Machine Learning (CS 181):
7. Probabilistic Classification

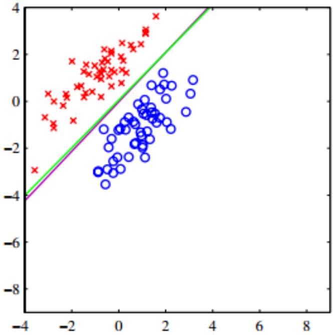
David Parkes and Sasha Rush

Contents

Contents

Last Class: Binary Classification

- Output space $\mathcal{Y}$ is a fixed set of classes.
- Simplest case $\mathcal{Y} = \{-1, 1\}$ (red/blue)
- Discriminant function:



Binary Classification

Generative Classification View

Model the joint probability of call y and data $\mathbf{x}$,

$$p(\mathbf{x}, y) = p(y)p(\mathbf{x}|y)$$

Note: Switch to different binary class representation,

$$\mathcal{Y} = \{0, 1\}$$

Process:

- Class is generated with probability $p(y)$
- Input $\mathbf{x}$ is generated conditional on class $p(\mathbf{x}|y)$

Class Probability

Choice of prior can depend on problem format,

- In binary case, we will use Bernoulli distribution,

$$p(y = 1; \theta) = \theta \quad \text{and} \quad p(y = 0; \theta) = 1 - \theta$$

  or more compactly (motivates notation change),

$$p(y; \theta) = \theta^y (1 - \theta)^{1-y}$$

Class-Conditional Probability

- Choice depends on modeling assumptions
- Select parameteric model for data given class

$$p(\mathbf{x}|y, \mathbf{w}_0, \mathbf{w}_1)$$

- Use continuous data, use multivariate Gaussian (HW)

$$p(\mathbf{x}|y = 0; \boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$$

$$p(\mathbf{x}|y = 1; \boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$$

- Today's class, discrete inputs,

$$p(\mathbf{x}|y; \boldsymbol{\pi}_0, \boldsymbol{\pi}_1)$$

Class-Conditional Probability

- Choice depends on modeling assumptions
- Select parameteric model for data given class

$$p(\mathbf{x}|y, \mathbf{w}_0, \mathbf{w}_1)$$

- Use continuous data, use multivariate Gaussian (HW)

$$p(\mathbf{x}|y = 0; \boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$$

$$p(\mathbf{x}|y = 1; \boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$$

- Today's class, discrete inputs,

$$p(\mathbf{x}|y; \boldsymbol{\pi}_0, \boldsymbol{\pi}_1)$$

Maximum Likelihood

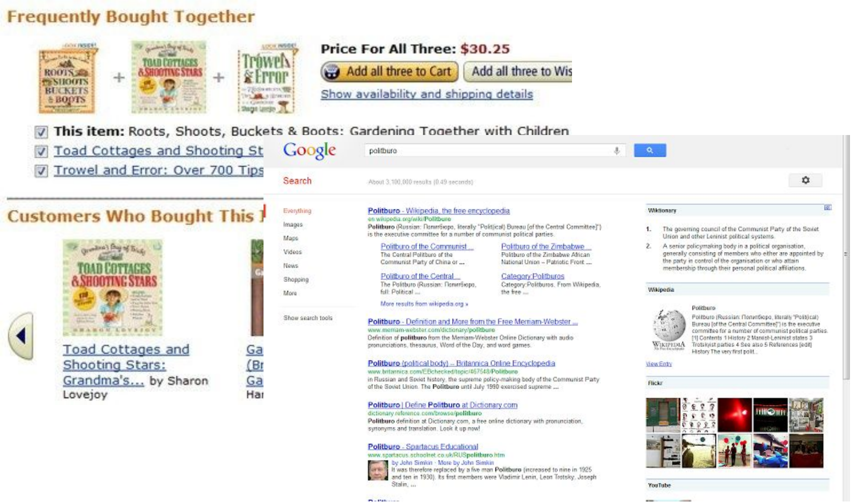
Same probabilistic approach as before,

1. Decide on generating process
2. Fix the parameterization of the model
3. Minimize negative log-likelihood of the data.

