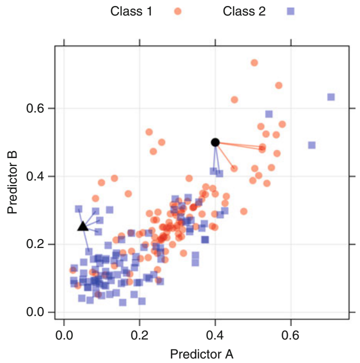
Models for classification

Nonlinear methods

A quick recap

Last lecture, we did...

- **Models for classification**: K-neareast neighbor

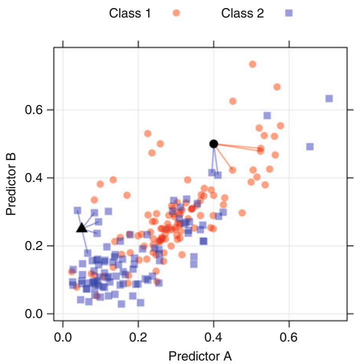


- A new sample is predicted based on the K-closest data points in the training set.
- Two new samples (● and ▲) are being predicted.
- A choice of too few neighbors may over-fit the individual points of the training set while too many neighbors may not be sensitive enough to yield reasonable performance.

A quick recap

Last lecture, we did...

- **Models for classification**: K-neareast neighbor

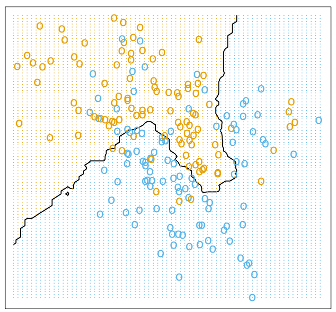


- Nearest neighbor methods use those observations in the training set closest in input space to x to form $\hat{Y}$.
- Closeness implies a metric (for the moment we assume is Euclidean distance).
- We find the K observations with x_i closest to x in input space, and average their responses.

A quick recap

Last lecture, we did...

- **Models for classification**: K-neareast neighbor

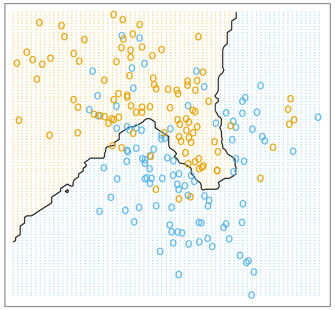


- We use 15-nearest-neighbor averaging of the binary coded response as the method of fitting.
- $\hat{Y}$ is the proportion of orange's in the neighborhood.
- Assigning class orange to the class variable if $\hat{Y} > 0.5$ amounts to a **majority vote in the neighborhood**.

A quick recap

Last lecture, we did...

- **Models for classification**: K-neareast neighbor

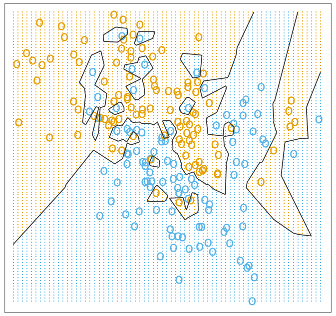


- The colored regions indicate all those points in input space classified as blue or orange by such a rule, in this case found by evaluating the procedure on a fine grid in input space.
- We see that the decision boundaries that separate the blue from the textcolororangeorange regions are far more irregular, and respond to local clusters where one class dominates.

A quick recap

Last lecture, we did...

- **Models for classification**: K-neareast neighbor

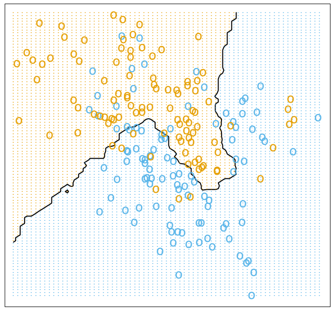


- For 1-nearest-neighbor classification: $\hat{Y}$ is assigned the value y_l of the closest point x_l to x in the training data.
- The regions of classification can be computed relatively easily, and correspond to a **Voronoi tessellation** of the training data.
- Each point x_i has an associated tile bounding the region for which it is the closest input point.

A quick recap

Last lecture, we did...

