—INF4820—

Vector Space Models
Classification

Erik Velldal

University of Oslo

Nov. 03, 2009

Topics for Today

- Quick recap from last lecture
  - "The contextual theory of meaning" (aka "the distributional hypothesis")
  - Representing words in a vector space model
- Representing *classes* in the vector space
  - Clusters, centroids, medoids...
- Representing class *membership*
  - Boolean, fuzzy, probabilistic...
- Automatically assigning objects to classes
  - Rocchio classification and kNN-classification
- Reading: Section 14-14.4 in Manning, Raghavan & Schütze (2008), *Introduction to Information Retrieval*; `http://informationretrieval.org/`.

Topics We Covered Last Week

Modeling meaning by context

- The basic idea: Capture the meaning of a word in terms of its context.
- Motivation: Can compare the meaning of words by comparing their contexts. No need for prior knowledge.
- Each word o_i represented by a set of feature functions $\{f_1, \ldots, f_n\}$. Each f_j records some property of the observed contexts of o_i.
- Different ways to define context; windows of $\pm n$ words, BoW, grammatical relations.

Topics We Covered Last Week

Modeling meaning by context

- The basic idea: Capture the meaning of a word in terms of its context.
- Motivation: Can compare the meaning of words by comparing their contexts. No need for prior knowledge.
- Each word o_i represented by a set of feature functions $\{f_1, \ldots, f_n\}$. Each f_j records some property of the observed contexts of o_i.
- Different ways to define context; windows of $\pm n$ words, BoW, grammatical relations.

The Contextual Theory of Meaning / The Distributional Hypothesis

- *Meaning is use.* (Wittgenstein, 1953)
- The availability of large amounts of electronic texts, coupled with the computing power of modern machines, lets us implement in practice the classic empiricist claims of Firth, Harris, Wittgenstein, et al.

Topics We Covered Last Week (cont'd)

Vector space models

- Interpret the feature vectors $\vec{f}$ as points positioned in multidimensional space $\Re^n$.
- Each feature defines a dimension in the space.
- Can measure how close different words are in the space by computing e.g. the euclidean distance or the cosine.
- Semantic similarity $\Rightarrow$ Distributional similarity $\Rightarrow$ Spatial proximity

Topics We Covered Last Week (cont'd)

Vector space models

- Interpret the feature vectors $\vec{f}$ as points positioned in multidimensional space $\Re^n$.
- Each feature defines a dimension in the space.
- Can measure how close different words are in the space by computing e.g. the euclidean distance or the cosine.
- Semantic similarity $\Rightarrow$ Distributional similarity $\Rightarrow$ Spatial proximity

Association Weighting

- Raw frequencies not a good indicator of relevance.
- Instead, compute association scores for word–feature pairs based on measures of statistical dependence (e.g. pointwise mutual information, log odds ratio, log likelihood ratio, etc.)

An Aside: Term—Document Vector Models for IE

- So far we've looked at vector space models for detecting *words* with similar *meanings*.
- It's important to realize that similar models (feature vectors + the spatial metaphor) is widely used for other purposes as well.

An Aside: Term—Document Vector Models for IE

- So far we've looked at vector space models for detecting *words* with similar *meanings*.
- It's important to realize that similar models (feature vectors + the spatial metaphor) is widely used for other purposes as well.
- For example, vector space models are commonly used in IR/IE for finding *documents* with similar *content*.
- Each document d_j is represented by a feature vector, with features corresponding to the terms $t_1, \ldots, t_n$ occurring in the documents.
- Computing the spatial distance between the document vectors in the feature space, lets us rank how similar they are in content.

An Aside: Term—Document Vector Models for IE (cont'd)

- Just as the association measures we applied in our context—word semantic space model, a *weighting function* is applied to the raw term frequencies to better bring out the *relevance* of different term features.

An Aside: Term—Document Vector Models for IE (cont'd)

- Just as the association measures we applied in our context—word semantic space model, a *weighting function* is applied to the raw term frequencies to better bring out the *relevance* of different term features.
- The most commonly used weighting function is tf-idf:
  - The term frequency $\text{tf}(t_i, d_j)$ denote the number of times the term t_i occurs in document d_j.
  - The document frequency $\text{df}(t_i)$ denote the total number of documents in the collection that the term occurs in.
  - The inverse document frequency is defined as $\text{idf}(\text{t}_\text{i}) = \log\left(\frac{N}{df(t_i)}\right)$, where N is the total number of documents in the collection.
  - The weight given to term t_i in document d_j is then computed as

$$\text{tf-idf}(t_i, d_j) = \text{tf}(t_i, d_j) \times \text{idf}(t_i)$$

  - A high tf-idf is obtained if a term has a *high* frequency in the given *document* and a *low* frequency in the document *collection* as whole.

