CS 281: Advanced Machine Learning

taught by Sasha Rush

Fall 2017

Contents

CS281: Advanced ML **September 6, 2017**

Lecture 1: Discrete Models

Lecturer: Sasha Rush *Scribes: Anna Sophie Hilgard, Diondra Peck*

- Discrete models take values from a countable set, e.g. {0,1}, {cold, flu, asthma} and are simpler than continuous models.
- We will use simple discrete models to develop our tactics such as marginalization and conditioning.
- Today, we will focus coins as a real-world example.

1.1 Bernoulli model

The likelihood is of the form $p(\text{heads}) = \theta$.

Easy Prior

Assume we know the coin came from one of 3 unknown manufacturers (later, we'll have mixture model estimation, but for now assume these probabilities come from an oracle).

1. $\theta = 0.4$ with probability .1
2. $\theta = 0.5$ with probability .8
3. $\theta = 0.6$ with probability .1

$$p(\theta) = 0.1 \cdot \delta(\theta = 0.4) + 0.8 \cdot \delta(\theta = 0.5) + 0.1 \cdot \delta(\theta = 0.6)$$

Likelihood

Likelihood = $p(\text{data}|\text{parameters})$. For the coin example,

$$p(\text{coin flips}|\theta) = \text{Bin}(N_1|N, \theta) = \binom{N}{N_1}\theta^{N_1}(1-\theta)^{N-N_1} \quad \text{where } N = N_0 + N_1 = \text{number of flips}$$

Note that the last two terms, the "score", is our focus since they are the only terms that depend on θ. The first term normalizes the distribution.

1.2 Inference

Inference 1: $p(\theta|x)$ ($x \in N_0, N_1$). How can we estimate θ?

Maximum Likelihood Estimation (MLE)

$$\theta_{MLE} = \text{argmax}_\theta\ p(N_0, N_1|\theta) = \text{argmax}_\theta \log\left[p(N_0, N_1|\theta)\right]$$

$$\theta_{MLE} = \text{argmax}_\theta \log\binom{N}{N_1} + N_1 \text{log}\theta + N_0\text{log}(1-\theta) \quad \text{Because the first term is not a function of } \theta\text{, we can ignore it.}$$

$$\frac{d}{d\theta} = \frac{N_1}{\theta} + \frac{N_0}{1-\theta}\cdot(-1) \rightarrow \theta_{MLE} = \frac{N_1}{N_0 + N_1}$$

Note that Inference $\neq$ Decision Making. If we asked you to make a bet on the coin, based on this you could either

1. Always take heads if $\theta > .5$. In this case, $p(\text{win}) = \theta$
2. Take heads with probability = θ. In this case, $p(\text{win}) = \theta^2 + (1-\theta)^2$ [p(is heads) * p(choose heads) + ...]

If θ = 0.6, for option 1, p(win) = θ = 0.6. For option 2, p(win) = $\theta^2 + (1-\theta)$ = 0.52. In this case, the additional information used in the calculation of option 2 does not result in a better decision.

Maximizing the Posterior (MAP)

Bayes Rule : $p(\theta|\text{data}) \propto p(\text{data}|\theta)p(\theta)$

- Posterior: $p(\theta|\text{x})$
- Likelihood: $p(\text{x}|\theta)$
- Prior: $p(\theta)$

$$\theta_{MAP} = \text{argmax}_\theta\ p(\theta|\text{x}) = \text{argmax}_\theta \log\left[p(\text{x}|\theta)p(\theta)\right] \quad \text{from Bayes' Rule: } p(\theta|\text{x}) \propto p(\text{x}|\theta)p(\theta)$$

Consider an example:

$$p(\theta = 0.4|N_0, N_1) \propto \binom{N}{N_1}(.4)^{N_1}(1-.4)^{N_0}(0.1)$$
$$p(\theta = 0.45|N_0, N_1) = 0 \quad \text{Due to the sparsity of the prior - similar result for } \theta \text{ = 0.5 and 0.6}$$

$\theta_{MAP} = \theta_{MLE}$ when we have a uniform prior since the MLE calculation does not explicitly factor a prior into its calculation.

Full Posterior

Partition or Marginal Likelihood: $p(N_0, N_1) = \int_\theta p(N_0, N_1, \theta)$.

$$p(\theta|N_0, N_1) = \frac{p(x|\theta)p(\theta)}{p(N_0, N_1)} \qquad \text{Note that } p(N_0, N_1) \text{ is a very difficult term to compute.}$$

Beta Prior

$$p(\theta|\alpha_0, \alpha_1) = \frac{\Gamma(\alpha_0 + \alpha_1)}{\Gamma(\alpha_0)\Gamma(\alpha_1)}\theta^{\alpha_1 - 1}(1-\theta)^{\alpha_0 - 1} \qquad \text{support} \in [0, 1]$$