- **Models for classification**: K-neareast neighbor

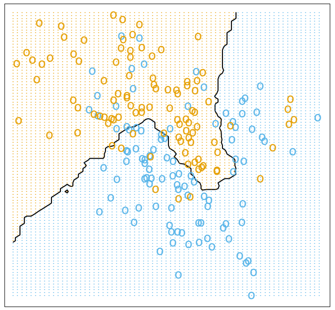


- For K-nearest neighbor fits, the error on the training data should be approximately an increasing function of k.
- It will always be 0 for $K = 1$.
- An independent test set would give us a more satisfactory means for comparing the different methods.

A quick recap

Last lecture, we did...

- **Models for classification**: K-neareast neighbor

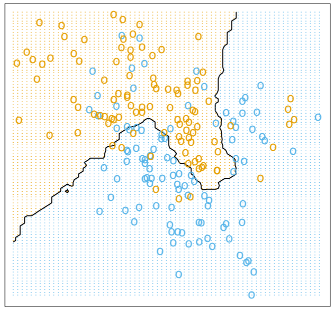


- K-nearest neighbor fits have a single parameter: the number of neighbors k, compared to the p parameters in least-squares fits.
- The effective number of parameters of k-nearest neighbors is N/k and is generally bigger than p, and decreases with increasing k.

A quick recap

Last lecture, we did...

- **Models for classification**: K-neareast neighbor

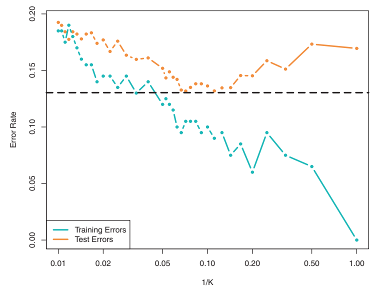


- If the neighborhoods were non overlapping, there would be N/k neighborhoods and we would fit one parameter (a mean) in each neighborhood.
- We cannot use sum-of-squared errors on the training set as a criterion for picking k, since we would always pick $k = 1$.

A quick recap

Last lecture, we did...

- **Models for classification**: K-neareast neighbor



- The training error rate will decline but the test error rate may not.
- As 1/K increases, the method become more flexible.
- The **training error** consistently declines as the flexibility increases.
- The **test error** exhibits a characteristic U-shape: it declines first before increasing again when the method becomes excessively flexible and overfits.

Today's goal

Today, we going to do...

- **Support vector machines**

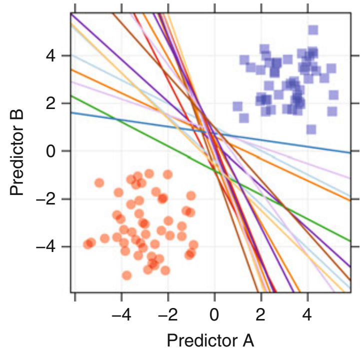
Reading list

- Max Kuhn and Kjell Johnson. *Applied Predictive Modeling*, Springer (2014)
- Gareth James, Daniela Witten, Trevor Hastie and Robert Tibshirani. *An Introduction to Statistical Learning with Applications in R*, Springer (2017)

Support vector machines

Support vector machines are a class of statistical models first developed in the mid-1960s by Vladimir Vapnik.

- In later years, the model has evolved considerably into one of the most flexible and effective machine learning tools available.
- Vapnik in 2010 provides a comprehensive treatment.

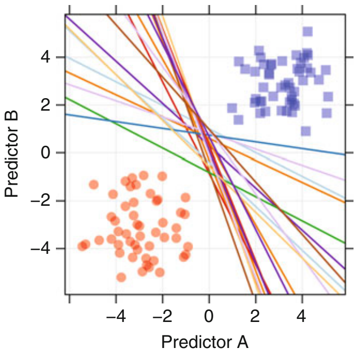


- **Two variables** are used to predict **two classes** of samples that are completely separable.
- There are a multitude (in fact an infinite) number of linear boundaries that perfectly classify these data.
- How would we choose an appropriate class boundary?

Support vector machines

Support vector machines are a class of statistical models first developed in the mid-1960s by Vladimir Vapnik.

- In later years, the model has evolved considerably into one of the most flexible and effective machine learning tools available.
- Vapnik in 2010 provides a comprehensive treatment.

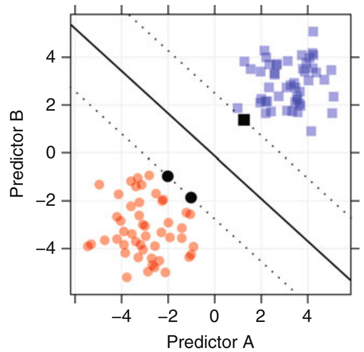


- Many performance measures, such as accuracy, are insufficient since all the curves would be deemed equivalent.
- What would a more appropriate metric be for judging the efficacy of a model?

