CS540 Machine learning
L8
Announcements

• Linear algebra tutorial by Mark Schmidt, 5:30 to 6:30 pm today, in the CS X-wing 8th floor lounge (X836).
• Move midterm from Tue Oct 14 to Thu Oct 16?
• Hw3sol handed out today
• Change in order
Last time

• Multivariate Gaussians
• Eigenanalysis
• MLE
• Use in generative classifiers
• Naïve Bayes classifiers
• Bayesian parameter estimation I: Beta-Binomial model
Bayes rule for classifiers

\[ p(y = c | x) = \frac{p(x | y = c)p(y = c)}{\sum_{c'} p(x | y = c')p(y = c')} \]

- **Class posterior**
- **Class-conditional density**
- **Class prior**
- **Normalization constant**
Class prior

- Let \((Y_1, \ldots, Y_C) \sim \text{Mult}(\pi, 1)\) be the class prior.

\[
P(y_1, \ldots, y_C | \pi) = \prod_{c=1}^{C} \pi_c^{I(y_c=1)} \quad \sum_{c=1}^{C} \pi_c = 1
\]

- Since \(\sum_c Y_c = 1\), only one bit can be on. This is called a 1-of-C encoding. We can write \(Y=c\) instead.

\[Y=2 \equiv (Y_1, Y_2, Y_3) = (0,1,0)\]

\[
P(y | \pi) = \prod_{c=1}^{C} \pi_c^{I(y=c)} = \pi_y
\]

- e.g., \(p(\text{man})=0.7, p(\text{woman})=0.1, p(\text{child})=0.2\)
Correlated features

- Height and weight are not independent
Fitting the model

- Fit each class conditional density separately

\[
\mu_c = \frac{1}{n_c} \sum_{i=1}^{n} I(y_i = c) x_i = \frac{1}{n_c} \sum_{i:y_i=c} x_i
\]

\[
\Sigma_c = \frac{1}{n_c} \sum_{i=1}^{n} I(y_i = c)(x_i - \mu_c)(x_i - \mu_c)^T
\]

\[
n_c = \sum_{i=1}^{n} I(y_i = c)
\]

\[
\pi_c = \frac{n_c}{n}
\]
Ignoring the correlation...

- If \( X_j \in R \), we can use product of 1d Gaussians

\[
X_j | y=c \sim N(\mu_{jc}, \sigma_{jc})
\]

\[
p(x|y = c) = \prod_{j=1}^{d} \frac{1}{\sqrt{2\pi\sigma_{jc}^2}} \exp\left(-\frac{1}{2\sigma_{jc}^2}(x_j - \mu_{jc})^2\right)
\]

\[
\Sigma_c = \begin{pmatrix}
\sigma_{1c}^2 & \cdots & 0 \\
\vdots & \ddots & \vdots \\
0 & \cdots & \sigma_{dc}^2
\end{pmatrix}
\]
Document classification

• Let \( Y \in \{1,\ldots,C\} \) be the class label and \( x \in \{0,1\}^d \)

• eg \( Y \in \{\text{spam, urgent, normal}\} \),

\[ x_i = I(\text{word } i \text{ is present in message}) \]

• Bag of words model

\[ \begin{array}{ccccccc}
1 & 2 & 3 & 4 & 5 & 6 & 7 \\
\end{array} \]

Words = \{john, mary, sex, money, send, meeting, unk\}

“John sent money to Mary after the meeting about money”

\[ \begin{array}{ccccccc}
1 & 7 & 4 & 2 & 7 & 6 & 7 & 4 \\
\end{array} \]

Stop word removal

“john sent money mary after meeting about money”

Tokenization

\[ [1, 1, 0, 2, 0, 1] \]

Word counting

\[ [1, 1, 0, 1, 0, 1] \]

Thresholding (binarization)
Binary features (multivariate Bernoulli)