From the image of the beta function for different parameters, we can see that it can ether be balanced, skewed to one side, or tend toward infinity on one side.
With a beta prior:

$$p(\theta|N_0, N_1) = \frac{\Gamma(\alpha_0 + \alpha_1)}{\Gamma(\alpha_0)\Gamma(\alpha_1)}\theta^{\alpha_1 - 1}(1-\theta)^{\alpha_0 - 1}\theta^{N_1}(1-\theta)^{N_0} \cdot (\text{constant normalization term w.r.t } \theta)$$

The key insight is that we get additive terms in the exponent and the resulting distribution looks like another beta. The prior "counts" (pseudocounts) from the hyperparameters can be interpreted as counts we have beforehand.

$$p(\theta|N_0, N_1) = \frac{\Gamma(\alpha_0 + \alpha_1)}{\Gamma(\alpha_0)\Gamma(\alpha_1)}\theta^{N_1 + \alpha_1 - 1}(1-\theta)^{N_0 + \alpha_0 - 1} \cdot (\text{constant normalization term w.r.t } \theta)$$

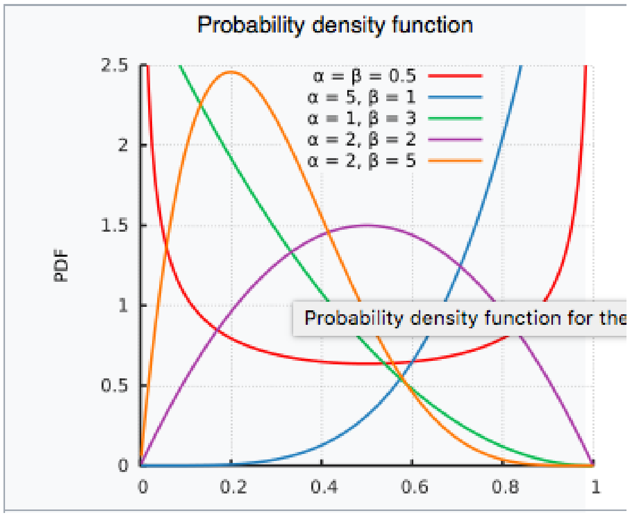


Figure 1.1: Beta Params

To make this distribution sum to 1, use the known beta normalizer

$$p(\theta|N_0, N_1) = \frac{\Gamma(\alpha_0 + \alpha_1 + N_0 + N_1)}{\Gamma(\alpha_0)\Gamma(\alpha_1)\Gamma(N_0)\Gamma(N_1)} \theta^{N_1+\alpha_1-1}(1-\theta)^{N_0+\alpha_0-1} \sim \text{Beta}(\theta|N_0 + \alpha_0, N_1 + \alpha_1) \qquad \text{(posterior)}$$

The mode of the Beta gives us back θ_{MAP}, but with additional information about the shape of the distribution. What does the prior that tends to infinity at 1 imply? That in the absence of other information, the coin is definitely heads.

Predictive Distribution

$$\begin{aligned} p(\hat{x}|N_0, N_1) &= \int_\theta p(x|\theta, N_0, N_1)p(\theta|N_0, N_1)d\theta \\ &= \int_\theta \theta p(\theta|N_0, N_1)d\theta \\ &= \mathbb{E}_{\theta \sim p(\theta|N_0, N_1)}\theta \end{aligned}$$

This is the expectation under the posterior of θ which is the mean of the Beta distribution. Feel free to prove this as an exercise.

Marginal Likelihood

$$\begin{aligned} p(N_0, N_1) &= \int_\theta p(x_1, \dots x_n|\theta)p(\theta)d\theta \\ &= \int_\theta \frac{\Gamma(\alpha_1 + \alpha_1)}{\Gamma(\alpha_0)\Gamma(\alpha_1)} \theta^{\alpha_1+N_1-1}(1-\theta)^{\alpha_0+N_0-1} \end{aligned}$$

The first term can be moved outside, as it does not depend on θ. After introducing our normalization term and making the distribution insidesum to 1,

$$p(N_0, N_1) = \frac{\Gamma(\alpha_1 + \alpha_1)}{\Gamma(\alpha_0)\Gamma(\alpha_1)} \frac{\Gamma(N_0 + \alpha_0)\Gamma(N_1 + \alpha_1)}{\Gamma(N_0 + N_1 + \alpha_0 + \alpha_1)}$$

1.3 Extensions on the Coin Flip Model: Super Coins

- Many correlated coins: models of binary data, important for discrete graphical models
- Many-sided coins aka dice: models of categorical data, generalization of Bernoulli

1.4 Other Distributions

$$\begin{aligned}
\text{Bernoulli}(x|\theta) &= \theta^x(1-\theta)^x \\
\text{Categorical}(x|\theta) &= \prod_k \theta_k^{x_k} && \text{generalization of Bernoulli} \\
\text{Multinomial}(x|\theta) &= \frac{(\sum x_k)!}{\prod_k x_k!}\prod_k \theta_k^{x_k} && \text{generalization of Binomial} \\
\text{Dirichlet}(x|\alpha) &= \frac{\Gamma(\sum \alpha_k)}{\prod \Gamma(\alpha_k)}\prod_k \theta_k^{\alpha_k - 1} && \text{generalization of Beta, often used as a prior}
\end{aligned}$$

Note that the Dirichlet distribution is the conjugate prior of the Categorical and Multinomial distributions.