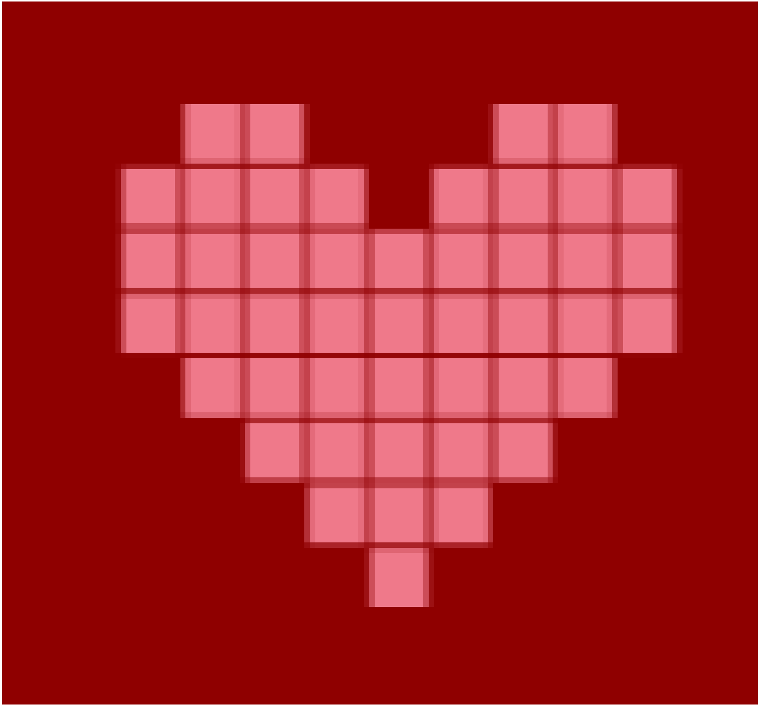
$$\min_{\boldsymbol{\pi}_0, \boldsymbol{\pi}_1, \theta} \mathcal{L}(\boldsymbol{\pi}_0, \boldsymbol{\pi}_1, \theta) = \min_{\boldsymbol{\pi}_0, \boldsymbol{\pi}_1, \theta} -\sum_{i=1}^{n} (\ln p(y_i; \theta) + \ln p(\mathbf{x}_i | y_i; \boldsymbol{\pi}_0, \boldsymbol{\pi}_1))$$

- Again, benefits will be having explicit probabilities for classes.
- Can also combine with Bayesian approaches from last class.

Probabilistic Classification



www.lifehacker.com.au chrome.google.com



Operation: Match

7. Dating someone of my own religion is:
   (1) unimportant
   (2) slightly important
   (3) moderately important
   (4) very important

Answer "1" (yes) or "2" (no) to each of the following five questions

My date's religious background may be:

8. Protestant	(1) Yes	(2) No
9. Catholic	(1) Yes	(2) No
10. Jewish	(1) Yes	(2) No
11. other	(1) Yes	(2) No
12. unaffiliated	(1) Yes	(2) No

In answering the following three questions refer to the table at right

1 13. My college class is: P1

2, 14. The ideal college class for my date is:

3 15. Men: I would consider dating a girl whose college class is as low as (indicate lowest acceptable college class):

Women: I would consider dating a man whose college class is as high as (indicate highest acceptable college class):

(1) first year in college
(2) second year in college
(3) third year in college
(4) fourth year in college
(5) graduated from college this year
(6) graduated from college one year ago
(7) graduated from college two years ago
(8) graduated from college three or more years ago

OPERATION MATCH

Though computer-dating was still a new concept in 1965-back then, the answers to personality questionnaires were converted into punch cards which were then fed into computers the size of small cars-two rival outfits had already popped up at Harvard: Operation Match and Contact Incorporated. Very little distinguished the two companies. Operation Match sold its questionnaires for $3. Contact charged $4. The Operation Match questionnaire was somewhat playful. Contact posed more serious questions. (One Operation Match survey question, "My race is...," had only three options... Still, both aimed to expand the campus dating pool from Wheaton to Wellesley, from Pembroke to Mount Holyoke.

Contents

Discrete Features / Basis

- Features so far, mostly continuous:
  - revenue, distance, crime rate, etc.
- For many domains, discrete features are more natural.
- Features represent (sparse) indicators or counts
  - visits, responses, properties, etc

$$\mathbf{x} = [0; 0; 1; \ldots; 0; 10; 0]$$

- Want to model with discrete distributions as opposed to Gaussians.

Discrete Features / Basis

- Features so far, mostly continuous:
  - revenue, distance, crime rate, etc.
- For many domains, discrete features are more natural.
- Features represent (sparse) indicators or counts
  - visits, responses, properties, etc

$$\mathbf{x} = [0; 0; 1; \dots; 0; 10; 0]$$

- Want to model with discrete distributions as opposed to Gaussians.

