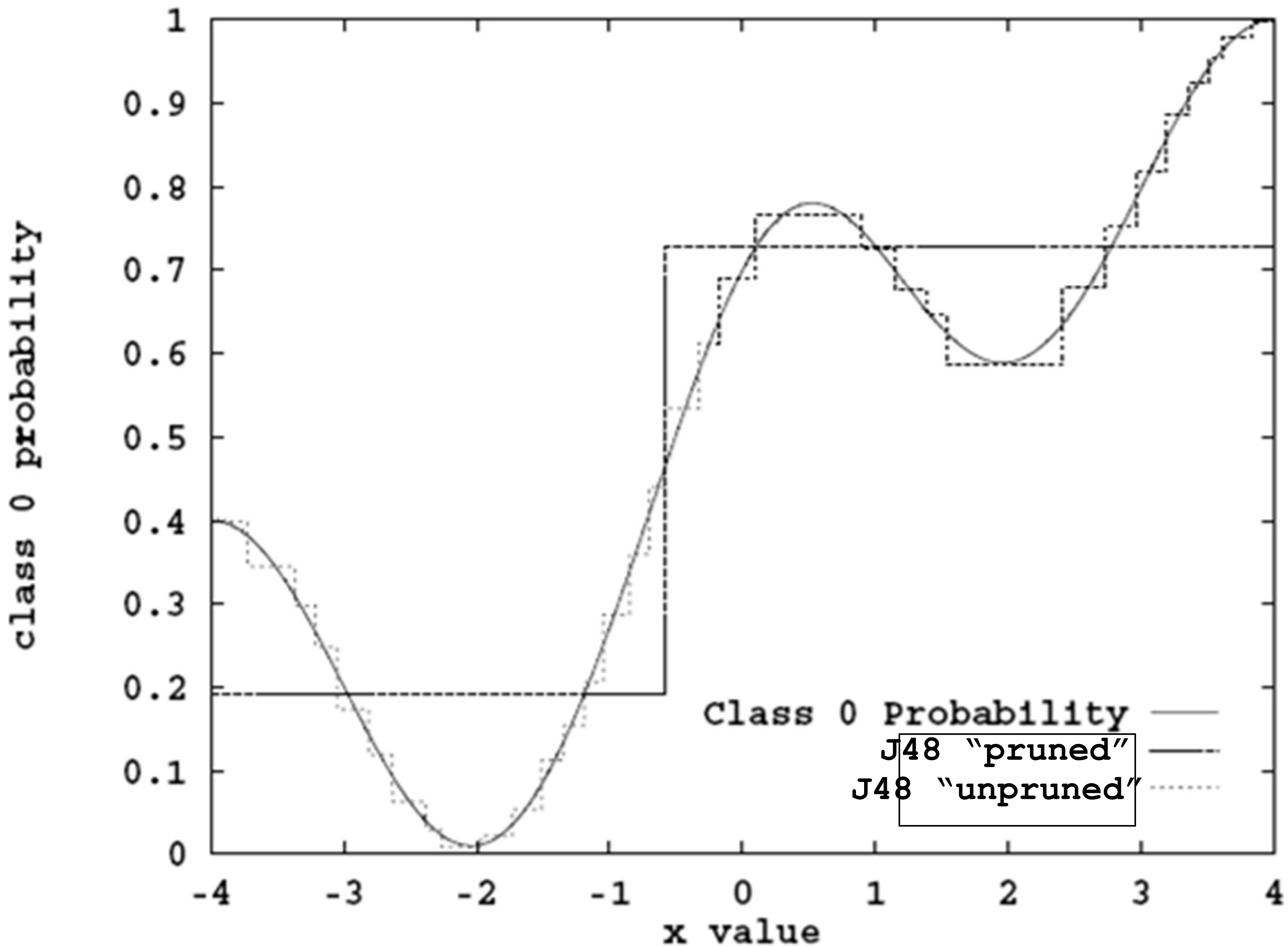
- Regularized linear model:

$$\underset{\boldsymbol{w}}{\operatorname{argmax}}[\operatorname{fit}(\boldsymbol{x}, \boldsymbol{w}) - \lambda * \operatorname{penalty}(\boldsymbol{w})]$$