- Let $X_i | y = c \sim \text{Ber}(\mu_{ic})$ so $p(X_i = 1 | y = c) = \mu_{ic}$

$$p(x | y = c, \mu) = \prod_{j=1}^{d} \mu_{jc}^{I(x_j = 1)} (1 - \mu_{jc})^{I(x_j = 0)}$$
Fitting the model

\[ \mu_{jc} = \frac{1}{n_c} \sum_{i=1}^{n} I(y_i = c)I(x_{ij} = 1) = \frac{n_{jc}}{n_c} \]

\[ n_{jc} = \sum_{i=1}^{n} I(y_i = c, x_{ij} = 1) \]
Class posterior

- Bayes rule
  \[ p(y = c| x) = \frac{p(y = c)p(x|y = c)}{p(x)} = \frac{\pi_c \prod_{i=1}^{d} \theta_{ic}^{I(x_i=1)}(1 - \theta_{ic})^{I(x_i=0)}}{p(x)} \]

- Since numerator and denominator are very small number, use logs to avoid underflow
  \[ \log p(y = c| x) = \log \pi_c + \sum_{i=1}^{d} I(x_i = 1) \log \theta_{ic} + I(x_i = 0) \log(1 - \theta_{ic}) - \log p(x) \]

- How compute the normalization constant?
  \[ \log p(x) = \log[\sum_c p(y = c, x)] = \log[\sum_c \pi_c f_c] \]
Log-sum-exp trick

• Define

\[
\log p(x) = \log \left[ \sum_c \pi_c f_c \right]
\]

\[
b_c = \log \pi_c + \log f_c
\]

\[
\log p(x) = \log \sum_c e^{b_c} = \log \left[ \left( \sum_c e^{b_c} \right) e^{-B} e^B \right]
\]

\[
= \log \left[ \left( \sum_c e^{b_c-B} \right) e^B \right] = \left[ \log(\sum_c e^{b_c-B}) \right] + B
\]

\[
B = \max_c b_c
\]

\[
\log(e^{-120} + e^{-121}) = \log \left( e^{-120} (e^0 + e^{-1}) \right) = \log(e^0 + e^{-1}) - 120
\]

• In Matlab, use Minka’s function \( S = \text{logsumexp}(b) \)

\[
\log \text{joint} = \log(\text{prior}) + \text{counts} \times \log(\theta) + (1-\text{counts}) \times \log(1-\theta);
\]

\[
\log \text{post} = \log \text{joint} - \log \text{sumexp}(\log \text{joint})
\]
• Suppose the value of $x_1$ is unknown
• We can simply drop the term $p(x_1 | y = c)$.

$$p(y = c | x_{2:d}) \propto p(y = c, x_{2:d})$$

$$= \sum_{x_1} p(y = c, x_1, x_{2:d})$$

$$= \sum_{x_1} p(y = c) \prod_{j=1}^{d} p(x_j | y = c)$$

$$= p(y = c) \left[ \sum_{x_1} p(x_1 | y = c) \right] \prod_{j=2}^{d} p(x_j | y = c)$$

$$= p(y = c) \prod_{j=2}^{d} p(x_j | y = c)$$

• This is a big advantage of generative classifiers over discriminative classifiers
Form of the class posterior

- We can derive an analytic expression for \( p(y=c|x) \) that will be useful later.

\[
p(Y = c | x, \theta, \pi) = \frac{p(x | y = c)p(y = c)}{\sum_{c'} p(x | y = c')p(y = c')}
\]

\[
= \frac{\exp[\log p(x | y = c) + \log p(y = c)]}{\sum_{c'} \exp[\log p(x | y = c') + \log p(y = c')]} \]

\[
= \frac{\exp [\log \pi_c + \sum_i I(x_i = 1) \log \theta_{ic} + I(x_i = 0) \log(1 - \theta_{ic})]}{\sum_{c'} \exp [\log \pi_{c'} + \sum_i I(x_i = 1) \log \theta_{i,c'} + I(x_i = 0) \log(1 - \theta_{i,c'})]} \]
Form of the class posterior