Earn a Degree based on your Life Experience
Obtain a Bachelors, Masters, MBA, or PhD based on your present knowledge and life experience.
No required tests, classes, or books. Confidentiality assured.
Join our fully recognized Degree Program.
Are you a truly qualified professional in your field but lack the appropriate, recognized documentation to achieve your goals?
Or are you venturing into a new field and need a boost to get your foot in the door so you can prove your capabilities?
Call us for information that can change your life and help you to achieve your goals!!!
CALL NOW TO RECEIVE YOUR DIPLOMA WITHIN 30 DAYS

Sentiment

Good Sentences

- A thoughtful, provocative, insistently humanizing film.
- Occasionally melodramatic, it's also extremely effective.
- Guaranteed to move anyone who ever shook, rattled, or rolled.

Bad Sentences

- A sentimental mess that never rings true.
- This 100-minute movie only has about 25 minutes of decent material.
- Here, common sense flies out the window, along with the hail of bullets, none of which ever seem to hit Sascha.

Features 1: Sparse Bag-of-Words Features

Example: Movie review input,

`A sentimental mess`

$$\boldsymbol{\phi}(\mathbf{x}) = [\phi_{\texttt{word:A}}(\mathbf{x}); \ldots; \phi_{\texttt{word:sentimental}}(\mathbf{x}); \phi_{\texttt{word:mess}}(\mathbf{x}); \ldots]$$

$$\boldsymbol{\phi}(\mathbf{x}) = \begin{bmatrix} 1 \\ \vdots \\ 1 \\ 1 \\ 0 \end{bmatrix} \begin{array}{l} \texttt{word:A} \\ \vdots \\ \texttt{word:mess} \\ \texttt{word:sentimental} \\ \end{array}$$

Features 2: Sparse Word Properties

Example: Spam Email

`Your diploma puts a UUNIVERSITY JOB PLACEMENT COUNSELOR at your disposal.`

$$\phi(\mathbf{x}) = \begin{array}{c} \\ \\ \\ \\ \\ \end{array}\begin{bmatrix} 1 \\ \vdots \\ 1 \\ 1 \\ 0 \end{bmatrix} \begin{array}{l} \texttt{misspelling} \\ \vdots \\ \texttt{capital} \\ \texttt{word:diploma} \\ \\ \end{array}$$

Contents

Multinomial Distribution

Let $\boldsymbol{\pi}$ represent probability of independent draws over $\{1 \dots m\}$ with $\sum_{j=1}^{m} \pi_j = 1$ and $\pi_j \geq 0$.

Then probability of count histogram $\mathbf{x}$, assuming $\sum_{j=1}^{m} x_j$ trials, (exp trick)

$$\begin{aligned} p(\mathbf{x}; \boldsymbol{\pi}) &= \frac{(\sum x_j)!}{\prod x_j!} \prod_{j=1}^{m} \pi_j^{x_j} \\ &\propto \prod_{j=1}^{m} \pi_j^{x_j} \end{aligned}$$

Fitting Multinomial

Maximum likelihood estimate over $\mathbf{x}_1 \dots \mathbf{x}_n$, (log trick)

$$\begin{aligned}
\underset{\boldsymbol{\pi} \geq 0}{\arg\max} \quad & \prod_{i=1}^{n} p(\mathbf{x}_i; \boldsymbol{\pi}) = \\
\underset{\boldsymbol{\pi} \geq 0}{\arg\max} \quad & \sum_{i=1}^{n} \sum_{j=1}^{m} x_{ij} \ln \pi_j \\
\text{s.t.} \quad & \sum_{j=1}^{m} \pi_j = 1
\end{aligned}$$

Maximum Likelihood (1)

Lagrange multipliers:

$$L(\boldsymbol{\pi}, \lambda) = \sum_{i=1}^{n} \sum_{j=1}^{m} x_{ij} \ln \pi_j + \lambda(1 - \sum_{j=1}^{m} \pi_j)$$

Solve π for all j,

$$\begin{aligned} \frac{\partial}{\partial \pi_j} L(\boldsymbol{\pi}, \lambda) &= \sum_{i=1}^{n} x_{ij}/\pi_j - \lambda = 0 \\ \pi_j &= \sum_{i=1}^{n} x_{ij}/\lambda \end{aligned}$$

Maximum Likelihood (1)

Lagrange multipliers:

$$L(\boldsymbol{\pi}, \lambda) = \sum_{i=1}^{n} \sum_{j=1}^{m} x_{ij} \ln \pi_j + \lambda(1 - \sum_{j=1}^{m} \pi_j)$$

Solve $\boldsymbol{\pi}$ for all j,

$$\begin{aligned} \frac{\partial}{\partial \pi_j} L(\boldsymbol{\pi}, \lambda) &= \sum_{i=1}^{n} x_{ij}/\pi_j - \lambda = 0 \\ \pi_j &= \sum_{i=1}^{n} x_{ij}/\lambda \end{aligned}$$

