Probabilistic Learning with Complete Data: Learning Goals

1. Applying Bayes Rule to Prediction
2. Maximum likelihood parameter learning
3. Naïve Bayes Classifier

- Russell 20.1, 20.2.1-20.2.3

Many Classification Approaches

- Decision trees
- Linear models, e.g., regression, ridge, lasso
- Neural Nets
- Naïve Bayes
- kNN
- SVM
- Ensembles
- Random forests
- Hidden Markov Models
- Conditional Random Fields

The Bayes (Update) Rule

- Pr (A | B) = Pr (A B) / Pr (B)
- For us, interested in Pr (hypothesis | data)
- That is, Pr (h | d) = Pr (h d) / Pr (d)
- But we may be evaluating multiple hypotheses $h_1, \ldots, h_u$ at the same time
- The term Pr (d) can be further broken down to Pr (d) = Σ_i Pr (d | h_i) Pr (h_i)

E.g. #1: Spam Mails

- Suppose we have 100 emails: 60 spam, 40 “good”
- Within the 60 spam mails, the word “buy” occur in 36 emails
- Within the 40 good mails, “buy” occurs in 4 emails
- Now given a new email E containing the word “buy”, what is the probability that E is a spam mail?

Clicker Question

- What is the probability that E is a spam mail given that it contains the word “buy”?

a) 0.6

b) 0.7

c) 0.8

d) 0.9

E.g. #1: Spam Mails

- Pr (E = spam | ”buy”) = Pr (E = spam and “buy”)/ Pr(“buy”)
- Pr (E = spam and “buy”) = 36/100
- Pr(“buy”) = Pr(“buy” | spam) Pr (spam) +
  Pr(“buy” | good) Pr (good)
  = (36/60) * (60/100) + (4/40)*(40/100)
  = (36+4)/100

Thus, Pr (E = spam | ”buy”) = 36/(36+4) = 0.9

E.g. #2: Spam Mails II

- Suppose we have 100 emails: 60 spam, 40 “good”
- Within the 60 spam mails, the words “buy” and “lottery” occur in 12 emails
- Within the 40 good mails, “buy” and “lottery” occur in 1 email
- Now given a new email E containing both “buy”and “lottery”, what is the probability that E is a spam mail?

E.g. #2: Spam Mails II

- Pr (E = spam | ”buy” and “lottery”) = Pr (E = spam and “buy” and “lottery”)/Pr(“buy” and “lottery”)
- Pr (E = spam and “buy” and “lottery”) = 12/100
- Pr(“buy”) = Pr(“buy” and “lottery” | spam) Pr (spam) + Pr(“buy” and “lottery” | good) Pr (good)

  = (12/60) * (60/100) + (1/40)*(40/100)

  = (12+1)/100

Thus, Pr (E = spam | ”buy”and “lottery”) = 12/(12+1) = 0.93

One more observation from Spam Mails II

- Suppose we don’t have the direct data on Pr(“buy” and “lottery” | spam)
  - Within the 60 spam mails, the words “buy” and “lottery” occur in 12 emails
  - Instead, we know that the word “buy” occurs in 36 emails and the word “lottery” occurs in 18 emails
- Then we *assume* feature independence:

Pr(“buy” and “lottery” | spam)

= Pr(“buy” | spam) * Pr(“lottery” | spam)

= (36/60) * (18/60) = 0.18

E.g. #3: Cherry-Lime Candies

- Multiple hypotheses:
  - h_1: 100% cherry (prior = 0.1)
  - h_2: 3/4 cherry, 1/4 lime (prior = 0.2)
  - h_3: 1/2 cherry, 1/2 lime (prior = 0.4)
  - h_4: 1/4 cherry, 3/4 lime (prior = 0.2)
  - h_5: 100% lime (prior = 0.1)
- Update the posterior probs of the hypotheses, given that d_1 = lime. Then what is the prob that the next candy, d_2, is also lime?

E.g. #3: Cherry-Lime Candies (2)

- $Pr(h_i | d_1) = Pr(h_i \text{ and } d_1) / Pr(d_1)$
  - $Pr(h_1 \text{ and } d_1) = Pr(d_1 | h_1) Pr(h_1) = 0$
  - $Pr(h_2 \text{ and } d_1) = Pr(d_1 | h_2) Pr(h_2) = 1/4 * 2/10 = 2/40$
  - $Pr(h_3 \text{ and } d_1) = Pr(d_1 | h_3) Pr(h_3) = 2/4 * 4/10 = 8/40$
  - $Pr(h_4 \text{ and } d_1) = Pr(d_1 | h_4) Pr(h_4) = 3/4 * 2/10 = 6/40$
  - $Pr(h_5 \text{ and } d_1) = Pr(d_1 | h_5) Pr(h_5) = 4/4 * 1/10 = 4/40$
- $Pr(d_1) = \Sigma_i Pr(d_1 | h_i) Pr(h_i) = (2+8+6+4)/40$
- Thus, $Pr(h_i | d_1) = 0, 0.1, 0.4, 0.3, 0.2$ respectively (see fig 20.1a)

