

— INF4820 —
Algorithms for AI and NLP
Classification

Erik Velldal & Stephan Oepen

Language Technology Group (LTG)

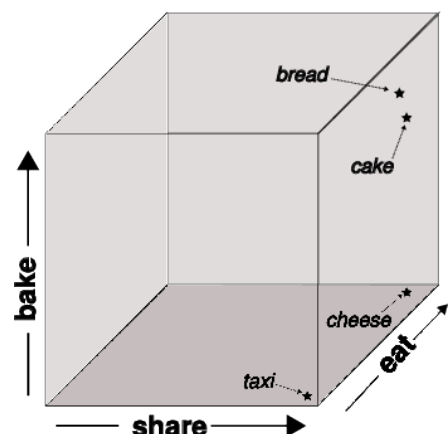
September 16, 2015



Summing up the previous lecture



- ▶ **Semantic spaces:** Vector space models for distributional semantics.
- ▶ Words are represented as points/vectors in a feature space, positioned by their co-occurrence counts for various context features.
- ▶ For each word, extract context features across a corpus.
- ▶ Let each feature type correspond to a dimension in the space.
- ▶ Each word o_i is represented by a (length-normalized) n -dimensional feature vector $\vec{x}_i = \langle x_{i1}, \dots, x_{in} \rangle \in \mathbb{R}^n$.
- ▶ We can now measure, say, the Euclidean distance of words in the space, $d(\vec{x}, \vec{y})$.
- ▶ Semantic relatedness \approx
distributional similarity \approx
spatial proximity





- ▶ So far we've looked at vector space models for detecting *words* with similar *meanings*.
- ▶ It's important to realize that vector space models are widely used for other purposes as well.
- ▶ For example, vector space models are commonly used in **IR** for finding *documents* with similar *content*.
- ▶ Each document d_j is represented by a feature vector, with features corresponding to the terms t_1, \dots, t_n occurring in the documents.
- ▶ Spatial distance \approx similarity of content.
- ▶ Can also represent a search **query** as a vector:
- ▶ The relevance of a documents given by their distance to the query.

3

Today's main topic



- ▶ Machine learning: **Classification**
- ▶ Representing classes and membership
- ▶ Rocchio classifiers
- ▶ k NN classifiers

4



Clustering

- ▶ **Unsupervised** learning from **unlabeled** data.
- ▶ Automatically group similar objects together.
- ▶ No predefined classes or structure, we only specify the similarity measure. Relies on “self-organization”.
- ▶ (The topic for the next lectures.)

Classification

- ▶ **Supervised** learning, requiring **labeled** training data.
- ▶ Train a classifier to automatically assign *new* instances to *predefined* classes, given some set of examples.
- ▶ (Topic for today.)

5

Classification

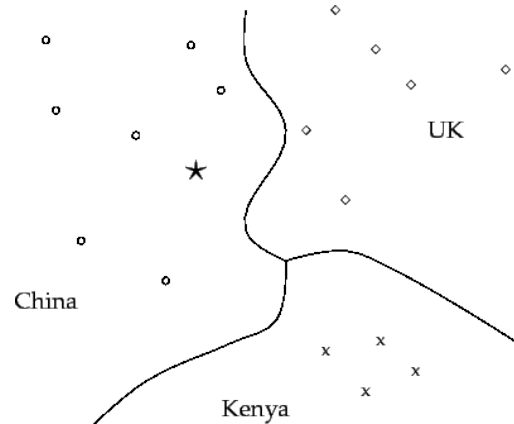


Some examples of classification tasks

- ▶ Named entity recognition
 - ▶ Document (topic) classification
 - ▶ Authorship attribution
 - ▶ Sentiment analysis
 - ▶ Spam filtering
-
- ▶ We'll look at two simple examples of vector space classifiers:
 - ▶ Rocchio
 - ▶ k NN

6

- ▶ A class can simply be thought of as a collection of objects.
- ▶ In our vector space model, objects are represented as *points*, so a class will correspond to a collection of points; a **region**.
- ▶ Vector space classification is based on the **the contiguity hypothesis**:
 - ▶ Objects in the same class form a contiguous region, and regions of different classes do not overlap.
 - ▶ Classification amounts to computing the boundaries in the space that separate the classes; **the decision boundaries**.
 - ▶ How we draw the boundaries is influenced by how we choose to represent the classes.