1.5 Example notebook

See Beta.ipynb

CS281: Advanced ML **September 11, 2017**

Lecture 3: Multivariate Normal Distributions

Lecturer: Sasha Rush *Scribes: Christopher Mosch, Lindsey Brown, Ryan Lapcevic*

3.6 Examples

Multivariate gaussians are used for modeling in various applications, where knowing mean and variance is useful:

- radar: mean and variance of approaching objects (like invading aliens)
- weather forecasting: predicting the position of a hurricane, where the uncertainty in the storm's position increases for timepoints farther away
- tracking the likely outcome of a sports game: last year's superbowl is an example of a failure of modeling with multivariate gaussians as the Patriots still won after a large Falcons' lead

3.7 Review: Eigendecomposition

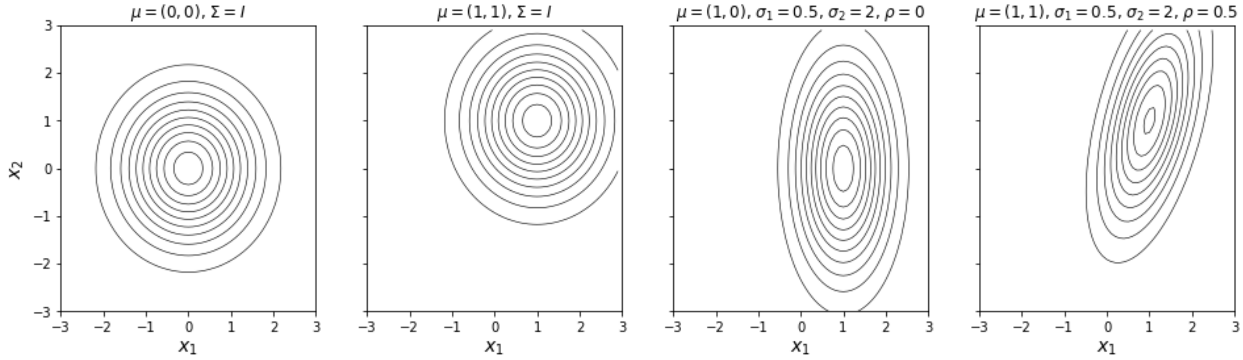
Let $\mathbf{\Sigma}$ be a square, symmetric matrix. Then its eigendecomposition is given by $\mathbf{\Sigma} = \boldsymbol{U}^T \mathbf{\Lambda} \boldsymbol{U}$, where $\boldsymbol{U}$ is an orthogonal matrix and $\mathbf{\Lambda}$ is a diagonal matrix. In the special case that $\mathbf{\Sigma}$ is positive semidefinite (as is the case for covariance matrices), denoted $\mathbf{\Sigma} \succeq 0$, all its eigenvalues are nonnegative, $\Lambda_{ii} \geq 0$, and we can decompose its inverse as $\mathbf{\Sigma}^{-1} = \boldsymbol{U}^T \mathbf{\Lambda}^{-1} \boldsymbol{U}$, where $\Lambda_{ii}^{-1} = 1/\Lambda_{ii}$.

3.8 Multivariate Normal Distributions (MVNs)

Let X be a D-dimensional MVN random vector with mean $\boldsymbol{\mu}$ and covariance matrix $\mathbf{\Sigma}$, denoted $X \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{\Sigma})$. Then the pdf of X is

$$p(x) = (2\pi)^{-D/2} \, |\mathbf{\Sigma}|^{-1/2} \exp{[-\frac{1}{2}(x-\boldsymbol{\mu})^T \mathbf{\Sigma}^{-1}(x-\boldsymbol{\mu})]},$$

where for many problems we focus on the quadratic form $(x-\boldsymbol{\mu})^T \mathbf{\Sigma}^{-1}(x-\boldsymbol{\mu})$ (which geometrically can be thought of as distance) and ignore the normalization factor $(2\pi)^{-D/2} \, |\mathbf{\Sigma}|^{-1/2}$. The figure below plots the contours of a bivariate Normal for various $\boldsymbol{\mu}$ and $\mathbf{\Sigma}$ (in the figure, ρ denotes the off-diagonal elements of $\mathbf{\Sigma}$, given by the covariance of x_1 and x_2).