E.g. #3: Cherry-Lime Candies (3)

- Seeing that the first candy is lime, is the second candy more likely to be lime or cherry? (Now we are predicting)
- $Pr(d_2 = lime | d_1) = Pr(d_2 \text{ and } d_1) / Pr(d_1)$
- $Pr(d_2 d_1) = \Sigma_i Pr(d_2 d_1 | h_i) Pr(h_i)$
  - $Pr(d_2 d_1 | h_2) Pr(h_2) = 1/4 * 1/4 * 2/10 = 2/160$
  - $Pr(d_2 d_1 | h_3) Pr(h_3) = 2/4 * 2/4 * 4/10 = 16/160$
  - $Pr(d_2 d_1 | h_4) Pr(h_4) = 3/4 * 3/4 * 2/10 = 18/160$
  - $Pr(d_2 d_1 | h_5) Pr(h_5) = 4/4 * 4/4 * 1/10 = 16/160$
- $Pr(d_1) = 0.5$ from previous slide
- Thus, $Pr(d_2 = lime | d_1) = (52/160)/0.5 = 0.65$ (see fig 20.1b)

E.g. #3: Cherry-Lime Candies (4)

- Suppose that d_2 is lime again. Update the posterior probs of the hypotheses, given that $d_1 = d_2 =$ lime. Predict again the chance that the next candy, d_3, is also lime.
- From Bayes, $Pr(h_i | d_1\, d_2) = Pr(d_1\, d_2 | h_i)\, Pr(h_i) \,/\, Pr(d_1\, d_2)$
- From the previous slide, we have: $Pr(h_i | d_1\, d_2) =$ 2/52, 16/52, 18/52, 16/52 respectively for $h_2\, h_3\, h_4\, h_5$ respectively (see fig 20.1a)
- $Pr\,(d_3 = \text{lime} | d_1\, d_2) = Pr(d_1\, d_2\, d_3)/Pr(d_1\, d_2) = 19/26$ (see fig 20.1b)

MAP Prediction

- Optimal Bayesian prediction can be computationally demanding
- For many real-world problems, predictions are made from a single, most probable hypothesis, called the Maximum A Posteriori hypothesis
- E.g., declaring h_5 in our candy example as the MAP hypothesis after seeing 3 lime candies
- Even in our example, predicting using the MAP hypothesis can lead to lower accuracy

Using or Not using Priors

- In our candy example, the prior prob $\Pr(h_i)$ has a large impact
- To minimize overfitting, one can use the prior probs to penalize complex hypotheses, i.e., lower $\Pr(h_i)$ if h_i is complex
- On the other hand, one may question the subjective nature of the priors, in which case a uniform prior is assigned
- This is called Maximum-Likelihood (ML) learning (an example we have seen before)

ML Parameter Learning

- Continuing with our candy example, suppose the proportion of cherries-lime is unknown, called θ
- Our task is to estimating θ, given that our training data consist of c cherries and m = (N – c) limes
- Log likelihood of h_θ given d = $d_1,\ldots,d_N$ is:

  $L(d \mid h_\theta) = \log \Pr(d \mid h_\theta)$

  $= \log \left(\prod_{i=1}^{N} \Pr(d_i \mid h_\theta)\right)$

  $= \log (\theta^c (1 - \theta)^m) = c \log \theta + m \log (1 - \theta)$

ML Parameter Learning (2)

- To find the ML value of θ , differential L wrt θ and set the resulting expression to 0 (as in calculus):

  $d\,L(d \mid h_\theta) / d\,\theta = c / \theta - (m / (1- \theta))$

- Setting this to 0 gives $\theta = c/(c + m) = c / N$
- That is, the maximum likelihood hypothesis, h_{ML}, suggests that the actual proportion of the cherries in the bag is equal to the observed proportion of the data so far (Quite natural!)