Maximum Likelihood (2)

Lagrange multipliers:

$$L(\boldsymbol{\pi}, \lambda) = \sum_{i=1}^{n} \sum_{j=1}^{m} x_{ij} \ln \pi_j + \lambda(1 - \sum_{j=1}^{m} \pi_j)$$

$$\pi_j = \sum_{i=1}^{n} x_{ij} / \lambda$$

Solve λ,

$$\begin{aligned} \frac{\partial}{\partial \lambda} L(\boldsymbol{\pi}, \lambda) &= \sum_{j=1}^{m} \pi_j = 1 \\ \sum_{j=1}^{m} \sum_{i=1}^{n} x_{ij} / \lambda &= 1 \\ \lambda &= \sum_{i=1}^{n} \sum_{j=1}^{m} x_{ij} \end{aligned}$$

Maximum Likelihood (2)

Lagrange multipliers:

$$L(\boldsymbol{\pi}, \lambda) = \sum_{i=1}^{n} \sum_{j=1}^{m} x_{ij} \ln \pi_j + \lambda(1 - \sum_{j=1}^{m} \pi_j)$$

$$\pi_j = \sum_{i=1}^{n} x_{ij} / \lambda$$

Solve λ,

$$\begin{aligned} \frac{\partial}{\partial \lambda} L(\boldsymbol{\pi}, \lambda) &= \sum_{j=1}^{m} \pi_j = 1 \\ \sum_{j=1}^{m} \sum_{i=1}^{n} x_{ij} / \lambda &= 1 \\ \lambda &= \sum_{i=1}^{n} \sum_{j=1}^{m} x_{ij} \end{aligned}$$

Multinomial MLE Matrix Form

The maximum likelihood parameters become the counts over the data:

$$\hat{\boldsymbol{\pi}} = \frac{\sum_{i=1}^{n} \mathbf{x}_i}{\sum_{i=1}^{n} \sum_{j=1}^{m} x_{ij}} = \frac{\mathbf{X}^\top \mathbf{1}}{\mathbf{1}\mathbf{X}\mathbf{1}}$$

Mulitinomial Binary-Class Naive Bayes

Assume each discrete features $\mathbf{x}$, model joint probability as

$$p(\mathbf{x}, y) = p(y; \theta) p(\mathbf{x}|y; \boldsymbol{\pi}_0, \boldsymbol{\pi}_1)$$

- Class probability, Bernoulli $p(y; \theta)$
- Features conditional Multinomial:

$$p(\mathbf{x}|y; \boldsymbol{\pi}_0, \boldsymbol{\pi}_1) \propto \prod_{j=1}^{m} \pi_{yj}^{x_j}$$

- Multinomial: each feature is generated independently (Naive)
- Conditional classification using Bayes rule (Bayes)

$$p(y|\mathbf{x}) \propto p(y; \theta) p(\mathbf{x}|y; \boldsymbol{\pi}_0, \boldsymbol{\pi}_1)$$

Mulitinomial Binary-Class Naive Bayes

Assume each discrete features $\mathbf{x}$, model joint probability as

$$p(\mathbf{x}, y) = p(y; \theta) p(\mathbf{x}|y; \boldsymbol{\pi}_0, \boldsymbol{\pi}_1)$$

- Class probability, Bernoulli $p(y; \theta)$
- Features conditional Multinomial:

$$p(\mathbf{x}|y; \boldsymbol{\pi}_0, \boldsymbol{\pi}_1) \propto \prod_{j=1}^{m} \pi_{yj}^{x_j}$$

- Multinomial: each feature is generated independently (Naive)
- Conditional classification using Bayes rule (Bayes)

$$p(y|\mathbf{x}) \propto p(y; \theta) p(\mathbf{x}|y; \boldsymbol{\pi}_0, \boldsymbol{\pi}_1)$$

Loss for Multinomial Naive Bayes

$$\max_{\theta,\boldsymbol{\pi}_0,\boldsymbol{\pi}_1} \sum_{i=1}^{n} \ln p(\mathbf{x}_i, y_i) = \max_{\boldsymbol{\pi}_0,\boldsymbol{\pi}_1} \sum_{i=1}^{n} \ln p(\mathbf{x}_i | y_i; \boldsymbol{\pi}_0, \boldsymbol{\pi}_1) + \max_{\theta} \sum_{i=1}^{n} \ln p(y_i; \theta)$$