An Aside: Term—Document Vector Models for IE (cont'd)

- The term–document vectors can also be used for scoring and ranking a document's relevance relative to a given *search query*.
  - Represent the search query as a vector, just like for the documents.
  - The relevance of documents relative to the query can be ranked according to their distance to the query in the feature space.

The Proximity Matrix

- For a given set of objects $\{o_1, \ldots, o_m\}$ the proximity matrix, R is a square $m \times m$ matrix where element R_{ij} gives the proximity (or distance) between o_i and o_j.
- In our semantic space model, R_{ij} would give the dot-product of the normalized feature vectors $\vec{x}_i$ and $\vec{x}_j$, representing the words o_i and o_j.

The Proximity Matrix

- For a given set of objects $\{o_1, \ldots, o_m\}$ the proximity matrix, R is a square $m \times m$ matrix where element R_{ij} gives the proximity (or distance) between o_i and o_j.
- In our semantic space model, R_{ij} would give the dot-product of the normalized feature vectors $\vec{x}_i$ and $\vec{x}_j$, representing the words o_i and o_j.
- Note that, if our similarity measure is symmetric, i.e. $\text{sim}(\vec{x}, \vec{y}) = \text{sim}(\vec{y}, \vec{x})$, then R will also be symmetric, i.e. $R_{ij} = R_{ji}$
- Metrics like Euclidean distance, cosine similarity, or the dot-product are all symmetric.

The Proximity Matrix

- For a given set of objects $\{o_1, \ldots, o_m\}$ the proximity matrix, R is a square $m \times m$ matrix where element R_{ij} gives the proximity (or distance) between o_i and o_j.
- In our semantic space model, R_{ij} would give the dot-product of the normalized feature vectors $\vec{x}_i$ and $\vec{x}_j$, representing the words o_i and o_j.
- Note that, if our similarity measure is symmetric, i.e. $\mathrm{sim}(\vec{x}, \vec{y}) = \mathrm{sim}(\vec{y}, \vec{x})$, then R will also be symmetric, i.e. $R_{ij} = R_{ji}$
- Metrics like Euclidean distance, cosine similarity, or the dot-product are all symmetric.
- Computing all the pairwise similarities *once* and then storing them in R can help save time in many applications.
- E.g., by sorting the row elements of R, we get access to an important type of similarity relation; nearest neighbors. . .

Nearest Neighbors

kNN

- The k points closest in the space to a given target point.
- Such kNN sets can be used as an approximation to *classes*.
- Dagan, Lee, & Pereira, 1999 suggest the use of *nearest neighbors averaging* , e.g. when smoothing the probabilities in a language-model. Like class-based methods, but each word defines its own class.

Nearest Neighbors

kNN

- The k points closest in the space to a given target point.
- Such kNN sets can be used as an approximation to *classes*.
- Dagan et al., 1999 suggest the use of *nearest neighbors averaging* , e.g. when smoothing the probabilities in a language-model. Like class-based methods, but each word defines its own class.

RNN

- Reciprocal/Respective Nearest Neighbors
- Can be used to identify words that are substitutable or near-synonyms, (Hindle, 1990; Lin, 1998),
- Query expansion for IR, etc.

The 10 nearest neighbors of the noun ***norge*** (**Norway**):

Rank	Neighbor	Similarity
1	*danmark* (Denmark)	0.579
2	*sverige* (Sweden)	0.567
3	*tyskland* (Germany)	0.562
4	*russland* (Russia)	0.550
5	*kina* (China)	0.533
6	*bergen* (Bergen)	0.512
7	*frankrike* (France)	0.511
8	*land* (land, country)	0.508
9	*england* (England)	0.499
10	*finland* (Finland)	0.498

(kNN relations computed for our semantic space example; log-odds weighted noun vectors with features based on grammatical relations extracted from the Oslo Corpus.)

The 10 nearest neighbors of the target noun ***konflikt*** (**conflict**):

Rank	Neighbor	Similarity
1	*problem* (problem)	0.537
2	*krise* (crisis)	0.481
3	*strid* (fight, discord, controversy)	0.470
4	*uenighet* (disagreement)	0.453
5	*motsetning* (contrast, difference, opposition)	0.438
6	*vanskelighet* (difficulty, trouble)	0.432
7	*kris* (?)	0.426
8	*misforståelse* (misunderstanding)	0.425
9	*brudd* (break, rupture)	0.420
10	*krangel* (quarrel)	0.412

(kNN relations computed for our semantic space example; log-odds weighted noun vectors with features based on grammatical relations extracted from the Oslo Corpus.)