$\mu = (0, 0),\ \Sigma = I$

$\mu = (1, 1),\ \Sigma = I$

$\mu = (1, 0),\ \sigma_1 = 0.5,\ \sigma_2 = 2,\ \rho = 0$

$\mu = (1, 1),\ \sigma_1 = 0.5,\ \sigma_2 = 2,\ \rho = 0.5$

Note that we can decompose $\boldsymbol{\Sigma}$ as

$$\begin{aligned}
\boldsymbol{\Sigma} &= (x-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(x-\boldsymbol{\mu}) \\
&= (x-\boldsymbol{\mu})^T\left(\boldsymbol{U}^T\boldsymbol{\Lambda}^{-1}\boldsymbol{U}\right)(x-\boldsymbol{\mu}) \\
&= (x-\boldsymbol{\mu})^T\left(\sum_d \frac{1}{\lambda_d}U_d U^T\right)(x-\boldsymbol{\mu}) \\
&= \sum_d \frac{1}{\lambda_d}(x-\boldsymbol{\mu})^T U_d U_d^T(x-\boldsymbol{\mu}),
\end{aligned}$$

where $(x-\boldsymbol{\mu})^T U_d$ can be interpreted as the projection of $(x-\boldsymbol{\mu})$ onto U_d (which can each be thought of as univariate gaussians), the eigenvector corresponding to the eigenvalue λ_d. Since $\boldsymbol{\Sigma}$ is the weighted sum of the dot product of such projections (with weights being given by $1/\lambda_d$, which can be thought of as the scale $1/\sigma^2$), we can describe the MVN as tiling of univariates.

3.8.1 Manipulating MVNs: Stretches, Rotations, and Shifts

Let $x \sim \mathcal{N}(0, \boldsymbol{I})$ and $y = \boldsymbol{A}x + \boldsymbol{b}$. We want to consider two ways of obtaining the complete distribution of y.

- 'Overkill': We can perform a change of variables[1]. Here, we have $x = \boldsymbol{A}^{-1}$ and $|dx/dy| = |A^{-1}|$, leading to

$$\begin{aligned}
p(y) &= \mathcal{N}\left(\boldsymbol{A}^{-1}(y-\boldsymbol{b})|0,\boldsymbol{I}\right)|\boldsymbol{A}^{-1}| \\
&= \frac{1}{z}\exp\left[(\boldsymbol{A}^{-1}(y-\boldsymbol{b}))^T(\boldsymbol{A}^{-1}(y-\boldsymbol{b}))\right] \\
&= \frac{1}{z}\exp\left[(y-\boldsymbol{b})^T(\boldsymbol{A}^{-1})^T(\boldsymbol{A}^{-1})(y-\boldsymbol{b})\right] \\
&= \mathcal{N}(y|\boldsymbol{b}, \boldsymbol{A}\boldsymbol{A}^T),
\end{aligned}$$

  where z is the normalizing constant.

- Using the properties of MVN, we know that y is also MVN, so is completely specified by its mean and covariance matrix which can easily be derived,

$$\mathbb{E}(y) = \mathbb{E}(\boldsymbol{A}x+\boldsymbol{b}) = \boldsymbol{A}\mathbb{E}(x)+\boldsymbol{b} \qquad \mathrm{cov}(y) = \boldsymbol{A}\boldsymbol{A}^T.$$

Thus, we can generate MVN from $\mathcal{N}(0,\boldsymbol{I})$ via the transformation $y = \boldsymbol{A}x+\boldsymbol{b}$, where we set $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Lambda}^{1/2}$, leading to $\boldsymbol{\Sigma}_Y = \boldsymbol{U}^T\boldsymbol{\Lambda}\boldsymbol{U}$. Then shifts are represented by $\boldsymbol{b}$, stretches by $\boldsymbol{\Lambda}$, and rotations by $\boldsymbol{U}$.

3.8.2 Detour: MVN in High-Dimensions ($D \gg 0$)

Let x be a D-dimensional random vector, distributed as $\mathcal{N}(0, I/D)$, where I is the identity. The expected length of x is given by

$$\mathbb{E}\left(\|x\|^2\right) = \mathbb{E}\left(\sum_d x_d^2\right) = D\sigma_d^2 = 1,$$

which means that x is expected to be on the boundary of a unit sphere centered at the origin. Moreover, the variance of the length is

$$\mathrm{var}\left(\|x\|^2\right) = D\cdot\left(\mathbb{E}(x^4) - \mathbb{E}\left(x^2\right)^2\right) = D\cdot(3\sigma^4-\sigma^4) = 2D/D^2 = 2/D$$

[1] A change of variables can be done in the following way: Let $y = f(x)$ and assume f is invertible so that $x = f^{-1}(y)$. Then $p(y) = p(x)|dx/dy|$. This is a technique which will be used often in this course

Thus, it is not only expected that x lies on the boundary but as D increases most of its realizations will in fact fall on the boundary[2].