- Class probability

$$\hat{\theta} = \frac{\mathbf{1}\mathbf{y}}{n}$$

- Class-Conditional probability

$$\hat{\boldsymbol{\pi}}_0 = \frac{\sum_{i=1}^{n} (1 - y_i)\mathbf{x}_i}{\sum_{i=1}^{n} \sum_{j=1}^{m} (1 - y_i) x_{ij}}$$

$$\hat{\boldsymbol{\pi}}_1 = \frac{\sum_{i=1}^{n} y_i \mathbf{x}_i}{\sum_{i=1}^{n} \sum_{j=1}^{m} y_i x_{ij}}$$

Loss for Multinomial Naive Bayes

$$\max_{\theta,\boldsymbol{\pi}_0,\boldsymbol{\pi}_1} \sum_{i=1}^{n} \ln p(\mathbf{x}_i, y_i) = \max_{\boldsymbol{\pi}_0,\boldsymbol{\pi}_1} \sum_{i=1}^{n} \ln p(\mathbf{x}_i | y_i; \boldsymbol{\pi}_0, \boldsymbol{\pi}_1) + \max_{\theta} \sum_{i=1}^{n} \ln p(y_i; \theta)$$

- Class probability

$$\hat{\theta} = \frac{\mathbf{1y}}{n}$$

- Class-Conditional probability

$$\hat{\boldsymbol{\pi}}_0 = \frac{\sum_{i=1}^{n}(1-y_i)\mathbf{x}_i}{\sum_{i=1}^{n}\sum_{j=1}^{m}(1-y_i)x_{ij}}$$

$$\hat{\boldsymbol{\pi}}_1 = \frac{\sum_{i=1}^{n} y_i\mathbf{x}_i}{\sum_{i=1}^{n}\sum_{j=1}^{m} y_i x_{ij}}$$

Loss for Multinomial Naive Bayes

$$\max_{\theta, \boldsymbol{\pi}_0, \boldsymbol{\pi}_1} \sum_{i=1}^{n} \ln p(\mathbf{x}_i, y_i) = \max_{\boldsymbol{\pi}_0, \boldsymbol{\pi}_1} \sum_{i=1}^{n} \ln p(\mathbf{x}_i | y_i; \boldsymbol{\pi}_0, \boldsymbol{\pi}_1) + \max_{\theta} \sum_{i=1}^{n} \ln p(y_i; \theta)$$

- Class probability

$$\hat{\theta} = \frac{\mathbf{1y}}{n}$$

- Class-Conditional probability

$$\hat{\boldsymbol{\pi}_0} = \frac{\sum_{i=1}^{n} (1 - y_i)\mathbf{x}_i}{\sum_{i=1}^{n} \sum_{j=1}^{m} (1 - y_i) x_{ij}}$$

$$\hat{\boldsymbol{\pi}_1} = \frac{\sum_{i=1}^{n} y_i \mathbf{x}_i}{\sum_{i=1}^{n} \sum_{j=1}^{m} y_i x_{ij}}$$

Bayesian Multinomial Naive Bayes

In brief: both class and class-conditional can be estimated with non-MLE methods using priors.

- Class probability (Beta-Bernoulli)
- Class-Conditional probability (Dirichlet-Multinomial)

Effect: add α pseudo-counts to features, helps with rare or unseen features.

Bayesian Multinomial Naive Bayes

In brief: both class and class-conditional can be estimated with non-MLE methods using priors.

- Class probability (Beta-Bernoulli)
- Class-Conditional probability (Dirichlet-Multinomial)

Effect: add α pseudo-counts to features, helps with rare or unseen features.

Multinomial Naive Bayes (In Practice)

$$\ln p(y|\mathbf{x}) \propto \ln p(\mathbf{x}|y) + \ln p(y)$$

How do you decide what class? (Discriminant function)

$$\begin{aligned} h(\mathbf{x}) &= (\ln p(\mathbf{x}|y=1) + \ln p(y=1)) - (\ln p(\mathbf{x}|y=0) + \ln p(y=0)) \\ &= [\ln \prod_{j=1}^{m} \pi_{1j}^{x_j} - \ln \prod_{j=1}^{m} \pi_{0j}^{x_j}] + [\ln\theta - \ln(1-\theta)] \\ &= \sum_{j=1}^{m} x_j \ln \frac{\pi_{1j}}{\pi_{0j}} + \ln \frac{\theta}{1-\theta} = \mathbf{x}^\top (\ln \frac{\boldsymbol{\pi}_1}{\boldsymbol{\pi}_0}) + \ln \frac{\theta}{1-\theta} \end{aligned}$$

But this formulation is a linear model.