7

Different ways of representing classes

Exemplar-based

- ▶ No abstraction. Every stored instance of a group can potentially represent the class.
- ▶ Used in so-called *instance based* or *memory based learning* (MBL).
- ▶ In its simplest form; the class = the collection of points.
- ▶ Another variant is to use **medoids**, – representing a class by a single member that is considered central, typically the object with maximum average similarity to other objects in the group.

Centroid-based

- ▶ The average, or the *center of mass* in the region.
- ▶ Given a class c_i , where each object o_j being a member is represented as a feature vector \vec{x}_j , we can compute the class **centroid** $\vec{\mu}_i$ as

$$\vec{\mu}_i = \frac{1}{|c_i|} \sum_{\vec{x}_j \in c_i} \vec{x}_j$$

8



Some more notes on centroids, medoids and typicality

- ▶ Both *centroids* and *medoids* represent a group by a single **prototype**.
- ▶ But while a *medoid* is an actual member of the group, a *centroid* is an *abstract* prototype; an average.
- ▶ **Typicality** can be defined by a member's distance to the prototype.
- ▶ The centroid could also be **distance weighted**:
Let each member's contribution to the average be determined by its average pairwise similarity to the other members of the group.
- ▶ There are parallel discussions on how to represent classes and determine typicality within linguistic and psychological prototype theory.

9

Representing class membership



Hard classes

- ▶ Membership considered a Boolean property: a given object is either part of the class or it is not.
- ▶ A *crisp* membership function.
- ▶ A variant: disjunctive classes. Objects can be members of more than one class, but the memberships are still crisp.

Soft classes

- ▶ Class membership is a graded property.
- ▶ **Distance weighted**.
- ▶ **Probabilistic**: The degree of membership for a given object restricted to $[0, 1]$, and the sum across classes must be 1.
- ▶ **Fuzzy**: The membership function is still restricted to $[0, 1]$, but without the probabilistic constraint on the sum.

10

- ▶ AKA **nearest centroid classifier** or nearest prototype classifier.
- ▶ Uses centroids to represent classes.
- ▶ Each class c_i is represented by its **centroid** $\vec{\mu}_i$, computed as the average of the normalized vectors \vec{x}_j of its members;

$$\vec{\mu}_i = \frac{1}{|c_i|} \sum_{\vec{x}_j \in c_i} \vec{x}_j$$

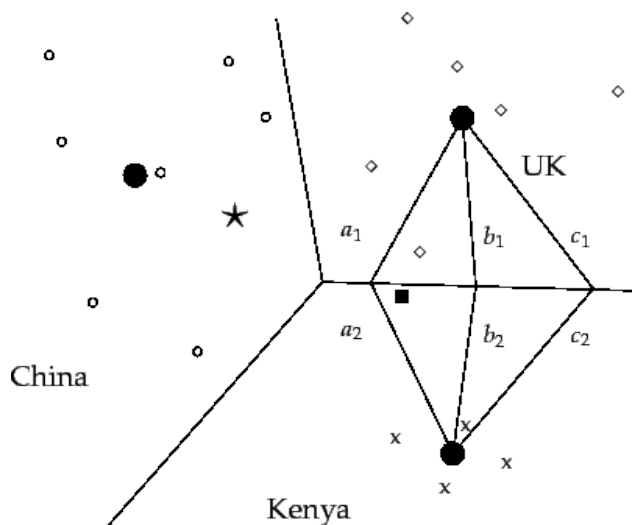
- ▶ To classify a new object o_j (represented by a feature vector \vec{x}_j);
 - determine which centroid $\vec{\mu}_i$ that \vec{x}_j is closest to,
 - and assign it to the corresponding class c_i .
- ▶ The centroids define the boundaries of the class regions.

11

The decision boundary of the Rocchio classifier



- ▶ Defines the boundary between two classes by the set of points equidistant from the centroids.
- ▶ In two dimensions, this set of points corresponds to a **line**.
- ▶ In multiple dimensions: A line in 2D corresponds to a **hyperplane** in a higher-dimensional space.