3.8.3 Key Formulas for MVN: Marginalization and Conditioning

Let $X \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ with

$$X = \begin{pmatrix} X_1 \\ X_2 \end{pmatrix}, \quad \boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{pmatrix}, \quad \boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{pmatrix}.$$

Note that Σ is written in block matrix form, rather than scalar entries. It turns out that the marginals, X_1 and X_2, are also MVN, and their mean and covariance matrice are given by $\boldsymbol{\mu}_1$ and $\boldsymbol{\Sigma}_{11}$ and $\boldsymbol{\mu}_2$ and $\boldsymbol{\Sigma}_{22}$ respectively. A sketch of the proof is provided below.

$$p(x_1) = \int_{x_2} N(x|\boldsymbol{\mu}, \boldsymbol{\Sigma}) dx_2,$$

which can be written as

$$0.5 \int_{x_2} \exp \Big[(x_1 - \boldsymbol{\mu}_1)^T \boldsymbol{\Sigma}_{11}^{-1} (x_1 - \mu_1) + 2(x_1 - \boldsymbol{\mu}_1)^T \boldsymbol{\Sigma}_{12}^{-1} (x_2 - \mu_2) + (x_2 - \boldsymbol{\mu}_2)^T \boldsymbol{\Sigma}_{22}^{-1} (x_2 - \mu_2) \Big] dx_2.$$

Note that this equals

$$p(x_1) \int_{x_2} p(x_2|x_1) dx_2,$$

implying that $X_1 \sim \mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_{11})$.

While the marginals have a simple form, the conditionals are more complicated. (For a complete derivation, which requires matrix inversion lemmas, refer to Murphy.) It can be shown that $X_1|X_2 \sim \boldsymbol{\mu}_{\infty|\in}, \boldsymbol{\Sigma}_{\infty|\in}$ with

$$\boldsymbol{\mu}_{1|2} = \boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_{12} \boldsymbol{\Sigma}_{22}^{-1} (x_2 - \boldsymbol{\mu}_2), \quad \boldsymbol{\Sigma}_{1|2} = \boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12} \boldsymbol{\Sigma}_{22}^{-1} \boldsymbol{\Sigma}_{21}.$$

3.8.4 Information Form

An alternative formulation, called information form, uses the precision matrix (inverse variance) $\Lambda = \Sigma^{-1}$. Partitioning Λ as

$$\boldsymbol{\Lambda} = \begin{pmatrix} \boldsymbol{\Lambda}_{11} & \boldsymbol{\Lambda}_{12} \\ \boldsymbol{\Lambda}_{21} & \boldsymbol{\Lambda}_{22} \end{pmatrix},$$

the covariance matrices of the conditional distributions have a simple form. For example, the covariance matrix of X_1 given X_2 is $\Lambda_{1|2} = \Lambda_{11}$. However, the simplicity of the conditional precision comes at the cost of marginalization (which was simple when using Σ) becoming a more complicated expression (see Murphy subsection 4.3 for more details).

[2] It is left as an exercise to show that this formula holds. Hint: Use the fact that we assumed no covariance.

CS281: Advanced ML **September 13, 2017**

Lecture 4: Linear Regression

Lecturer: Sasha Rush *Scribes: Kojin Oshiba, Michael Ge, Aditya Prasad*

4.9 Multivariate Normal (MVN)

The multivariate normal distribution of a D-dimensional random vector X is defined as:

$$N(X|\mu,\Sigma) \sim (2\pi)^{-\frac{D}{2}}|\Sigma|^{-\frac{1}{2}}\exp\left(-\frac{1}{2}(X-\mu)^T\Sigma^{-1}(X-\mu)\right)$$

Note:

- $(2\pi)^{-\frac{D}{2}}|\Sigma|^{-\frac{1}{2}}$ and $-\frac{1}{2}$ are constants we can ignore in MLE and MAP calculations.
- $(X-\mu)^T\Sigma^{-1}(X-\mu)$ is a quadratic term.

There are three types of inference we're interested in doing: MLE, MAP, and prediction.

4.10 Maximum Likelihood of MVN

Let $\theta = (\mu, \Sigma)$, where Σ can be approximated as a diagonal/low rank matrix. If there are $x_1, \dots, x_n$ observations, the MLE estimate of μ is

$$\begin{aligned}\mu^* &= \arg\max_{\mu} - \sum_n logN(x_n|\mu,\Sigma)\\ &= \arg\max_{\mu} \quad log(constant) - \sum_n (x_n-\mu)^T\Sigma^{-1}(x_n-\mu)\\ &= \arg\max_{\mu} - \sum_n (x_n-\mu)^T\Sigma^{-1}(x_n-\mu)\end{aligned}$$

Let $L = \sum_n (x_n-\mu)^T\Sigma^{-1}(x_n-\mu)$.

