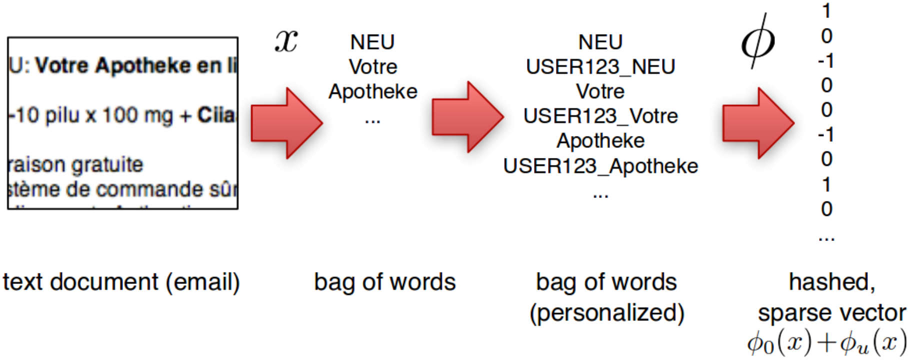
Feature Hashing for Large Scale Multitask Learning

Weinberger et al. 2009

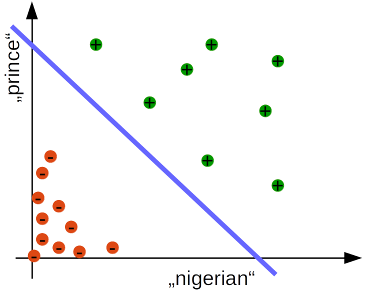
Seminar aus Data Mining und Maschinellem Lernen



Introduction

Task: classify email into spam / not spam

- using *linear classifier*
  - classification can be expressed as a dot product of a *feature vector* **x** and a *weight vector* **w**
    $c = \langle \mathbf{x}, \mathbf{w} \rangle$



Introduction

TECHNISCHE UNIVERSITÄT DARMSTADT

How to turn a text document into a vector?

naive approach: dictionary

- $D : words \rightarrow \mathbb{N}$
- every index in a feature vector corresponds to a word
  - one-hot encoding
- normalized sum over all words in a document

Introduction

TECHNISCHE UNIVERSITÄT DARMSTADT

Problem: high dimensionality (~40 million)

Size of the feature vector:

- number of words
- in every language
- + every misspelling

$\Rightarrow$ Solution: dimensionality reduction

- Only consider frequent words?
- misspellings and unknown brand names are probably good indicators for spam

Feature Hashing

TECHNISCHE UNIVERSITÄT DARMSTADT

Reduces the dimensionality by hashing into a space of size m

Advantages:

- no (original) features are ignored
- no dictionary
- outputs fixed-size feature vector

- hash function $h : \mathbb{N} \rightarrow \{1, ..., m\}$
- hash function $\xi : \mathbb{N} \rightarrow \{\pm 1\}$
- $\phi_i^{h,\xi}(\mathbf{x}) = \sum_{j:h(j)=i} \xi(j) x_j$

Feature Hashing

TECHNISCHE UNIVERSITÄT DARMSTADT

- hash function $h : \mathbb{N} \to \{1, ..., m\}$
- hash function $\xi : \mathbb{N} \to \{\pm 1\}$
- $\phi_i^{h,\xi}(\mathbf{x}) = \sum_{j:h(j)=i} \xi(j) x_j$

Pseudocode (assuming **x** is a normalized sum of one-hot encoded vectors):

```
function hashing_vectorizer(features : array of string, N : integer):
    x := new vector[N]
    for f in features:
        h := hash(f)
        idx := h mod N
        if ξ(f) == 1:
            x[idx] += 1
        else:
            x[idx] -= 1
    return x
```

source: `https://en.wikipedia.org/wiki/Feature_hashing`

Feature Hashing

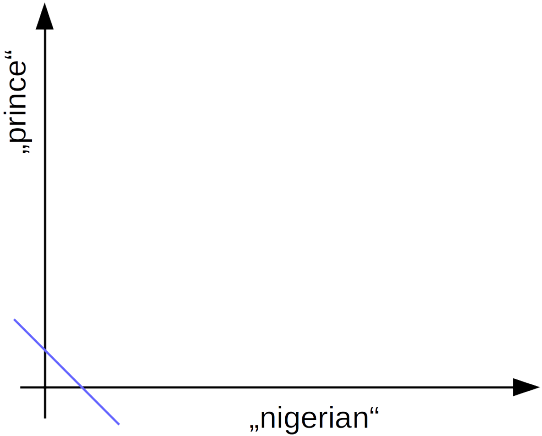
TECHNISCHE UNIVERSITÄT DARMSTADT

Properties of this formulation:

- unbiased
  $\mathbf{E}_\phi[\langle\phi(\mathbf{x}), \phi(\mathbf{x}')\rangle] = \langle\mathbf{x}, \mathbf{x}'\rangle$
- variance shrinks with dimensionality m of the hash-space
  variance rises with big values in $\mathbf{x}$ or $\mathbf{x}'$
  $\mathbf{Var}_\phi[\langle\phi(\mathbf{x}), \phi(\mathbf{x}')\rangle] = \frac{1}{m}(\sum_{i \neq j} x_i^2 {x'}_j^2 + x_i x'_i x_j x'_j)$
- length-preserving with high probability
- the larger a value in $\mathbf{x}$ or $\mathbf{x}'$, the larger the distortion from collisions
- hashing multiple times
  - decreases large values in $\mathbf{x}$ or $\mathbf{x}'$
  - decreases sparsity
  - increases variance

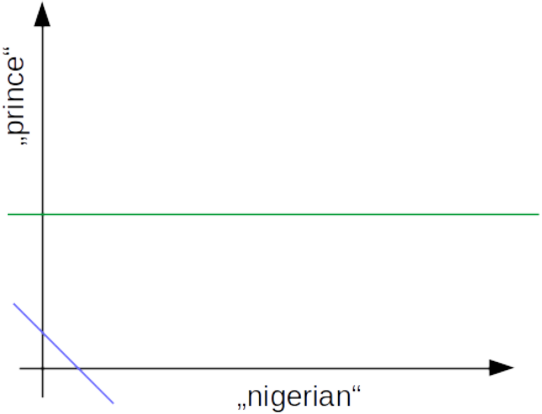
Personalization

Spam is subjective.



Personalization

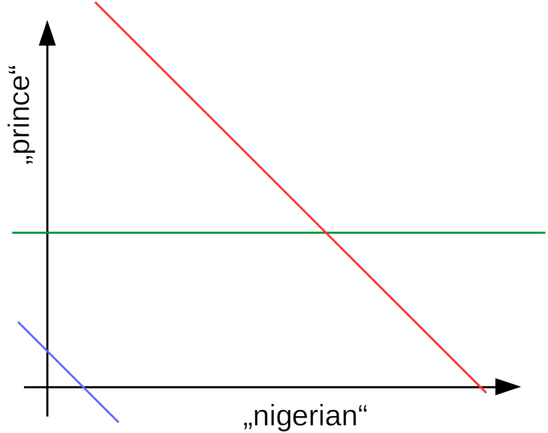
Spam is subjective.



Personalization

TECHNISCHE UNIVERSITÄT DARMSTADT

Spam is subjective.



Multi-task Learning

Goal: Learning a number of related tasks (individual spam-filter per user).

Challenges:

- little training data per user (task)
- *no* training data for most users

- learn shared parts of all tasks
- learn a specialized classifier for each task

$\Rightarrow$ inductive transfer between tasks

Example:

- **everybody** can agree that “Viagra” indicates spam (shared)
- **some** people regard “nigerian” + “prince” as spam (task-specific)

Multi-task Learning

- hash functions $\phi_0, ..., \phi_{|U|}$
- for each user that provided labels:
  - one local hash function ϕ_u
  - one local predictor $\mathbf{w_u}$
- one global hash function ϕ_0
- one local predictor $\mathbf{w_0}$ shared between all users