Regularization : L1 (or L2)-penalty:

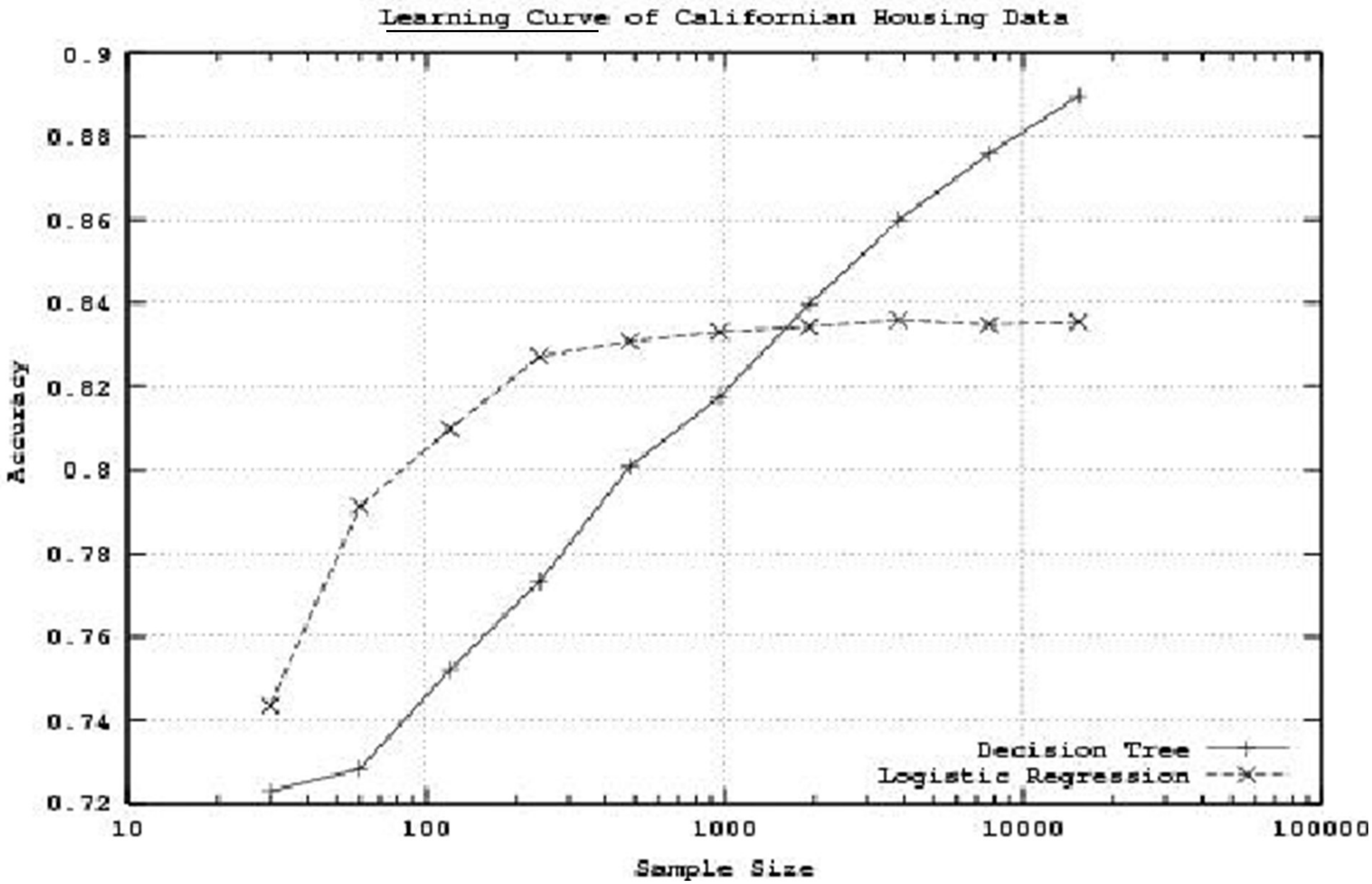
- L2-norm: sum of the *squares* of the weights (w) or coefficients (betas)
  - L2-norm + standard least-squares linear regression = ridge regression
- L1-norm: sum of weights' *absolute values*
  - L1-norm + standard least-squares linear regression = lasso
  - Automatic feature selection