- From previous slide
  \[ p(Y = c|x, \theta, \pi) \propto \exp \left[ \log \pi_c + \sum_i I(x_i = 1) \log \theta_{ic} + I(x_i = 0) \log(1 - \theta_{ic}) \right] \]

- Define
  \[
  x' = [1, I(x_1 = 1), I(x_1 = 0), \ldots, I(x_d = 1), I(x_d = 0)] \\
  \beta_c = [\log \pi_c, \log \theta_{1c}, \log(1 - \theta_{1c}), \ldots, \log \theta_{dc}, \log(1 - \theta_{dc})] 
  \]

- Then the posterior is given by the softmax function
  \[
  p(Y = c|x, \beta) = \frac{\exp[\beta_c^T x']}{\sum_{c'} \exp[\beta_{c'}^T x']} 
  \]
Discriminative vs generative

- Discriminative: \( p(y|x, \theta) \)
- Generative: \( p(y, x| \theta) \)
Logistic regression vs naïve Bayes

<table>
<thead>
<tr>
<th>Easy to fit?</th>
<th>Discriminative</th>
<th>Generative</th>
</tr>
</thead>
<tbody>
<tr>
<td>Can handle basis function expansion?</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Fit classes separately?</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Handle missing data?</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Best for

- **Large sample size** for Discriminative
- **Small sample size** for Generative
Sparse data problem

- Consider naïve Bayes for binary features.

\[
p(x | y = S) = \left( \frac{1}{N_S} \right) \left( \frac{10}{N_S} \right) \left( \frac{0}{N_S} \right) = 0
\]

\[X = \text{“you will receive our limited time offer if you send us $1M today”}\]

<table>
<thead>
<tr>
<th></th>
<th>Spam</th>
<th>Ham</th>
</tr>
</thead>
<tbody>
<tr>
<td>Limited</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Time</td>
<td>10</td>
<td>9</td>
</tr>
<tr>
<td>Offer</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Total</td>
<td>Ns</td>
<td>Nh</td>
</tr>
</tbody>
</table>
Outline

• Bayes: what/why?
• Bernoulli
Fundamental principle of Bayesian statistics

• In Bayesian stats, everything that is uncertain (e.g., $\theta$) is modeled with a probability distribution.

• We incorporate everything that is known (e.g., D) by conditioning on it, using Bayes rule to update our prior beliefs into posterior beliefs.

\[
p(h \mid d) = \frac{p(d \mid h) p(h)}{\sum_{h' \in H} p(d \mid h') p(h')}
\]

Bayesian inference = Inverse probability theory
In praise of Bayes

• Bayesian methods are conceptually simple and elegant, and can handle small sample sizes (e.g., one-shot learning) and complex hierarchical models without overfitting.

• They provide a single mechanism for answering all questions of interest; there is no need to choose between different estimators, hypothesis testing procedures, etc.

• They avoid various pathologies associated with orthodox statistics.

• They often enjoy good frequentist properties.
Why isn’t everyone a Bayesian?

- The need for a prior.
- Computational issues.
The need for a prior

- Bayes rule requires a prior, which is considered “subjective”.
- However, we know learning without assumptions is impossible (no free lunch theorem).
- Often we actually have informative prior knowledge.
- If not, it is possible to create relatively “uninformative” priors to represent prior ignorance.
- We can also estimate our priors from data (empirical Bayes).
- We can use posterior predictive checks to test goodness of fit of both prior and likelihood.
• Computing the normalization constant requires integrating over all the parameters

\[ p(\theta|D) = \frac{p(\theta)p(D|\theta)}{\int p(\theta')p(D|\theta')d\theta'} \]

• Computing posterior expectations requires integrating over all the parameters

\[ Ef(\Theta) = \int f(\theta)p(\theta|D)d\theta \]
Approximate inference

- We can evaluate posterior expectations using Monte Carlo integration

\[
E f(\Theta) = \int f(\theta)p(\theta|D)d\theta \approx \frac{1}{N} \sum_{s=1}^{N} f(\theta^s) \quad \text{where } \theta^s \sim p(\theta|D)
\]

