FNLP
More About Classification

Yansong Feng
fengyansong@pku.edu.cn

Wangxuan Institute of Computer Technology
Peking University

March 5, 2025

Outline

1. **Generative Models and Discriminative Models**

2. Model Evaluation

Two Views

- Goal: find a function $g : y = g(x)$
- Probabilistically:
  - $p(y|x)$
  - $g(x) = \arg\max_y p(y|x)$
- Two views:
  - discriminative models: learn $p(y|x)$ directly
  - generative models: learn $p(x, y)$ first

Examples from Dan Jurafsky

Your task is to distinguish cat from dog images

[images from imagenet]

Generative Models (from Dan)

Build your model to describe what is a cat/dog image

- dense and soft fur, 0.3 – 0.5m in size
- color in blue-grey to brownish yellow.
- long whiskers, round eyes
- short snout, short limbs, vertical ears

Generative Models (from Dan)

Build your model to describe what is a cat/dog image

- dense and soft fur, 0.3 – 0.5m in size
- color in blue-grey to brownish yellow.
- long whiskers, round eyes
- short snout, short limbs, vertical ears

- dense fur in varied colors
- 0.5 – 1m in size, long limbs
- short whiskers, oval eyes
- long snout, drooping/vertical ears

Generative Models (from Dan)

Build your model to describe what is a cat/dog image

- dense and soft fur, 0.3 – 0.5m in size
- color in blue-grey to brownish yellow.
- long whiskers, round eyes
- short snout, short limbs, vertical ears

- dense fur in varied colors
- 0.5 – 1m in size, long limbs
- short whiskers, oval eyes
- long snout, drooping/vertical ears

Now, given a new image, **run both models to see which is better**

Discriminative Models (from Dan)

Build your model to distinguish cat from dog, e.g., whether it is a cat/dog compared to dog/cat

Discriminative Models (from Dan)

Build your model to distinguish cat from dog, e.g., whether it is a cat/dog compared to dog/cat

- cat: long whiskers, short snout, ...
- dog: short whiskers, long snout, ...
- .....

Discriminative Models (from Dan)

Build your model to distinguish cat from dog, e.g., whether it is a cat/dog compared to dog/cat

- cat: long whiskers, short snout, ...
- dog: short whiskers, long snout, ...
- .....
- collars!, dogs have collars!

Discriminative Models (from Dan)

Build your model to distinguish cat from dog, e.g., whether it is a cat/dog compared to dog/cat

- cat: long whiskers, short snout, ...
- dog: short whiskers, long snout, ...
- .....
- collars!, dogs have collars!

Now, given a new image, **if there is a colloar, it is probably a dog!**

Generative Models v.s. Discriminative Models

Generative Models – Naïve Bayes

- $g(x) = \arg\max_y p(y)p(x|y)$

Discriminative Models – Log-Linear model

- $g(x) = \arg\max_y \sum_i \lambda_{f_i(x,y)} f_i(x,y)$

Generative Models v.s. Discriminative Models

Generative Models – Naïve Bayes

- $g(x) = \arg\max_y p(y)p(x|y)$

Discriminative Models – Log-Linear model

- $g(x) = \arg\max_y \sum_i \lambda_{f_i(x,y)} f_i(x,y)$
- $g(x) = \arg\max_y p(x|y) = \arg\max_y \frac{\exp \sum_i \lambda_{f_i(x,y)} f_i(x,y)}{\sum_{y'} \exp \sum_i \lambda_{f_i(x,y')} f_i(x,y')}$

Outline

Model Evaluation

In most times, we are given **a set of training data**, and asked to build a classifier, which will be submitted to somewhere else and evaluated in **an unknown test set** .

Model Evaluation

In most times, we are given **a set of training data**, and asked to build a classifier, which will be submitted to somewhere else and evaluated in **an unknown test set** .

- design your main model
- implement your system
- test and optimize your system
- submit it for a close evaluation

Model Evaluation

In most times, we are given **a set of training data**, and asked to build a classifier, which will be submitted to somewhere else and evaluated in **an unknown test set** .