Class 0 Probability
class 0 probability
x value
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
-4
-3
-2
-1
0
1
2
3
4

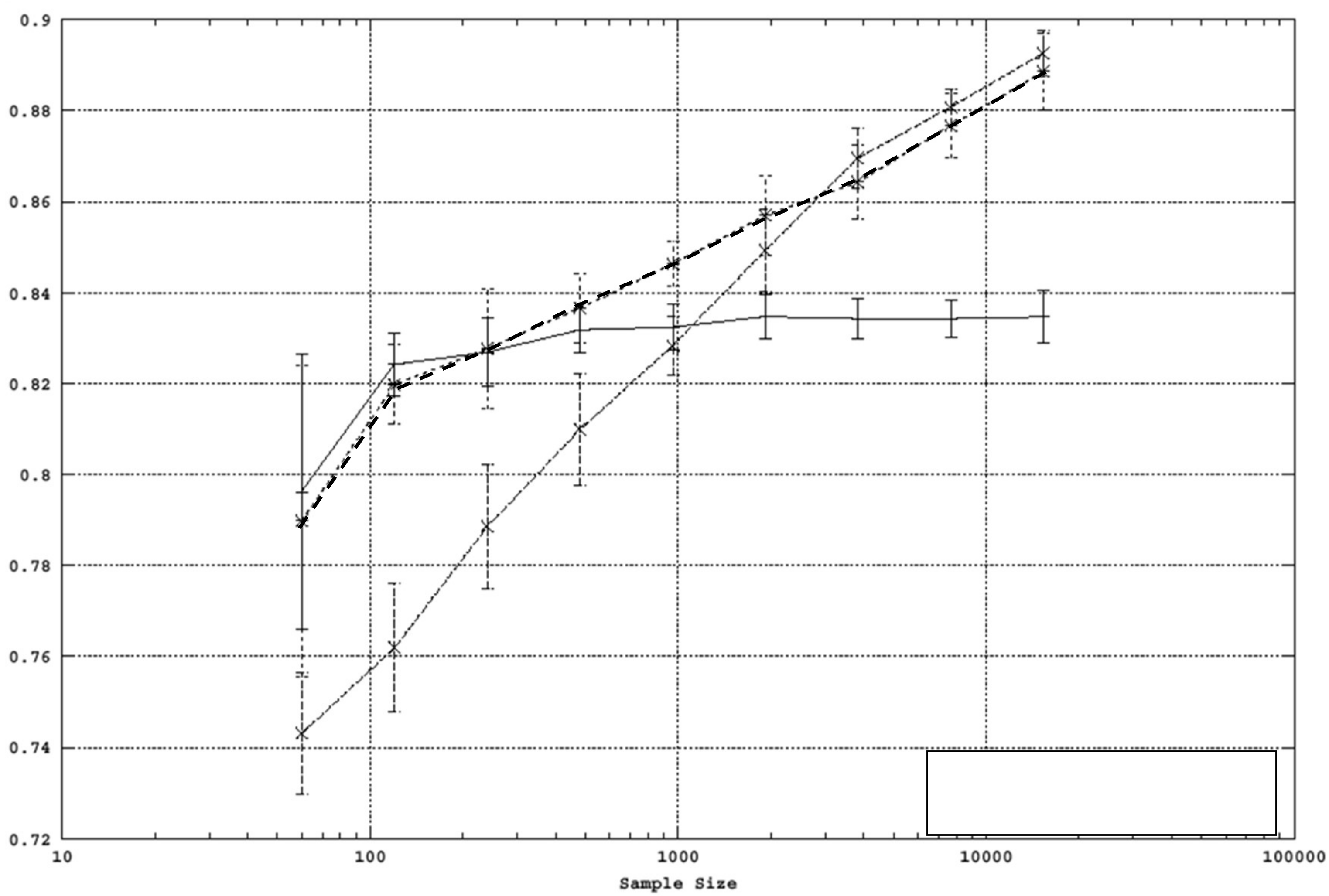


class 0 probability
x value