Support vector machines

Support vector machines are a class of statistical models first developed in the mid-1960s by Vladimir Vapnik.

- In later years, the model has evolved considerably into one of the most flexible and effective machine learning tools available.
- Vapnik in 2010 provides a comprehensive treatment.

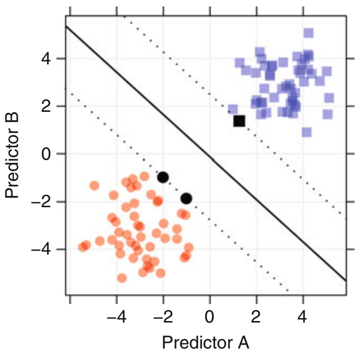


- Vapnik defined an alternate metric called the **margin**.
- The margin is the distance between the classification boundary and the closest training set point.
- The dashed lines on both sides of the boundary are at the **maximum distance** from the line to the closest training set data (equidistant from the boundary line).

Support vector machines

Support vector machines are a class of statistical models first developed in the mid-1960s by Vladimir Vapnik.

- In later years, the model has evolved considerably into one of the most flexible and effective machine learning tools available.
- Vapnik in 2010 provides a comprehensive treatment.



- In this example the three data points are equally closest to the classification boundary and are highlighted with solid black symbols.
- The margin defined by these data points can be quantified and used to evaluate possible models.
- In SVM terminology, the slope and intercept of the boundary that maximize the buffer between the boundary and the data is known as the maximum margin classifier.

Support vector machines

Suppose we have a two-class problem, we code

- Class #1 samples with a value of 1.
- Class #2 samples with a value -1.

The vectors x_i contain the predictor data for a training set sample.

The **maximum margin classifier** creates a decision value $D(\mathbf{x})$ that classifies samples

$\rightsquigarrow$ If $D(x) > 0$ we would predict a sample to be class #1, otherwise class #2.

Support vector machines

For an unknown sample **u**, the decision equation can be written in a similar form as a linear discriminant function that is parameterized in terms of an intercept and slopes as

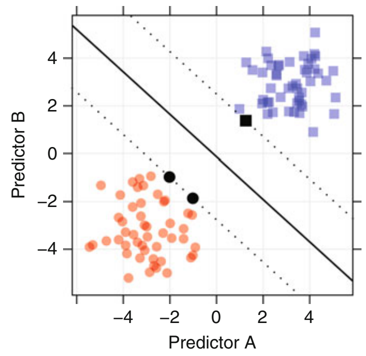
$$D(\mathbf{u}) = \beta_0 + \boldsymbol{\beta}'\boldsymbol{u} = \beta_0 + \sum_{j=1}^{P} \beta_j u_j$$

- Notice that this equation works from the viewpoint of the predictors.
- This equation can be transformed so that the maximum margin classifier can be written in terms of each data point in the sample.

$$D(\mathbf{u}) = \beta_0 + \sum_{j=1}^{P} \beta_j u_j = \sum_{i=1}^{n} y_i \alpha_i \boldsymbol{x}_i' \boldsymbol{u} \text{ with } \alpha_i \geq 0$$

It turns out that, in the completely separable case, the α parameters are exactly zero for all samples that are not on the margin.

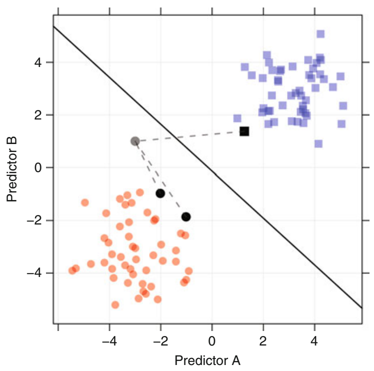
Support vector machines



- The set of nonzero α values are the points that fall on the boundary of the margin.
- The predictor equation is a function of only a subset of the training data points.
- These are referred to as the **support vectors**.
- The prediction function is only a function of the training set samples that are closest to the boundary and are predicted with the least amount of certainty.

Since the prediction equation is supported solely by these data points, the maximum margin classifier is the usually called the **support vector machine**.

Support vector machines



- A new sample, shown as a solid grey circle, is predicted by the model.
- The distances between each of the support vectors and the new sample are as grey dotted lines.
- For these data, there are three support vectors, and therefore contain the only information necessary for classifying the new sample.

Support vector machines

- The core is the summation of the product of: the sign of the class, the model parameter, and the dot product between the new sample and the support vector predictor values.