$$h(\mathbf{x};\mathbf{w}) = \mathbf{x}^\top \mathbf{w} + w_0$$

Multinomial Naive Bayes (In Practice)

$$\ln p(y|\mathbf{x}) \propto \ln p(\mathbf{x}|y) + \ln p(y)$$

How do you decide what class? (Discriminant function)

$$\begin{aligned} h(\mathbf{x}) &= (\ln p(\mathbf{x}|y=1) + \ln p(y=1)) - (\ln p(\mathbf{x}|y=0) + \ln p(y=0)) \\ &= [\ln \prod_{j=1}^{m} \pi_{1j}^{x_j} - \ln \prod_{j=1}^{m} \pi_{0j}^{x_j}] + [\ln \theta - \ln(1-\theta)] \\ &= \sum_{j=1}^{m} x_j \ln \frac{\pi_{1j}}{\pi_{0j}} + \ln \frac{\theta}{1-\theta} = \mathbf{x}^\top (\ln \frac{\boldsymbol{\pi}_1}{\boldsymbol{\pi}_0}) + \ln \frac{\theta}{1-\theta} \end{aligned}$$

But this formulation is a linear model.

$$h(\mathbf{x}; \mathbf{w}) = \mathbf{x}^\top \mathbf{w} + w_0$$

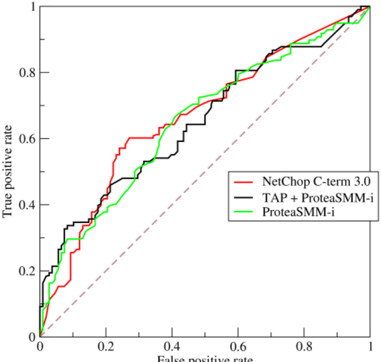
Role of Class Probabilities

$$\mathbf{x}^\top (\ln \frac{\boldsymbol{\pi}_1}{\boldsymbol{\pi}_0}) + \ln \frac{\theta}{1-\theta}$$

Class probabilities $p(y;\theta)$ determine the threshold.

$$h(\mathbf{x};\mathbf{w}) = \mathbf{x}^\top \mathbf{w} + w_0$$

Brings us back again to ROC curves.



Naive Bayes In Practice

- Super fast to train.
- Relatively interpretable.
- Performs quite well on small datasets.

Method	RT-s	MPQA	CR	Subj.
MNB-uni	77.9	85.3	79.8	**92.6**
MNB-bi	**79.0**	**86.3**	80.0	**93.6**
SVM-uni	76.2	86.1	79.0	90.8
SVM-bi	77.7	**86.7**	80.8	91.7
NBSVM-uni	**78.1**	85.3	80.5	92.4
NBSVM-bi	**79.4**	**86.3**	**81.8**	**93.2**
RAE	76.8	85.7	–	–
RAE-pretrain	77.7	**86.4**	–	–
Voting-w/Rev.	63.1	81.7	74.2	–

(RT-S [movie review], CR [customer reports], MPQA [opinion polarity], SUBJ [subjectivity])

Naive Bayes Example

Contents

Generative versus Discriminative Model

- Generative models: Parameterize Joint Distribution

$$\arg\max_{\mathbf{w}} \prod_i p(\mathbf{x}_i, y_i; \mathbf{w})$$

- Discriminative model: Parameterize Conditional Distribution

$$\arg\max_{\mathbf{w}} \prod_i p(y_i | \mathbf{x}_i; \mathbf{w})$$

- Why does this matter?

Linear Discriminative Model

Set log prob to be proportional to some linear model, shorthand h

$$\ln p(y = 1 | \mathbf{x}; \mathbf{w}) \propto \mathbf{w}^\top \mathbf{x} + w_0 = h$$

As before threshold at $h > 0$,

$$\ln p(y = 0 | \mathbf{x}; \mathbf{w}) \propto 0$$

Now remove log and normalize,

$$\begin{aligned} p(y = 1 | \mathbf{x}; \mathbf{w}) &= \frac{\exp h}{\exp h + \exp 0} = (1 + \exp -h)^{-1} \\ p(y = 0 | \mathbf{x}; \mathbf{w}) &= \frac{\exp 0}{\exp h + \exp 0} = (1 + \exp h)^{-1} \end{aligned}$$

Call this function the logistic sigmoid activation.

$$\sigma(h) = (1 + \exp -h)^{-1}$$

Linear Discriminative Model

Set log prob to be proportional to some linear model, shorthand h

$$\ln p(y=1|\mathbf{x};\mathbf{w}) \propto \mathbf{w}^\top \mathbf{x} + w_0 = h$$