Class 0 Probability
J48 “pruned”
J48 “unpruned”
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
-4
-3
-2
-1
0
1
2
3
4

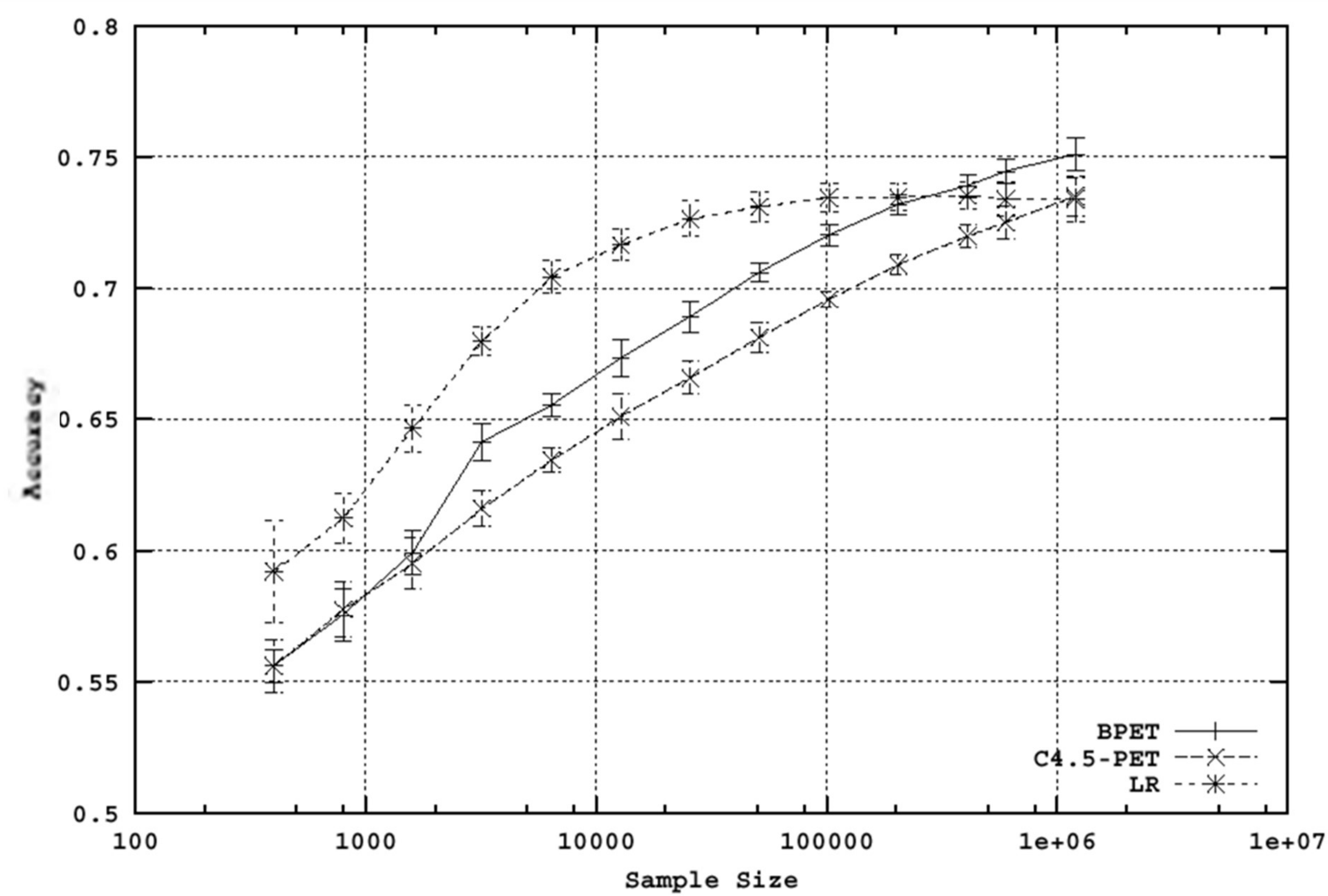
Choice of algorithm is not trivial!