Problems with our Nearest Neighbor Lists

- The semantic coherency of the lists are reduced by polysemous targets and polysemous neighbors.
- The retrieved words may be similar to *distinct senses* of the target.
- Given the target word, we tend to automatically select the appropriate reading of the neighboring words, —can make the kNN sets seem more semantically coherent than they really are.
- Resnik, 1993 on sense conflation:

  > *"If each token is associated with a single point in semantic space, then words having multiple senses will occupy a point determined by the relative frequencies of the individual senses."*

- Not necessarily a problem is our goal is restricted to capturing distributional similarity as such. . .

The 10 nearest neighbors of the target noun ***mønster*** (**pattern**):

Rank	Neighbor	Similarity
1	*norm* (norm)	0.350
2	*tradisjon* (tradition)	0.321
3	*variant* (variety)	0.320
4	*form* (form)	0.319
5	*stil* (style)	0.317
6	*struktur* (structure)	0.317
7	*ramme* (frame)	0.316
8	*trekk* (feature, property)	0.314
9	*fellesskap* (community)	0.311
10	*ram* (ram ?)	0.311

(kNN relations computed for our semantic space example; log-odds weighted noun vectors with features based on grammatical relations extracted from the Oslo Corpus.)

Random Examples of RNNs in the Semantic Space:

Word 1	Word 2	Similarity
år (year)	*måned* (month)	0.679
spørsmål (question)	*problemstilling* (problem)	0.587
kamp (fight, game)	*turnering* (tournament)	0.498
pasient (patient)	*klient* (client)	0.579
slutt (end)	*begynnelse* (beginning)	0.737
ressurs (resource)	*kapasitet* (capacity)	0.511
økning (increase)	*reduksjon* (decrease)	0.712
beløp (amount, sum)	*sum* (sum)	0.550
besøk (visit)	*opphold* (stay)	0.438
elv (river)	*bekk* (river, stream)	0.417
olje (oil)	*gass* (gass)	0.409
arbeidsmarked (labor marked)	*arbeidsliv* (employment sector)	0.466
jørn (Jørn)	*ingrid* (Ingrid)	0.706
koffert (suitcase)	*veske* (bag, purse)	0.536
skepsis (skepticism, disbelief)	*misnøye* (discontent, dissatisfaction)	0.365

Classes and Classification

- A class can be thought of as a collection of objects. Such as;
  - Documents sharing the same category of content, e.g. $\{d_i, \ldots, d_k\} \subset$ SPORTS, $\{d_l, \ldots, d_m\} \subset$ ENTERTAINMENT
  - Words expressing the same ontological concept, e.g. {*salmon*, *trout*, *tuna*, *barracuda*} $\subset$ FISH

Classes and Classification

- A class can be thought of as a collection of objects. Such as;
  - Documents sharing the same category of content, e.g. $\{d_i, \ldots, d_k\} \subset$ SPORTS, $\{d_l, \ldots, d_m\} \subset$ ENTERTAINMENT
  - Words expressing the same ontological concept, e.g. {*salmon*, *trout*, *tuna*, *barracuda*} $\subset$ FISH
- In our vector space model, objects are represented as *points*, so a class will correspond to a collection of points; a region.
- When a collection of points correspond to a densely populated region in the space, we refer to the collection as a cluster.

Classes and Classification

- A class can be thought of as a collection of objects. Such as;
  - Documents sharing the same category of content, e.g. $\{d_i, \ldots, d_k\} \subset$ SPORTS, $\{d_l, \ldots, d_m\} \subset$ ENTERTAINMENT
  - Words expressing the same ontological concept, e.g. {*salmon*, *trout*, *tuna*, *barracuda*} $\subset$ FISH
- In our vector space model, objects are represented as *points*, so a class will correspond to a collection of points; a region.
- When a collection of points correspond to a densely populated region in the space, we refer to the collection as a cluster.
- Classification: The task of automatically assigning class membership to a given object. A core task in supervised machine learning.