	True Class	Dot Product	y_i	α_i	Product
SV1	Class 2	-2.4	-1	1.00	2.4
SV2	Class 1	5.4	1	0.34	1.72
SV3	Class 1	1.2	1	0.66	0.79

- The first support vector has the largest single effect on the prediction equation (all other things being equal) and it has a negative slope.
- For our new sample, the dot product is negative,
  - $\rightsquigarrow$ The total contribution of this point is positive and pushes the prediction towards the first class (i.e., a positive value of the decision function $D(u)$).

Support vector machines

- The core is the summation of the product of: the sign of the class, the model parameter, and the dot product between the new sample and the support vector predictor values.

	True Class	Dot Product	y_i	α_i	Product
SV1	Class 2	-2.4	-1	1.00	2.4
SV2	Class 1	5.4	1	0.34	1.72
SV3	Class 1	1.2	1	0.66	0.79

- The remaining two support vectors have positive dot products and an overall product that increases the decision function value for this sample.
- For this model, the intercept is -4.372; $D(u)$ for the new sample is therefore 0.583.
  - ⇝ Since this value is greater than zero, the new sample has the highest association with the first class.

Support vector machines

When the **classes are not completely separable**, we can use extensions to the early maximum margin classifier.

- Cortes and Vapnik[1] put a cost on the sum of the training set points that are on the boundary or on the wrong side of the boundary.
- When determining the estimates of the α values, the margin is penalized when data points are on the wrong side of the class boundary or inside the margin.
- The cost value would be a tuning parameter for the model and is the primary mechanism to control the complexity of the boundary.
- Large penalties, similar to costs, impose limits on the model complexity.
- For SVMs, cost values are used to penalize number of errors; as a consequence, larger cost values induce higher model complexity rather than restrain it.

[1] Cortes C, Vapnik V (1995). "Support-Vector Networks". Machine Learning, 20(3), 273-297.

Support vector machines

The linear nature of the model to nonlinear classification boundaries by substituting the **kernel function** instead of the simple linear cross product:

$$D(\mathbf{u}) = \beta_0 + \sum_{i=1}^{n} y_i \alpha_i \boldsymbol{x}_i' \boldsymbol{u} = \beta_0 + \sum_{i=1}^{n} y_i \alpha_i K(\boldsymbol{x}_i \boldsymbol{u})$$

- $K(\cdot,\cdot)$ is a **kernel function** of the two vectors.
- For the linear case, the kernel function is the same inner product $\boldsymbol{x}_i' \boldsymbol{u}$.
- Other nonlinear transformations can be applied, including:

$$\text{polynomial} = (scale(\boldsymbol{x}'\boldsymbol{u}+1))^{degree}$$
$$\text{radial basis function} = \exp(-\sigma||\boldsymbol{x}-\boldsymbol{u}||^2)$$
$$\text{hyperbolic tangent} = \tanh(scale(\boldsymbol{x}'\boldsymbol{u})+1)$$

- The predictors should be centered and scaled prior to fitting
  - ⇝ Attributes whose values are large in magnitude do not dominate the calculations.

Support vector machines

The linear nature of the model to nonlinear classification boundaries by substituting the **kernel function** instead of the simple linear cross product:

$$D(\mathbf{u}) = \beta_0 + \sum_{i=1}^{n} y_i \alpha_i \boldsymbol{x}_i' \boldsymbol{u} = \beta_0 + \sum_{i=1}^{n} y_i \alpha_i K(\boldsymbol{x}_i \boldsymbol{u})$$

- The **kernel trick** allows the SVM model produce extremely flexible decision boundaries.
- The choice of the kernel function parameters and the cost value control the complexity and should be tuned appropriately so that the model does not over-fit the training data.
- When the cost value is low, the models clearly underfit the data.
- When the cost is relatively high (say a value of 16), the model can over-fit the data, especially if the kernel parameter has a large value.
- Using resampling to find appropriate estimates of these parameters tends to find a reasonable balance between under- and over-fitting.

Support vector machines in R

The `e1071` library contains implementations for a number of statistical learning methods.

- The `svm()` function can be used to fit a support vector classifier when the argument `kernel='linear'` is used.
- A `cost` argument allows us to specify the cost of a violation to the margin.
- When the cost argument is small, then the margins will be wide and many support vectors will be on the margin or will violate the margin.
- When the cost argument is large, then the margins will be narrow and there will be few support vectors on the margin or violating the margin.