- design your main model
- implement your system
- test and optimize your system
- submit it for a close evaluation

Training and Test Sets

- `Training dataset`: the resource that we can estimate parameters of a model
  - $p(s)$ and $p(v|s)$ in Naïve Bayes models
  - λs in Log-linear models
- `Test dataset`:
  - a held-out part
  - for evaluating our models
  - NOT part of training data: we should set our models done **BEFORE** we see the test data

Training and Test Sets

- Training dataset: the resource that we can estimate parameters of a model
  - $p(s)$ and $p(v|s)$ in Naïve Bayes models
  - λs in Log-linear models
- Test dataset:
  - a held-out part
  - for evaluating our models
  - NOT part of training data: we should set our models done **BEFORE** we see the test data

Example (Evaluate a Log-linear Classifier)

- use Training dataset to estimate those λs
- compute F-1 scores on the Test dataset

Training and Test Sets

- Training dataset: the resource that we can estimate parameters of a model
  - $p(s)$ and $p(v|s)$ in Naïve Bayes models
  - λs in Log-linear models
- Test dataset:
  - a held-out part
  - for evaluating our models
  - NOT part of training data: we should set our models done **BEFORE** we see the test data

Example (Evaluate a Log-linear Classifier)

- use Training dataset to estimate those λs
- compute F-1 scores on the Test dataset

a proper evaluation protocol

Held-out Data and Hyper-Parameter Estimations

We still have problems: How to choose from tens of feature templates in log-linear models ?

Choosing Hyper-Parameters

- hold out part of training data as a validation set
- also called development set
- creat several combinations of feature templates, $CF_1, CF_2, ..., CF_n$
- train the log-linear model with different feature templates on the training set, resulting in different models, e.g., $\mathcal{M}_1, \mathcal{M}_2, ..., \mathcal{M}_n$
- evaluate all models $\mathcal{M}_{1:n}$ on the validation set
- choose CF_* where the corresponding $\mathcal{M}_*$ obtains the best score (e.g., F_1) on the validation set

Held-out Data and Hyper-Parameter Estimations

We still have problems: How to choose from tens of feature templates in log-linear models ?

Choosing Hyper-Parameters

- hold out part of training data as a validation set
- also called development set
- creat several combinations of feature templates, $CF_1, CF_2, ..., CF_n$
- train the log-linear model with different feature templates on the training set, resulting in different models, e.g., $\mathcal{M}_1, \mathcal{M}_2, ..., \mathcal{M}_n$
- evaluate all models $\mathcal{M}_{1:n}$ on the validation set
- choose CF_* where the corresponding $\mathcal{M}_*$ obtains the best score (e.g., F_1) on the validation set

- Finally, report your results on the test set.

Cross Validation

Not sure at all: How will our models perform on a magic test set?

Cross Validation

Not sure at all: How will our models perform on a magic test set?
We expect:

the model performance should be stable and generalizable

Cross Validation

Not sure at all: How will our models perform on a magic test set?
We expect:

the model performance should be stable and generalizable

- K-fold cross-validation
  - partition the training set into K non-overlapping **folds**: $X_1, X_2, ..., X_K$
  - for $i \in \{1, ..., K\}$:
    - train your model on $X_{1:K} \backslash X_i$, using X_i as the development set
    - estimate the model performance on X_i, e.g., F_1^i
  - report the average and often the standard error:

$$F_1 = \frac{1}{K} \sum_{i=1}^{K} F_1^i$$

Cross Validation

Not sure at all: How will our models perform on a magic test set?
We expect:

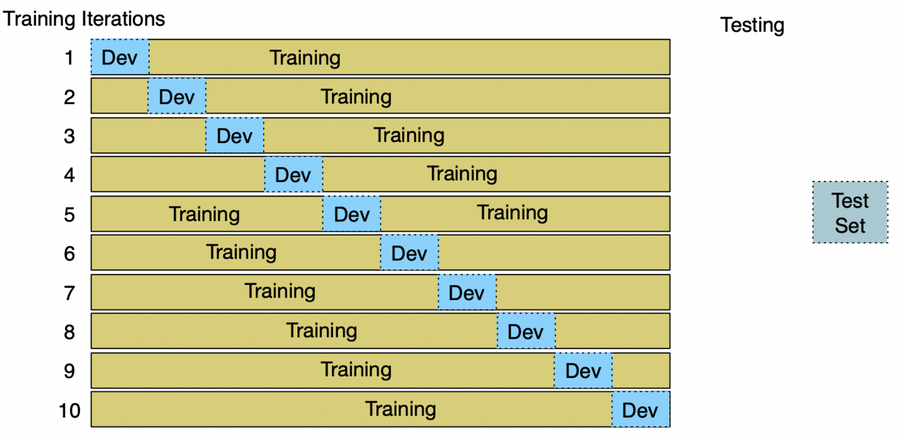
the model performance should be stable and generalizable

- K-fold cross-validation
  - partition the training set into K non-overlapping **folds**: $X_1, X_2, ..., X_K$
  - for $i \in \{1, ..., K\}$:
    - train your model on $X_{1:K} \backslash X_i$, using X_i as the development set
    - estimate the model performance on X_i, e.g., F_1^i
  - report the average and often the standard error:

$$F_1 = \frac{1}{K} \sum_{i=1}^{K} F_1^i$$

  - often, you can setup your model (hyper-)parameters through CV

Cross Validation



[Jurafsky and Martin, SLP3]