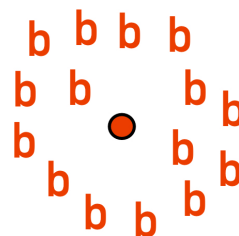
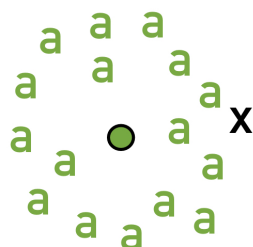
12



- ▶ The classification decision **ignores the distribution of members** locally within a class, only based on the centroid distance.
- ▶ Implicitly assumes that classes are **spheres** with **similar radiuses**.
- ▶ Does not work well for classes that cannot be accurately represented by a single prototype or center (e.g. disconnected or elongated regions).
- ▶ Because the Rocchio classifier defines a **linear decision boundary**, it is only suitable for problems involving *linearly separable* classes.

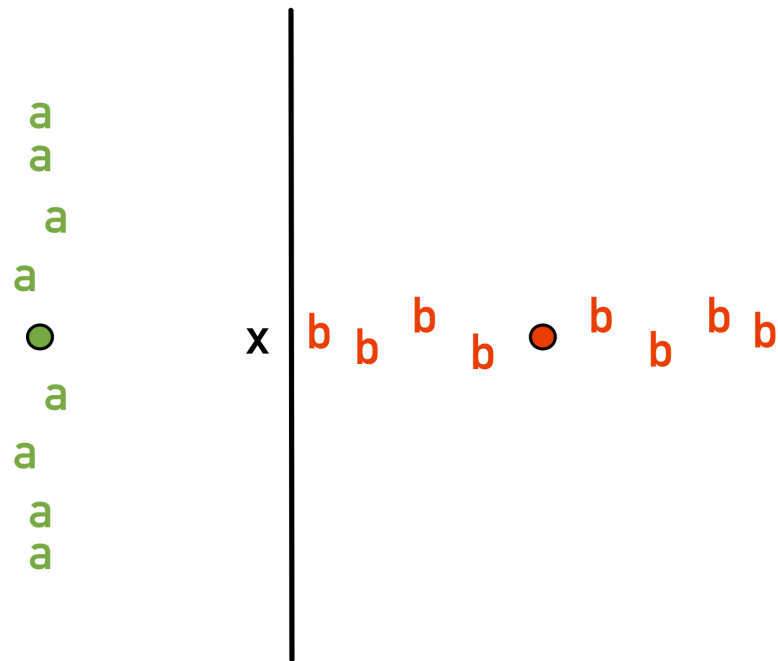
13

Ideal



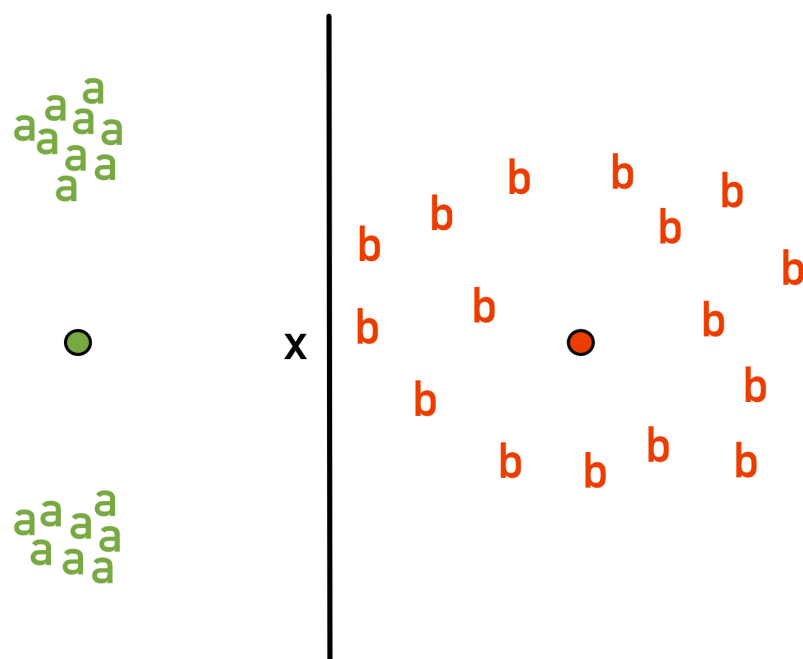
14

Problematic: Elongated regions

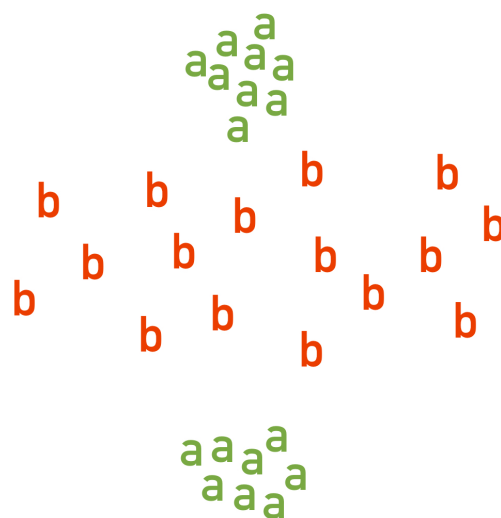
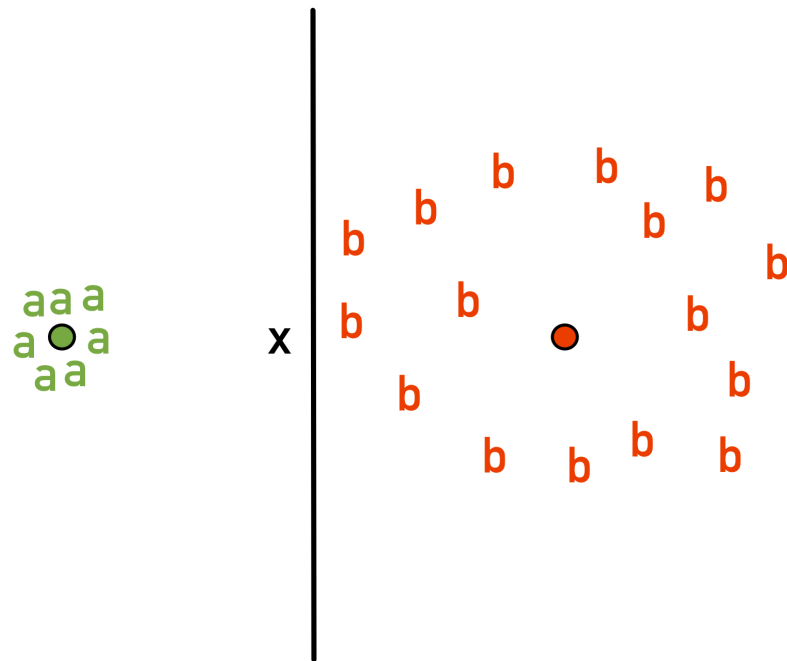