As before threshold at $h > 0$,

$$\ln p(y=0|\mathbf{x};\mathbf{w}) \propto 0$$

Now remove log and normalize,

$$\begin{aligned} p(y=1|\mathbf{x};\mathbf{w}) &= \frac{\exp h}{\exp h + \exp 0} = (1+\exp -h)^{-1} \\ p(y=0|\mathbf{x};\mathbf{w}) &= \frac{\exp 0}{\exp h + \exp 0} = (1+\exp h)^{-1} \end{aligned}$$

Call this function the logistic sigmoid activation.

$$\sigma(h) = (1+\exp -h)^{-1}$$

Linear Discriminative Model

Set log prob to be proportional to some linear model, shorthand h

$$\ln p(y = 1|\mathbf{x}; \mathbf{w}) \propto \mathbf{w}^\top \mathbf{x} + w_0 = h$$

As before threshold at $h > 0$,

$$\ln p(y = 0|\mathbf{x}; \mathbf{w}) \propto 0$$

Now remove log and normalize,

$$\begin{aligned} p(y = 1|\mathbf{x}; \mathbf{w}) &= \frac{\exp h}{\exp h + \exp 0} = (1 + \exp -h)^{-1} \\ p(y = 0|\mathbf{x}; \mathbf{w}) &= \frac{\exp 0}{\exp h + \exp 0} = (1 + \exp h)^{-1} \end{aligned}$$

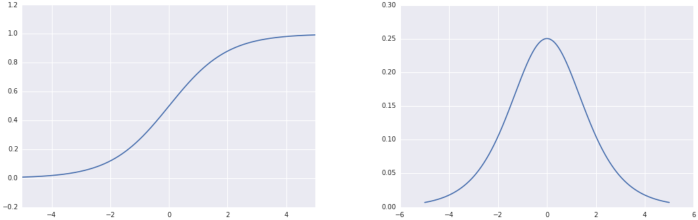
Call this function the logistic sigmoid activation.

$$\sigma(h) = (1 + \exp -h)^{-1}$$

Contents

(Logistic) Sigmoid Activation

$$\sigma(h) = (1 + \exp(-h))^{-1}$$



Sigmoid Function and Derivative

- "Squashes" $\mathbb{R}$ to a probabilities.

Logistic Regression

Linear model converted to probability estimated by sigmoid

$$p(y=1|\mathbf{x};\mathbf{w}) = \sigma(\mathbf{w}^\top \mathbf{x} + w_0) = (1+\exp(-h))^{-1}$$

$$p(y=0|\mathbf{x};\mathbf{w}) = 1-\sigma(\mathbf{w}^\top \mathbf{x} + w_0) = (1+\exp h)^{-1}$$

- Linear “Regression” transformed to probability estimate.
- Name is confusing, mostly used for *classification*.

Fitting Model

Reminder:

$$p(y=1|\mathbf{x};\mathbf{w}) = \sigma(\mathbf{w}^\top \mathbf{x} + w_0) = (1+\exp(-h))^{-1}$$

$$p(y=0|\mathbf{x};\mathbf{w}) = 1 - \sigma(\mathbf{w}^\top \mathbf{x} + w_0) = (1+\exp h)^{-1}$$

As this is now a probabilistic model, can fit with MLE.

$$\begin{aligned}
\mathcal{L}(\mathbf{w}) &= -\sum_{i=1}^{n} \ln p(y_i|\mathbf{x}_i;\mathbf{w}) = -\sum_{i=1}^{n} \ln \sigma(h)^{y_i}(1-\sigma(h))^{1-y_i} \\
&= \sum_{i=1}^{n} y_i \ln(1+\exp(-h)) + (1-y_i)\ln(1+\exp h)
\end{aligned}$$

Likelihood and Estimation

Reminder:

$$\begin{aligned} h &= \mathbf{w}^\top \mathbf{x}_i + w_0 \\ \mathcal{L}(\mathbf{w}) &= \sum_{i=1}^{n} y_i \ln(1 + \exp(-h)) + (1 - y_i) \ln(1 + \exp h) \end{aligned}$$

Take gradients:

$$\begin{aligned} \frac{\partial}{\partial \mathbf{w}} \ln(1 + \exp(-h)) = & \quad -\mathbf{x}_i \frac{\exp -h}{1 + \exp(-h)} = -\mathbf{x}_i p(y_i = 0|\mathbf{x}) \\ \frac{\partial}{\partial \mathbf{w}} \ln(1 + \exp h) = & \quad \mathbf{x}_i \frac{\exp h}{1 + \exp h} = \mathbf{x}_i p(y_i = 1|\mathbf{x}) \end{aligned}$$

$$\frac{\partial}{\partial \mathbf{w}} \mathcal{L}(\mathbf{w}) = \sum_{i=1}^{n} -y_i \mathbf{x}_i p(y_i = 0|\mathbf{x}_i) + (1 - y_i) \mathbf{x}_i p(y_i = 1|\mathbf{x}_i)$$