All hashed into a common space:

$$\mathbf{w_h} = \phi_0(\mathbf{w_0}) + \sum_{u \in U} \phi_u(\mathbf{w_u})$$

$\Rightarrow$ assumption: low interference between user hashes

Multi-task Learning

prediction for user u: $\left\langle \phi_0(\mathbf{x}) + \phi_u(\mathbf{x}), \mathbf{w_h} \right\rangle = \left\langle \mathbf{x}, \mathbf{w_0} + \mathbf{w_u} \right\rangle + \epsilon_i + \epsilon_d$

interference error ϵ_i

- caused by collisions between $\phi_0(\mathbf{x})$ or $\phi_u(\mathbf{x})$ with hash functions of other users
- shown to be small with high probability

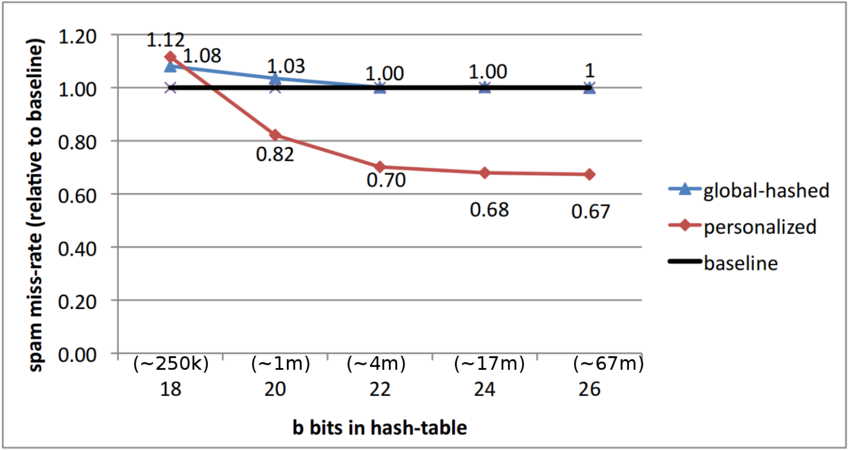
distortion error ϵ_d

- caused by self-collisions of hash functions
- shown to be small with high probability

Results

TECHNISCHE UNIVERSITÄT DARMSTADT

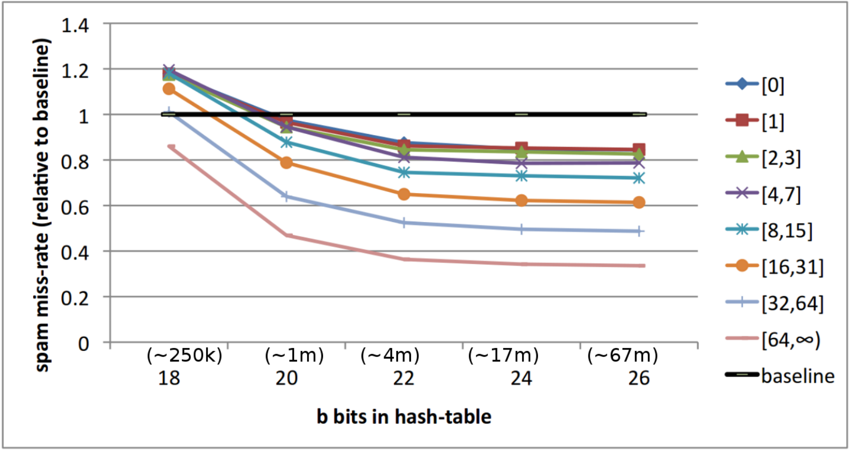
dimensionality of the raw data set: ~40 million



Results

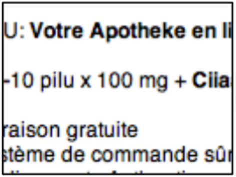
TECHNISCHE
UNIVERSITÄT
DARMSTADT

dimensionality of the raw data set: ~40 million



Summary

TECHNISCHE
UNIVERSITÄT
DARMSTADT


U: Votre Apotheke en li
-10 pilu x 100 mg + Ciia
raison gratuite
stème de commande sûr

x

NEU
Votre
Apotheke
...

NEU
USER123_NEU
Votre
USER123_Votre
Apotheke
USER123_Apotheke
...

ϕ

1
0
-1
0
0
-1
0
1
0
...

text document (email)

bag of words

bag of words (personalized)

hashed, sparse vector $\phi_0(x)+\phi_u(x)$