15

Problematic: Non-contiguous regions



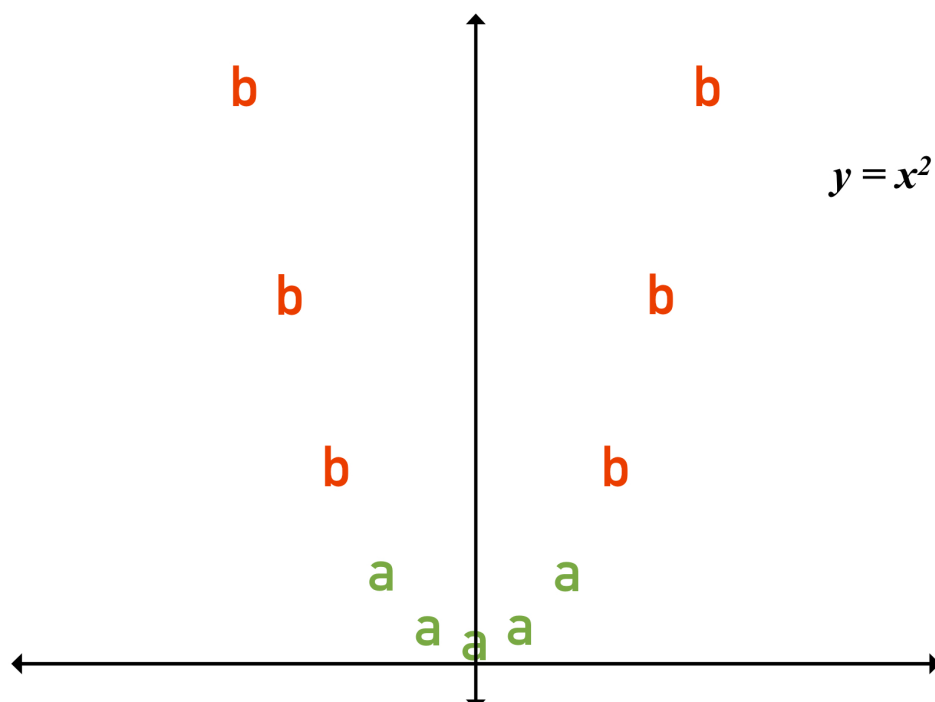
16



← b b b a a a a a b b b →

- Before we turn to talk about non-linear classifiers, note that:
Classes that are not linearly separable in a given feature space...

19



- ... may become linearly separable when the features are mapped to a higher-dimensional space (this is the basis for so-called **kernel** methods).

20

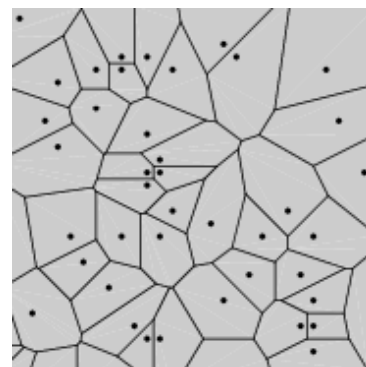
- ▶ k Nearest Neighbor classification.
- ▶ An example of a non-linear classifier.
- ▶ For $k = 1$: Assign each object to the class of its closest neighbor.
- ▶ For $k > 1$: Assign each object to the majority class among its k closest neighbors.
- ▶ Rationale: given the *contiguity hypothesis*, we expect a test object o_i to have the same label as the training objects in the local region of \vec{x}_i .
- ▶ The parameter k must be specified in advance.
- ▶ Unlike Rocchio, the k NN decision boundary is determined locally.
 - ▶ The decision boundary defined by the Voronoi tessellation.