$$\frac{dL}{d\mu} = \Sigma_n\Sigma^{-1}(x_n-\mu) = 0$$

$$\Leftrightarrow \mu^*_{MLE} = \frac{\Sigma_{X_n}}{N}$$

Similarly,

$$\frac{dL}{d\Sigma} = (exercise) = 0$$

$$\Leftrightarrow \Sigma^*_{MLE} = \frac{1}{N}\sum_n x_n x_n^T = \frac{1}{N}X^\top X$$

For calculating $\frac{dL}{d\Sigma}$ as an exercise , the following might be helpful:

- $\frac{d}{dA}ln|A| = A^{-1}$
- $\frac{d}{dA}tr(BA) = B^T$
- $tr(ABC) = tr(BCA)$

4.11 Linear-Gaussian Models

Let x be a vector of affine, noisy observations with a prior distribution:

$$x \sim N(m_0, S_0)$$

Let y be the outputs:

$$y|x \sim N(Ax + b, \Sigma_y)$$

4.11.1 $p(x|y)$

We are interested in calculating the posterior distribution: $p(x|y)$.

$$\begin{aligned} p(x|y) &\propto p(x)p(y|x) \\ &= \frac{1}{2}\exp\begin{cases} (x-m_0)^\top S_0^{-1}(x-m_0) \\ +(y-(Ax+b))^\top \Sigma_y^{-1}(y-(Ax+b)) \end{cases} \\ &= \frac{1}{2}\exp\begin{cases} \underline{x^\top S_0^{-1} x}^{\star\star} - \underline{2x^\top S_0^{-1} m_0}^{\star} + \dots \\ +\underline{x^\top (A^\top \Sigma_y^{-1} A) x}^{\star\star} - \underline{2x^\top (A^\top \Sigma_y^{-1}) y}^{\star} + \underline{2x^\top (A^\top \Sigma_y^{-1}) b}^{\star} + \dots \end{cases} \end{aligned}$$

The terms containing x are underlined. Double-starred ($\star\star$) terms are quadratic in x, while single-starred ($\star$) terms are linear in x. The remaining terms are constants that are swallowed up by the proportionality. By Gaussian-Gaussian conjugacy, we know the resulting distribution should be Gaussian. To find the parameters, we'll modify $p(x|y)$ to fit the form of a Normal. This requires completing the square!

4.11.2 Completing the Square

$$ax^2 + bx + c \to a(x-h)^2 + k, h = \frac{-b}{2a}, k = c - \frac{b^2}{4a}$$

We ignore the k term since it too is swallowed up in the proportionality. In application to our problem, we group the quadratic and linear terms together to calculate our terms for completing the square.

- "a" is $S_N^{-1} = S_0^{-1} + A^\top \Sigma_y^{-1} A$
- "h" is $m_N = S_N \left[S_0^{-1} m_0 + A^\top \Sigma_y^{-1}(y-b) \right]$

In this more "intuitive" representation, we find that $p(x|y)$ has the form of $N(m_N, S_N)$. Murphy also has a more explicit representation:

- $\Sigma_{x|y} = \Sigma_x^{-1} + A^\top \Sigma_y^{-1} A$
- $\mu_{x|y} = \Sigma_{x|y}[\Sigma_x^{-1}\mu_x + A^\top \Sigma_x^{-1}(y-b)]$

4.11.3 p(y)

We now calculate the normalizer term, $p(y)$. Now, x is fixed. y follows the linear model:

$$y = Ax + b + \epsilon$$

The result is that y follows a Normal distribution with the following form:

$$p(y) = N(y|Am_0 + b, \Sigma_y + A\Sigma_x A^\top)$$

4.11.4 Prior (just for μ)

$$p(\mu) = N(\mu|m_0, S_0)$$

where m_0, S_0 are pseudo mean, pseudo variance. $p(\mu)$ is defined Gaussian because Gaussian is the conjugate prior of itself. A prior is called a conjugate prior if it has the same distribution as the posterior distribution.

4.11.5 Posterior (just for μ)

$$p(\mu|X) \propto p(\mu)p(X|\mu) = N(\mu|m_0, s_0)N(X|\mu, \Sigma)$$

This is a special case of linear regression. Recall,

- "a" is $S_N^{-1} = S_0^{-1} + A^\top \Sigma_y^{-1} A$
- "h" is $m_N = S_N \left[S_0^{-1} m_0 + A^\top \Sigma_y^{-1}(y - b) \right]$

We let $b = 0$ and $A = I$. Then we obtain,

$$S_N^{-1} = S_0^{-1} + \Sigma^{-1}$$

$$m_N = S_N[S_0^{-1} m_0 + \Sigma^{-1} X]$$

Hence,

$$p(\mu|X) = N(m|m_N, S_N)$$

4.11.6 Unknown Variance

Similar to μ, we can also define a conjugate prior on Σ, which is Inverse Wishart distribution. It is defined as:

$$IW(\Sigma|S, \nu) = \frac{1}{2}|\Sigma|^{-(\nu-(D+1)/2)} exp\{\frac{1}{2} tr(S^{-1}\Sigma^{-1})\}$$

- distribution over positive semi definite Σ with two parameters S, ν.
- pseudo info $S = \Sigma X X^T$ is a psuedo scatter matrix called the scale matrix. $\nu = n\mu$ is degrees of freedom where $\nu - (D+1)$ is the number of observations.