- Generating posterior samples can be tricky
  - Importance sampling
  - Particle filtering
  - Markov chain Monte Carlo (MCMC)

- There are also deterministic approximation methods
  - Laplace
  - Variational Bayes
  - Expectation Propagation
Conjugate priors

- For simplicity, we will mostly focus on a special kind of prior which has nice mathematical properties.
- A prior \( p(\theta) \) is said to be *conjugate* to a likelihood \( p(D|\theta) \) if the corresponding posterior \( p(\theta|D) \) has the same functional form as \( p(\theta) \).
- This means the prior family is *closed under Bayesian updating*.
- So we can recursively apply the rule to update our beliefs as data streams in (online learning).
- A natural conjugate prior means \( p(\theta) \) has the same functional form as \( p(D|\theta) \).
Consider the problem of estimating the probability of heads $\theta$ from a sequence of $N$ coin tosses, $D = (X_1, ..., X_N)$

First we define the likelihood function, then the prior, then compute the posterior. We will also consider different ways to predict the future.

MLE is

$$\hat{\theta} = \frac{N_1}{N}$$

Suffers from sparse data problem
Black swan paradox

- Suppose we have seen $N=3$ white swans. What is the probability that swan $X_{N+1}$ is black?
- If we plug in the MLE, we predict black swans are impossible, since $N_b=N_1=0$, $N_w=N_0=3$
  \[
  \hat{\theta}_{MLE} = \frac{N_b}{N_b + N_w} = \frac{0}{N}, \quad p(X=b|\hat{\theta}_{MLE}) = \hat{\theta}_{MLE} = 0
  \]
- However, this may just be due to sparse data.
- Below, we will see how Bayesian approaches work better in the small sample setting.
The beta-Bernoulli model

- Consider the probability of heads, given a sequence of \( N \) coin tosses, \( X_1, \ldots, X_N \).
- Likelihood

\[
p(D|\theta) = \prod_{n=1}^{N} \theta^{X_n} (1 - \theta)^{1-X_n} = \theta^{N_1} (1 - \theta)^{N_0}
\]

- Natural conjugate prior is the Beta distribution

\[
p(\theta) = Be(\theta|\alpha_1, \alpha_0) \propto \theta^{\alpha_1-1} (1 - \theta)^{\alpha_0-1}
\]

- Posterior is also Beta, with updated counts

\[
p(\theta|D) = Be(\theta|\alpha_1 + N_1, \alpha_0 + N_0) \propto \theta^{\alpha_1-1+N_1} (1 - \theta)^{\alpha_0-1+N_0}
\]

Just combine the exponents in \( \theta \) and \( (1-\theta) \) from the prior and likelihood
The beta distribution

- Beta distribution
  \[ p(\theta | \alpha_1, \alpha_0) = \frac{1}{B(\alpha_1, \alpha_0)} \theta^{\alpha_1-1} (1 - \theta)^{\alpha_0-1} \]
- The normalization constant is the beta function
  \[ B(\alpha_1, \alpha_0) = \int_0^1 \theta^{\alpha_1-1} (1 - \theta)^{\alpha_0-1} d\theta = \frac{\Gamma(\alpha_1)\Gamma(\alpha_0)}{\Gamma(\alpha_1 + \alpha_0)} \]

\[ E[\theta] = \frac{\alpha_1}{\alpha_1 + \alpha_0} \]
Updating a beta distribution

• Prior is Beta(2,2). Observe 1 head. Posterior is Beta(3,2), so mean shifts from 2/4 to 3/5.

• Prior is Beta(3,2). Observe 1 head. Posterior is Beta(4,2), so mean shifts from 3/5 to 4/6.
Setting the hyper-parameters

- The prior hyper-parameters $\alpha_1$, $\alpha_0$ can be interpreted as pseudo counts.
- The effective sample size (strength) of the prior is $\alpha_1 + \alpha_0$.
- The prior mean is $\frac{\alpha_1}{\alpha_1 + \alpha_0}$.
- If our prior belief is $p(\text{heads}) = 0.3$, and we think this belief is equivalent to about 10 data points, we just solve

\[
\alpha_1 + \alpha_0 = 10, \quad \frac{\alpha_1}{\alpha_1 + \alpha_0} = 0.3
\]
Posterior mean

- Let $N = N_1 + N_0$ be the amount of data, and $M = \alpha_0 + \alpha_1$ be the amount of virtual data.