21

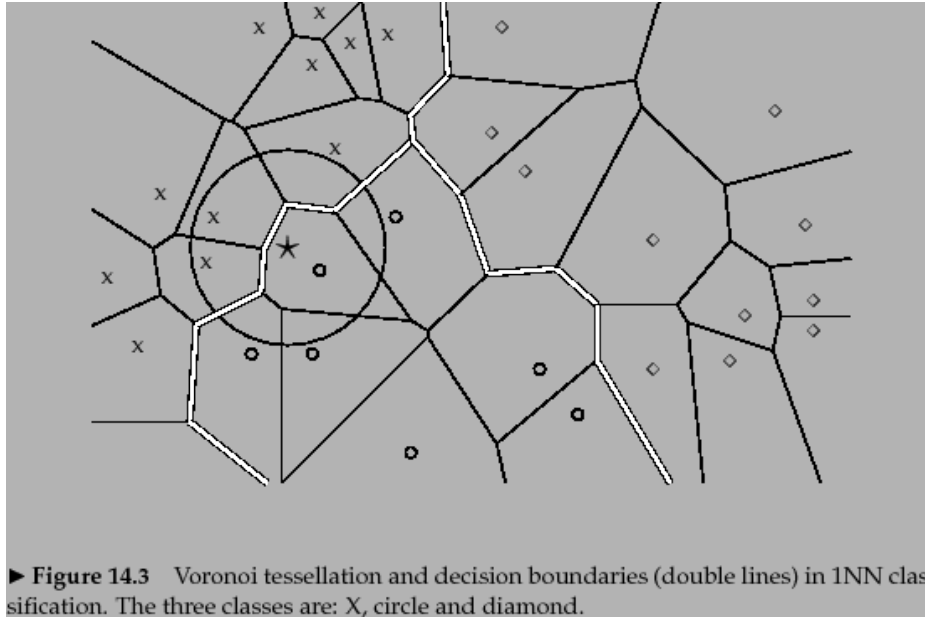
Voronoi tessellation



- ▶ Assuming $k = 1$: For a given set of objects in the space, let each object define a cell consisting of all points that are closer to that object than to other objects.
- ▶ Results in a set of convex polygons; so-called Voronoi cells.
- ▶ Decomposing a space into such cells gives us the so-called Voronoi tessellation.
- ▶ In the general case of $k \geq 1$, the Voronoi cells are given by the regions in the space for which the set of k nearest neighbors is the same.



22



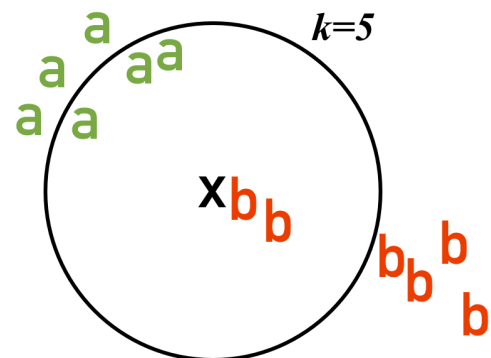
Decision boundary for 1NN: defined along the regions of Voronoi cells for the objects in each class. Shows the **non-linearity** of k NN.

23

“Softened” k NN-classification

A probabilistic version

- The probability of membership in a class c given by the proportion of the k nearest neighbors in c .



Distance weighted votes

- The score for a given class c_i can be computed as

$$\text{score}(c_i, o_j) = \sum_{\vec{x}_n \in \text{knn}(\vec{x}_j)} I(c_i, \vec{x}_n) \text{sim}(\vec{x}_n, \vec{x}_j)$$

where $\text{knn}(\vec{x}_j)$ is the set of k nearest neighbors of \vec{x}_j , **sim** is the similarity measure, and $I(c_i, \vec{x}_n)$ is 1 if $\vec{x}_n \in c_i$ and 0 otherwise.

- Can give more accurate results, and also help resolve ties.

24



- ▶ Not really any *learning* or estimation going on at all;
- ▶ simply **memorizes** all training examples.
- ▶ Generally with in ML; the more training data the better.
- ▶ But for k NN, large training sets come with an efficiency penalty.
- ▶ **Test time** is linear in the size of the training set,
- ▶ but independent of the number of classes.
- ▶ A potential advantage for problems with many classes.
- ▶ Notice the similarity to the problem of ad hoc **retrieval** (e.g., returning relevant documents for a given query);
 - ▶ Both are instances of finding nearest neighbors.

25

Obligatory assignment 2b



- ▶ Builds on oblig **2a**: Vector space representation of a set of words based on BoW features extracted from a sample of the Brown corpus.
- ▶ For **2b** we'll provide class labels for most of the words.
- ▶ Train a Rocchio classifier to predict labels for a set of unlabeled words.

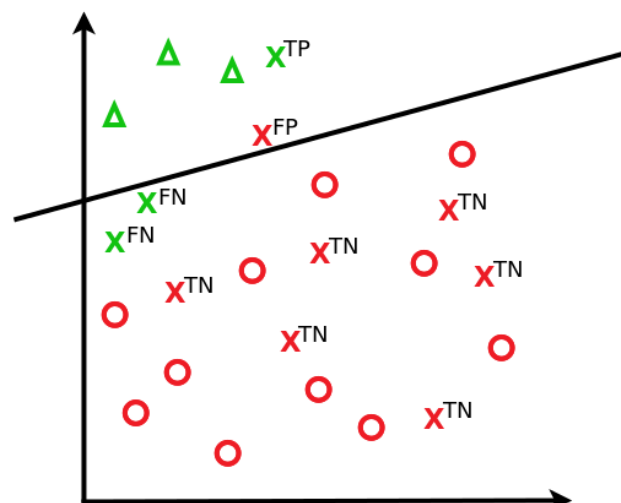
Label	Examples
FOOD	<i>potato, food, bread, fish, eggs ...</i>
INSTITUTION	<i>embassy, institute, college, government, school ...</i>
TITLE	<i>president, professor, dr, governor, doctor ...</i>
PLACE__NAME	<i>italy, dallas, france, america, england ...</i>
PERSON__NAME	<i>lizzie, david, bill, howard, john ...</i>
UNKNOWN	<i>department, egypt, robert, butter, senator ...</i>