4.12 Linear Regression

In an undergraduate version of the class, we might define the problem as follows: We are given "fixed" set of inputs, $\{\boldsymbol{x}_i\}$. We want to "predict" the outputs.

Here, we define the problem as attempting to compute $p(y \mid \boldsymbol{x}, \boldsymbol{\theta})$. Consider the following example. We assume that our data is generated as follows:

$$y = \boldsymbol{w}^T \boldsymbol{x} + \text{noise}$$

Further, we assume that the noise (denoted by ϵ) is distributed as Gaussian with mean 0; that is:

$$\epsilon \sim \mathcal{N}(0, \sigma^2)$$

Then, we have:

$$p(y \mid \boldsymbol{x}, \theta) = \mathcal{N}(y \mid \boldsymbol{w}^{\boldsymbol{T}} \boldsymbol{x}, \sigma^2)$$

Note that the bias term here is included as a dimension in $\boldsymbol{w}$, w_0.

4.12.1 Log Likelihood

Consider a data-set that looks like $\{(\boldsymbol{x_i}, y_i)\}_{i=1}^N$. The log-likelihood $\mathcal{L}(\boldsymbol{\theta})$ is given by:

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}) &= \log p(\text{data} \mid \boldsymbol{\theta}) \\ &= \sum_{n=1}^{N} \log p(y_n \mid \boldsymbol{x_n}, \boldsymbol{\theta}) \\ &= \sum_{n=1}^{N} \log(\text{constant}) - \frac{1}{2\sigma^2}(y_n - \boldsymbol{w}^T \boldsymbol{x_n})^2\end{aligned}$$

Note that data here refers to just the y_i's. The y_n's are called the target; the $\boldsymbol{w}$ represents the weights; and the x_n's are the observations. The term $(y_n - \boldsymbol{w}^T \boldsymbol{x_n})^2$ is essentially just the residual sum of squares.

4.12.2 Computing MLE

We want the argmax of the log-likelihood. We therefore have:

$$\begin{aligned}\text{argmax}_{w}\mathcal{L}(\boldsymbol{w}) &= \text{argmax} - \sum_{n=1}^{N} \frac{1}{2\sigma^2}(y_n - \boldsymbol{w}^T \boldsymbol{x_n})^2 \\ &= \text{argmax}_{w} - [\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}]^T [\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}] \\ &= \text{argmax}_{w} \left[\boldsymbol{w}^T \boldsymbol{X^T} \boldsymbol{X}\boldsymbol{w} - 2\boldsymbol{w}^T \boldsymbol{X^T} \boldsymbol{y} + \text{constant}\right]\end{aligned}$$

There is an analytical solution to this, and we obtain it by simply computing the gradient and setting it to $\mathbf{0}$.

$$\partial_w \left[\boldsymbol{w}^T \boldsymbol{X^T} \boldsymbol{X}\boldsymbol{w} - 2\boldsymbol{w}^T \boldsymbol{X^T} \boldsymbol{y}\right] = 2\boldsymbol{X^T}\boldsymbol{X}\boldsymbol{w} - 2\boldsymbol{X^T}\boldsymbol{y}$$

Setting this to $\mathbf{0}$, we obtain:

$$\boldsymbol{w}_{MLE} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$$

As we will see in homework 1, $(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$ can be viewed as the projection of $\boldsymbol{y}$ onto the column space of $\boldsymbol{X}$.

4.13 Bayesian Linear Regression

In the Bayesian framework, we also introduce a probability distribution on the weights. Here, we choose:

$$p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w} \mid \boldsymbol{m_0}, \boldsymbol{S_0})$$

Thus, we have:

$$p(\boldsymbol{y} \mid \boldsymbol{X}, \boldsymbol{w}, \boldsymbol{\mu}, \sigma^2) = \mathcal{N}(\boldsymbol{y} \mid \boldsymbol{\mu} + \boldsymbol{X^T}\boldsymbol{w}, \sigma^2\boldsymbol{I})$$

We assume that $\boldsymbol{\mu} = 0$.