The posterior mean is a convex combination of prior mean $\frac{\alpha_1}{M}$ and MLE $\frac{N_1}{N}$

\[
E[\theta|\alpha_1, \alpha_0, N_1, N_0] = \frac{\alpha_1 + N_1}{\alpha_1 + N_1 + \alpha_0 + N_0} = \frac{\alpha_1 + N_1}{N + M}
\]

\[
= \frac{M}{N + M} \frac{\alpha_1}{M} + \frac{N}{N + M} \frac{N_1}{N}
\]

\[
= w \frac{\alpha_1}{M} + (1 - w) \frac{N_1}{N}
\]

$w = \frac{M}{N + M}$ is the strength of the prior relative to the total amount of data

We shrink our estimate away from the MLE towards the prior (a form of regularization).
MAP estimation

- It is often easier to compute the posterior mode (optimization) than the posterior mean (integration).
- This is called maximum a posteriori estimation.
  \[ \hat{\theta}_{MAP} = \arg \max_{\theta} p(\theta|D) \]
- This is equivalent to penalized likelihood estimation.
  \[ \hat{\theta}_{MAP} = \arg \max_{\theta} \log p(D|\theta) + \log p(\theta) \]
- For the beta distribution,
  \[ MAP = \frac{\alpha_1 - 1}{\alpha_1 + \alpha_0 - 2} \]
Posterior predictive distribution

- We integrate out our uncertainty about $\theta$ when predicting the future (hedge our bets)

$$p(X|D) = \int p(X|\theta)p(\theta|D)d\theta$$

- If the posterior becomes peaked

$$p(\theta|D) \to \delta(\theta - \hat{\theta})$$

we get the *plug-in principle*.

$$p(x|D) = \int p(x|\theta)\delta(\theta - \hat{\theta})d\theta = p(x|\hat{\theta})$$

Sifting property of delta functions
Posterior predictive distribution

- Let $\alpha_i' = \text{updated hyper-parameters.}$
- In this case, the posterior predictive is equivalent to plugging in the posterior mean parameters

$$p(X = 1|D) = \int_0^1 p(X = 1|\theta)p(\theta|D)d\theta$$

$$= \int_0^1 \theta \text{Beta}(\theta|\alpha_1', \alpha_0')d\theta = E[\theta] = \frac{\alpha_1'}{\alpha_0' + \alpha_1'}$$

- If $\alpha_0 = \alpha_1 = 1$, we get Laplace’s rule of succession (add one smoothing)

$$p(X = 1|N_1, N_0) = \frac{N_1 + 1}{N_1 + N_0 + 2}$$
Solution to black swan paradox

- If we use a Beta(1,1) prior, the posterior predictive is
  \[ p(X = 1|N_1, N_0) = \frac{N_1 + 1}{N_1 + N_0 + 2} \]
  so we will never predict black swans are impossible.

- However, as we see more and more white swans, we will come to believe that black swans are pretty rare.
Summary of beta-Bernoulli model

- **Prior** \( p(\theta) = \text{Beta}(\theta|\alpha_1, \alpha_0) = \frac{1}{B(\alpha_1, \alpha_0)} \theta^{\alpha_1-1} (1 - \theta)^{\alpha_0-1} \)

- **Likelihood** \( p(D|\theta) = \theta^{N_1} (1 - \theta)^{N_0} \)

- **Posterior** \( p(\theta|D) = \text{Beta}(\theta|\alpha_1 + N_1, \alpha_0 + N_0) \)

- **Posterior predictive** \( p(X = 1|D) = \frac{\alpha_1 + N_1}{\alpha_1 + \alpha_0 + N} \)