Margins:
- Data linearly separable with a margin.
  - Perceptron stops once a separator is found, but we may want to compute the max-margin separator.

Max-margin separators are computed by Support Vector Machines (SVMs).

Linear Classification by a Hyperplane not thru Origin

We can transform this problem to linear classification by a hyperplane through the origin.

Original Problem:
Training data: \((x_i, y_i), i=1, \ldots, n, \ x_i \in \mathbb{R}^d, \ y_i \in \{-1, 1\}\)
Separating hyperplane: \(a^T x + b = 0\)
\((a \in \mathbb{R}^d, \ b = a \text{ scalar})\)

Transform it to:
Training data: \((z_i, y_i), i=1, \ldots, n, \ z_i = [\frac{x_i}{\|x_i\|}] \in \mathbb{R}^d, \ y_i \in \{-1, 1\}\)
Separating hyperplane: \(a^T z + b = 0\)
For all $i = 1, \ldots, n$,

$y_i < z_i, [b] > = y_i < [1, x_i], [b] > = y_i (b + a^T x_i) > 0$

(as $(x_i, y_i)$ are linearly separable)

Thus, if $(x_i, y_i)$'s have a separating hyperplane, then
then $(z_i, y_i)$'s have a separating hyperplane thru origin.

Now suppose $(z_i, y_i)$'s are linearly separable $\&$ through the
origin. Then, $\exists$ a vector $v = [a] \ s.t. \ for \ all \ i,$

$y_i v^T z_i > 0.$

Now: $y_i v^T z_i = y_i < [a], [1, x_i] > = y_i (a + v^T x_i) > 0.$

which means, $(x_i, y_i)$'s are linearly separable.

**Multiclass Classification:**

- Where there are >2 classes.
- For linear multiclass classification, reduce to many binary
  problems.

**One-vs-All (OVA) Reduction:**

- k classes
- Solve k binary classification problems (one per class)
  
  Class 1 vs. Rest, Class 2 vs. Rest, .., Class k vs. Rest

  Not class 1

- On a test example, if you always get a single answer.
  (i.e. you get class i in Class i vs. Rest, get Rest
  in all other cases), predict class i.

  O/w: there are a few options - predict Don't Know or
  tie break at random, or using some other rule.
All-vs-All (AVA) Reduction:

- k classes
- Solve $M = \binom{k}{2} = \frac{k(k-1)}{2}$ binary classification problems
- Class 1 vs. 2, 1 vs. 3, ..., 1 vs. k.
  2 vs. 3, ..., etc.
  Essentially class i vs. j, for $i = 1, ..., k$
  $j = i+1, ..., k$.
- On a test example, pick the most frequently assigned label.
  eg. if $k = 3$, and you get the following results:
  $1$ vs. $2$ : $1$
  $2$ vs. $3$ : $2$
  $1$ vs. $3$ : $1$
- If you get a tie, predict Don't Know.

Confusion Matrix:

$k$ classes, $k \times k$ matrix.

$M_{ij} =$ #examples with label $j$ that are classified as label $i$

$N_j =$ #examples with label $j$

$C_{ij} = M_{ij} / N_j$ (C = confusion matrix)

Measures which classes are easy/hard to separate.

High diagonal entry $\Rightarrow$ class easy to classify
High off-diagonal entry $\Rightarrow$ two classes easily confused.
Comparison:

kNN vs. Decision Trees vs. Linear classifiers.

1. Training time.  \( (kNN < DT, LC) \)
2. Testing time.  \( (kNN \text{ large}, DT, LC \text{ relatively lower}) \)
3. Storage.  \( (kNN \text{ large}, DT, LC \text{ less}) \)
4. Performance on high dimensional data.  \( (LC \text{ best}, DT \text{ sometimes excellent, KNN only when you can find the right features}) \)
5. Flexibility of Decision Boundary.
   \( (kNN \text{ very flexible, DT less so, least flexible is LC}) \)