Classes and Classification

- A class can be thought of as a collection of objects. Such as;
  - Documents sharing the same category of content, e.g. $\{d_i, \ldots, d_k\} \subset$ SPORTS, $\{d_l, \ldots, d_m\} \subset$ ENTERTAINMENT
  - Words expressing the same ontological concept, e.g. {*salmon*, *trout*, *tuna*, *barracuda*} $\subset$ FISH
- In our vector space model, objects are represented as *points*, so a class will correspond to a collection of points; a region.
- When a collection of points correspond to a densely populated region in the space, we refer to the collection as a cluster.
- Classification: The task of automatically assigning class membership to a given object. A core task in supervised machine learning.
- In a *vector space model*, 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.

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).

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).
- One 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.

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).
- One 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 *center of mass* or *center of gravity* 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

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

Different Ways of Representing Classes (cont'd)

Some more notes on centroids, medoids and typicality

- *Centroids* similar to *medoids* in that we represent a group of objects by a single point, a "prototype".
- But while a *medoid* is an actual member of the group (the "most typical" member), a *centroid* is an "abstract prototype"; an average.

Different Ways of Representing Classes (cont'd)

Some more notes on centroids, medoids and typicality

- *Centroids* similar to *medoids* in that we represent a group of objects by a single point, a "prototype".
- But while a *medoid* is an actual member of the group (the "most typical" member), a *centroid* is an "abstract prototype"; an average.
- The typicality of class members can be determined by their distance to the prototype.

Different Ways of Representing Classes (cont'd)

Some more notes on centroids, medoids and typicality

- *Centroids* similar to *medoids* in that we represent a group of objects by a single point, a "prototype".
- But while a *medoid* is an actual member of the group (the "most typical" member), a *centroid* is an "abstract prototype"; an average.
- The typicality of class members can be determined by their 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.

Different Ways of Representing Classes (cont'd)

Some more notes on centroids, medoids and typicality

- *Centroids* similar to *medoids* in that we represent a group of objects by a single point, a "prototype".
- But while a *medoid* is an actual member of the group (the "most typical" member), a *centroid* is an "abstract prototype"; an average.
- The typicality of class members can be determined by their 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.
- The discussion of how to represent classes in machine learning parallels the discussion of how to represent classes and determine typicality within linguistic and psychological prototype theory.

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.
- Probabilistic. The degree of membership for a given 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.

Classification

- As said, vector space classification relies on the assumption that objects in the same class form a contiguous region.
- Classification amounts to computing the boundaries in the space that separate the classes: *The decision boundaries*.
- What counts as a good or accurate boundary?
  - A boundary that leads to high classification accuracy on unseen test items.
- How we end up drawing the the boundaries is influenced by how we choose to represent the classes.

Rocchio Classification

- Uses centroids to represent classes and define the boundaries of the class regions.
- Each class c_i represented by its centroid, 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 classification rule in Rocchio is to classify a point in accordance with the region it falls into.

Next Week

- More on Rocchio classifiers and kNN classifiers
- Linear vs. non-linear classifiers
- Voronoi Tessellations
- Unsupervised machine learning for class discovery: Clustering
- Flat vs. hierarchical clustering.
- C-Means Clustering.
- Reading: Chapter 16 in Manning, Raghavan & Schütze (2008), *Introduction to Information Retrieval*; `http://informationretrieval.org/`.

Dagan, I., Lee, L., & Pereira, F. (1999). Similarity-based models of word cooccurrence probabilities. *Machine Learning*, *34*(1-3), 43 – 69.

Firth, J. R. (1968). A synopsis of linguistic theory. In F. R. Palmer (Ed.), *Selected papers of j. r. firth: 1952–1959.* Longman.

Grefenstette, G. (1992). SEXTANT: Exploring unexplored contexts for semantic extraction from syntactic analysis. In *Proceedings of the 30th Meeting of the Association for Computational Linguistics* (pp. 324 – 326). Newark, Delaware.

Harris, Z. S. (1968). *Mathematical structures of language.* New York: Wiley.

Hindle, D. (1990). Noun classification from predicate-argument structures. In *Acl:90* (pp. 268 – 275). Pittsburgh, USA.

Lin, D. (1998). Automatic retrieval and clustering of similar words. In *Proceedings of the 17th International Conference on Computational Linguistics and the 36th Annual Meeting of the Association for Computational Linguistics* (pp. 768 – 774). Montreal, Canada.

Resnik, P. (1993). *Selection and information: A class-based approach to lexical relationships.* Unpublished doctoral dissertation, Department of Computer and Information Science, University of Pennsylvania.

Wittgenstein, L. (1953). *Philosophical investigations.* Oxford: Blackwell.