ML Learning of Multiple Parameters

- In addition to θ , suppose the wrapper of a candy is probabilistically chosen based on an unknown distribution, but depending on the flavors:

  θ_1 = Pr(wrap = red | cherry)

  θ_2 = Pr(wrap = red | lime)

- thus, Pr(wrap = green | cherry) = 1- θ_1; and

  Pr(wrap = green | lime) = 1- θ_2

- Estimate θ , θ_1, θ_2 based on the data: from N unwrapped candies, there are c cherries (rc of which wrapped in red and gc wrapped in green) and m limes (rm of which wrapped in red and gm of which wrapped in green)

ML Multiple Parameters (2)

- Likehihood Pr(d | $h_{\theta,\theta 1,\theta 2}$)
  $= \theta^{c} (1 - \theta)^{m} \; \theta_1^{rc} (1 - \theta_1)^{gc} \; \theta_2^{rm} (1 - \theta_2)^{gm}$
- Log likelihood turns the product into a sum
- To maximize, take (partial) derivative wrt each of the three parameters θ, θ_1, θ_2 and set to 0
- Get three independent equations, each involving only one parameter (due to the sum, not the product)
  - $c/\theta - (m/(1-\theta)) = 0$ means $\theta = c/(c + m)$
  - $rc/\theta_1 - (gc/(1-\theta_1)) = 0$ means $\theta_1 = rc/(rc + gc)$
  - $rm/\theta_2 - (gm/(1-\theta_2)) = 0$ means $\theta_2 = rm/(rm + gm)$
- Key: learning multiple ML parameters turns into separate learning problems, one for each parameter!

ML Learning of a Continuous Variable

- So far, we have talked about ML learning of a discrete model
- Next we talk talk about ML learning of a single Gaussian variable – because of its ubiquity in real life
- Same recipe
  1. Write down an expression for the likelihood of the training data
  2. Obtain the derivative of the log likelihood wrt the parameters
  3. Find the ML parameter values by setting the derivative to 0
- A Gaussian has two parameters: mean μ, std deviation σ
- ML learning μ = sampled mean = $\Sigma_i \; x_i / N$
- ML learning σ^2 = sampled variance = $\Sigma_i \; (x_i - \mu)^2 / N$

Naïve Bayes Classifier

- Here the hypothesis is the class variable C, given the observed attr values $x_1, \ldots, x_n$
- $\Pr(C \mid x_1, \ldots, x_n) = \alpha \Pr(C) \prod \Pr(x_i \mid C)$ is the probability of prediction of each class
- Use the ML learning of all the parameters
- Then pick the most likely class
  - α is just a normalizing constant over all the classes
  - In picking the most likely class, suffices to ignore it and compare the product of the remaining terms

Naïve Bayes E.g., all Categorical

Outlook	Temperature	Humidity	Wind	PlayBall?
Sunny	Hot	High	Weak	No
Sunny	Hot	High	Strong	No
Overcast	Hot	High	Weak	Yes
Sunny	Mild	High	strong	Yes
Rain	Cool	Normal	Weak	Yes
Rain	Cool	Normal	Strong	No

Play or not: if sunny, cool, normal humidity but strong wind?

Naïve Bayes E.g., all Categorical

Pr(PlayBall = yes | sunny, cool, normal, strong)
= Pr(yes) Pr(sunny|yes) Pr(cool|yes) Pr(normal|yes)Pr(strong|yes)
= (1/2) * (1/3) * (1/3) * (1/3) * (1/3)
= 1 / 162

Pr(PlayBall = no | sunny, cool, normal, strong)
= Pr(no) Pr(sunny|no) Pr(cool|no) Pr(normal|no)Pr(strong|no)
= (1/2) * (2/3) * (1/3) * (1/3) * (2/3)
= 4 / 162

So the prediction is PlayBall = no
Odds ratio = Pr(yes|data) / Pr(no|data) = 1/4 (i.e, 4 times more likely not playing)

Naïve Bayes E.g., Auto Risk (Continuous + Categorical)

Age	CarType	Risk
23	Family	High
17	Sports	High
43	Sports	High
68	Family	Low
32	Truck	Low
20	Family	High

High or Low Risk for a new case (30,family)?

Assuming Gaussian for age

- $\mu_{hi} = 21.75$, $\sigma_{hi} = 10.9$
- $\mu_{lo} = 50$, $\sigma_{lo} = 18$

Pr(age = 30 | risk = high) = 0.027

Pr(carType=family|risk = high) = 2/4 = 0.5

Pr(age = 30 | risk = low) = 0.012

Pr(carType=family|risk = low) = 1/2 = 0.5

$$\text{Odds ratio} = (0.67*0.027*0.5) / (0.33*0.012*0.5) = 4.5$$

Towards high risk

Naïve Bayes Pros and Cons

- For more examples on continuous features, see the "gender classification" example on the wiki page for "naïve bayes classifier"
- What if we need to predict the risks for (30,truck)?
- Cons: for small training data, ML estimates can be way off, making Naïve Bayes way off
- Pros: very simple; quite good performance even if conditional independence of features is not satisfied

Concluding Remarks: Big Data

- Learning how to make probabilistic predictions using Bayes rule
- ML parameter learning is widely used in statistics as well as data mining
- Naïve Bayes classifier is simple and works well with large training datasets
- But it does not produce any "new knowledge" as in a decision tree