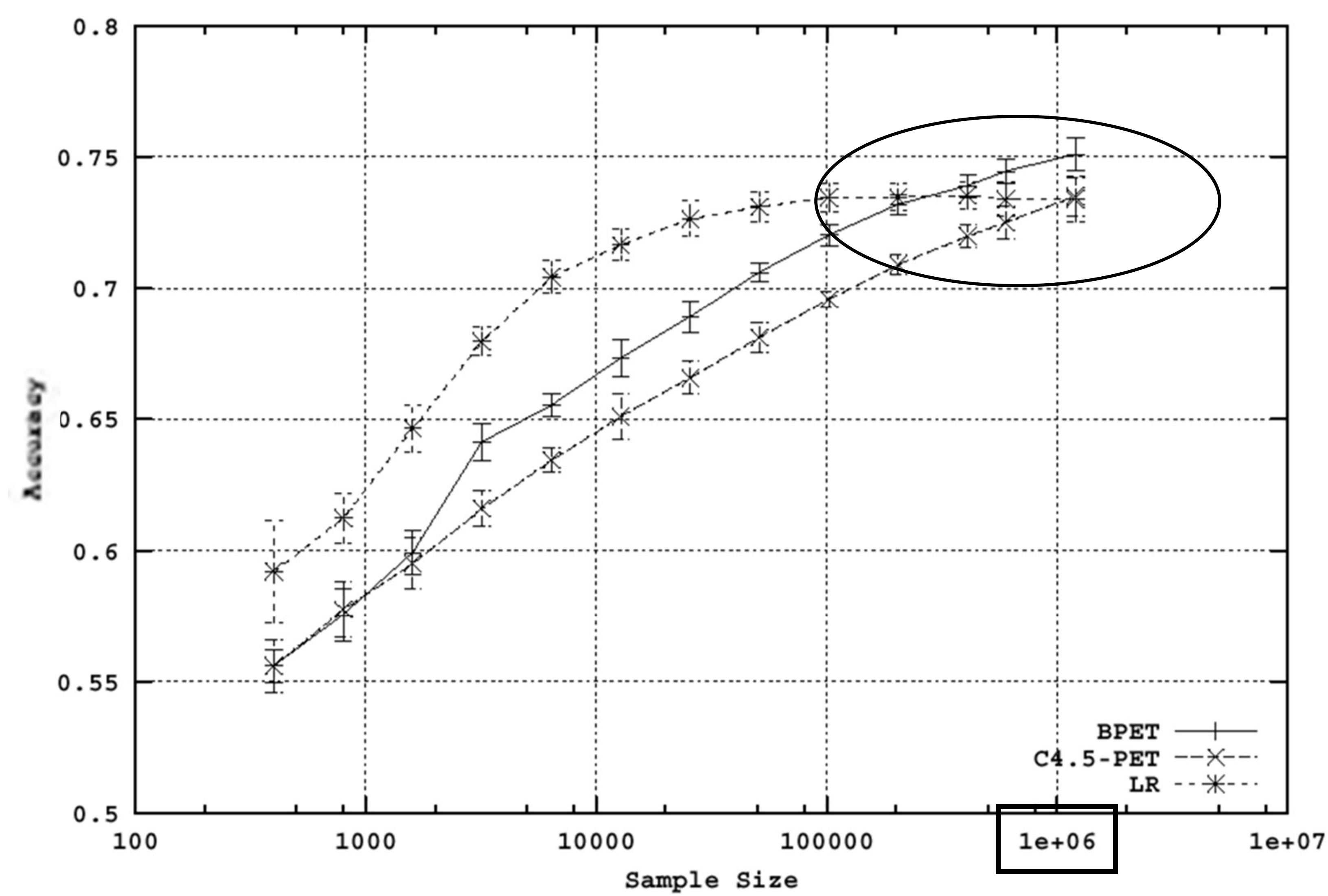
A tiny digression: combine algorithms



Comparing learning curves is essential

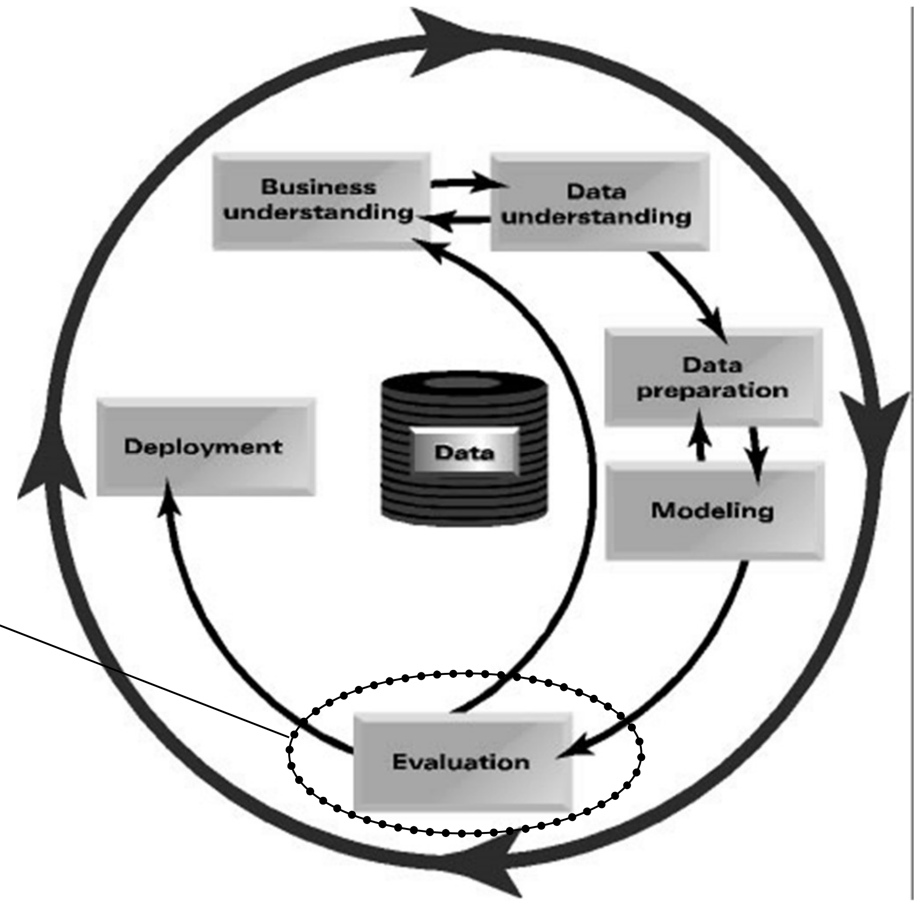


Comparing learning curves is essential



Or, why not try both!

- Integrated data mining packages now allow us to try multiple models…
- … and sort out them out in Evaluation



Summary

- Selecting informative attributes: possibly the most basic data mining technique
- Recursive application builds tree-structured models
  - tree induction is one of the most popular data mining tools
  - very popular for classification
  - also can be used for regression (what would be different?)
- Geometric visualization allows us to understand very different models in a common analytic framework
- Linear models are widely used
  - linear discriminant, support vector machines (SVM), logistic regression, linear regression
- Learning curves are a vital analytical tool when modeling data
  - e.g., can help to judge the potential of investing in more data