26

- ▶ Vector space classification amounts to computing the boundaries in the space that separate the class regions: *the decision boundaries*.
- ▶ To *evaluate* the boundary, we measure the number of correct classification predictions on unseen test items.
 - ▶ Many ways to do this...
- ▶ We want to test how well a model *generalizes* on a *held-out* test set.
- ▶ (Or, if we have little data, by *n*-fold cross-validation.)
- ▶ Labeled test data is sometimes referred to as the *gold standard*.
- ▶ Why can't we test on the training data?

27

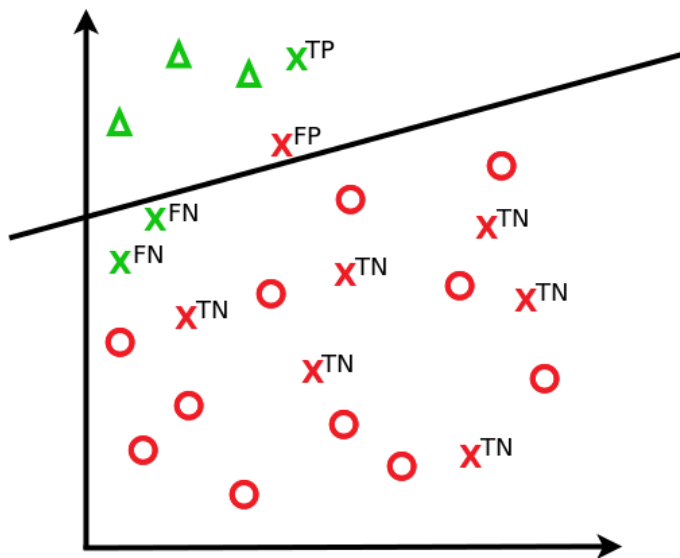
Example: Evaluating classifier decisions



- ▶ Predictions for a given class can be wrong or correct in two ways:

	gold = positive	gold = negative
prediction = positive	true positive (TP)	false positive (FP)
prediction = negative	false negative (FN)	true negative (TN)

28



$$\text{accuracy} = \frac{TP+TN}{N}$$

$$= \frac{1+6}{10} = 0.7$$

$$\text{precision} = \frac{TP}{TP+FP}$$

$$= \frac{1}{1+1} = 0.5$$

$$\text{recall} = \frac{TP}{TP+FN}$$

$$= \frac{1}{1+2} = 0.33$$

$$F\text{-score} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} = 0.4$$

29

Evaluation measures

- ▶ $\text{accuracy} = \frac{TP+TN}{N} = \frac{TP+TN}{TP+TN+FP+FN}$
 - ▶ The ratio of correct predictions.
 - ▶ Not suitable for unbalanced numbers of positive / negative examples.
- ▶ $\text{precision} = \frac{TP}{TP+FP}$
 - ▶ The number of detected class members that were correct.
- ▶ $\text{recall} = \frac{TP}{TP+FN}$
 - ▶ The number of actual class members that were detected.
 - ▶ Trade-off: Positive predictions for all examples would give 100% recall but (typically) terrible precision.
- ▶ $F\text{-score} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$
 - ▶ Balanced measure of precision and recall (harmonic mean).

30



Macro-averaging

- ▶ Sum precision and recall for each class, and then compute global averages of these.
- ▶ The **macro** average will be highly influenced by the **small** classes.

Micro-averaging

- ▶ Sum TPs, FPs, and FNs for all points/objects across all classes, and then compute global precision and recall.
- ▶ The **micro** average will be highly influenced by the **large** classes.

31

Next lecture



- ▶ Unsupervised machine learning for class discovery: **Clustering**
- ▶ Flat vs. hierarchical clustering.
- ▶ C-Means Clustering.
- ▶ Reading: Chapters 16 and 17 in Manning, Raghavan & Schütze (2008) (see course page for the relevant sections).

32