Likelihood and Estimation

Reminder:

$$
\begin{aligned}
h &= \mathbf{w}^\top \mathbf{x}_i + w_0 \\
\mathcal{L}(\mathbf{w}) &= \sum_{i=1}^{n} y_i \ln(1+\exp(-h)) + (1-y_i)\ln(1+\exp h)
\end{aligned}
$$

Take gradients:

$$
\begin{aligned}
\frac{\partial}{\partial \mathbf{w}} \ln(1+\exp(-h)) = & \quad -\mathbf{x}_i \frac{\exp -h}{1+\exp(-h)} = -\mathbf{x}_i p(y_i = 0|\mathbf{x}) \\
\frac{\partial}{\partial \mathbf{w}} \ln(1+\exp h) = & \quad \mathbf{x}_i \frac{\exp h}{1+\exp h} = \mathbf{x}_i p(y_i = 1|\mathbf{x})
\end{aligned}
$$

$$
\frac{\partial}{\partial \mathbf{w}} \mathcal{L}(\mathbf{w}) = \sum_{i=1}^{n} -y_i \mathbf{x}_i p(y_i = 0|\mathbf{x}_i) + (1-y_i)\mathbf{x}_i p(y_i = 1|\mathbf{x}_i)
$$

Recall: Perceptron Algorithm

1. Iterate over the data:
   - If correct $(y_i = \hat{y}_i)$, do nothing.
   - If incorrect, add/subtract $\eta \times \mathbf{x}_i$ to weights
2. If errors, repeat process.
3. Otherwise separator is found.

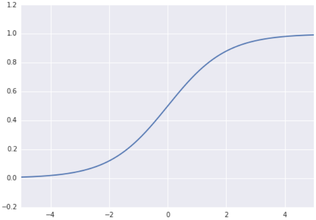
SGD on Logistic Regression

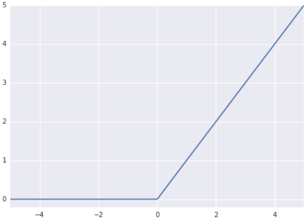
1. Iterate over the data:
   - Compute $p(y_i = 1|\mathbf{x}_i)$.
   - If $(y_i = 1)$, add $\eta \times \mathbf{x}_i p(y_i = 0|\mathbf{x}_i)$ to $\mathbf{w}$
   - If $(y_i = 0)$, add $\eta \times -\mathbf{x}_i p(y_i = 1|\mathbf{x}_i)$ to $\mathbf{w}$
2. Repeat until convergence.

Guaranteed to maximize conditional likelihood of data.

Algorithms Comes from Activations

- What is the difference between Perceptron and Logistic Regression?





Sigmoid Function versus Hinge

Contents

Multiclass

- ⋆⋆⋆⋆

  I visited The Abbey on several occasions on a visit to Cambridge and found it to be a solid, reliable and friendly place for a meal.

- ⋆⋆

  However, the food leaves something to be desired. A very obvious menu and average execution

- ⋆⋆⋆⋆⋆

  Fun, friendly neighborhood bar. Good drinks, good food, not too pricey. Great atmosphere!

Multiclass Naive Bayes

Multiclass outputs: $\mathcal{Y} = \{C_1, \dots C_c\}$

One-hot vectors

$$C_2 = [0; 1; 0; \dots; 0]$$

- Class distribution uses categorical for each $k \in \{1 \dots c\}$,

$$p(\mathbf{y} = C_k; \boldsymbol{\pi}) = \pi_k$$

- Class-Conditional uses separate parameters for each class

$$\{\mathbf{w}_\ell\}_{\ell=1}^{c}$$

$$p(\mathbf{x}|\mathbf{y}; \{\mathbf{w}_\ell\}_{\ell=1}^{c})$$

In homework, these each parameterize a multivariate Gaussian.

Multiclass Logistic Regression

- Multiclass logistic regression uses a $\mathbf{w}_\ell$ for each class.
- Generalization of sigmoid is softmax function.

$$p(\mathbf{y} = C_k | \mathbf{x}; \{\mathbf{w}_\ell\}_{\ell=1}^{c}) = \frac{\exp(\mathbf{x}^\top \mathbf{w}_k)}{\sum_\ell \exp(\mathbf{x}^\top \mathbf{w}_\ell)}$$

(Derivation and exploration on homework.)

Neural Network Preview: Softmax In Action