The posterior then is of the form:

$$p(\boldsymbol{w} \mid \ldots) \propto \mathcal{N}(\boldsymbol{w} \mid \boldsymbol{m_0}, \boldsymbol{S_0})\mathcal{N}(\boldsymbol{y} \mid \boldsymbol{X}^T\boldsymbol{w}, \sigma^2\boldsymbol{I})$$

Applying the results obtained above with the linear Gaussian results, with:

$$\begin{aligned}b &= 0 \\ \boldsymbol{A} &= \boldsymbol{X}^T \\ \boldsymbol{\Sigma_y} &= \sigma^2\boldsymbol{I}\end{aligned}$$

Thus, we have:

$$\boldsymbol{S_N}^{-1} = \boldsymbol{S_0}^{-1} + \frac{1}{\sigma^2}\boldsymbol{X^T X}$$
$$\boldsymbol{m_N} = \boldsymbol{S_N}\left[\boldsymbol{S_0^{-1} m_0} + \boldsymbol{X^T y}\frac{1}{\sigma^2}\right]$$

Now, we compute the posterior predictive:

$$p(y \ | \boldsymbol{x}, y) = \int \mathcal{N}(y \ | \boldsymbol{w}^T \boldsymbol{x}, \sigma^2)\mathcal{N}(\boldsymbol{w} \,|\, \boldsymbol{m_N}, \boldsymbol{S_N})d\boldsymbol{w}$$

Using the form for the marginal derived earlier, we have:

$$p(y \ | \boldsymbol{x}, y) = \mathcal{N}(y \ | \boldsymbol{X^T m_N}, \sigma^2 + \boldsymbol{X^T S_0 X})$$

The variance term is particularly interesting because now the variance has dependence on the actual data; thus, the Bayesian method has thus produced a different result. The mean, however, is the same as the MAP estimate ($\boldsymbol{x}^T \boldsymbol{m_N}$)

4.14 Non Linear Regression

All the examples done so far have been in linear space. To define an adaptive basis, we simply transform point x with the transormation of our choice:

$$x \to \phi(x)$$

Examples include:

- $\phi_1(x) = \sin(x)$
- $\phi_2(x) = \sin(\lambda x)$
- $\phi_3(x) = \max(0, x)$
- $\phi(x; w) = \max(0, w' \top x)$

The last example is the core of neural networks and deep learning where the weights are learned for each level of w.

CS281: Advanced ML **September 18, 2017**

Lecture 5: Linear Classification

Lecturer: Sasha Rush *Scribes: Demi Guo, Artidoro Pagnoni, Luke Melas-Kyriazi*

5.15 Classification Introduction

Last time we saw linear regression. In linear regression we were predicting $y \in \mathbb{R}$, in classification instead we deal with a discrete set, for example $y \in \{0, 1\}$ or $y \in \{1, \dots, C\}$. This distinction only matters for this lecture, starting from next class we will generalize the topics and treat them as the same thing.

Among the many applications, linear classification is used in sentiment analysis, spam detection, and facial and image recognition. We will use generative models of the data, which means that we will model both the x and the y explicitly, and we are not keeping x fixed. In the case of the spam filter earlier, x is the email body, and y is the label {spam, not spam}. A generative model of the email and labels, we would model the distribution of x, of the text in the email itself, and not only the distribution of the category y.

We will explore the basic method of Naïve Bayes in detail. Even with a very simple method like Naïve Bayes with basic features it is possible to perform extremely well on many classification tasks when large training data sets are available. For example, this simple model performs almost as well (one percent point difference) as very complex methods on spam detection.

5.16 Naïve Bayes

Note that the term ”Bayes” in Naïve Bayes (NB) does not have to do with Bayesian modeling, or the presence of priors on parameters. We won’t have any priors for the moment. General Naïve Bayes takes the following form:

$$\begin{aligned} y &\sim \text{Cat}(\pi) && \text{[class distribution]} \\ \boldsymbol{x} | y &\sim \prod_j p(x_j \mid y) && \text{[class conditional]} \end{aligned}$$

where y is the class label and comes from a categorical distribution, and x_j is a dimension of the input x.

In Naïve Bayes, the form of the class distribution is fixed and parametrized independently from the class conditional distribution. The ”Naïve” term in ”Naïve Bayes” precisely refers to the conditional independence between y and $x_j|y$. Depending of what the data looks like we can choose a different form for the class conditional distribution.

Here we present three possible choices for the class conditional distribution:

- **Multivariate Bernoulli Naïve Bayes**:

  $$x_j | y \sim \text{Bern}(\mu_{jc}) \quad \text{if } y = c$$

  Here y takes values in a set of classes, and μ_{jc} is a parameter associated with a specific feature (or dimension) in the input and a specific class. We use multivariate Bernoulli when we only allow two possible values for each feature, therefore $x_j|y$ follows a Bernoulli distribution.

  We can think of x as living in a hyper cube, with each dimension j having an associated μ for each class c. From here we get the name multivariate Bernoulli distribution.

- **Categorical Naïve Bayes**:

  $$x_j | y \sim \text{Cat}(\mu_{jc}) \quad \text{if } y = c$$

  We use the Categorical Naïve Bayes when we allow different classes for each feature j, so $x_j|y$ follows a Categorical distribution.

- **Multivariate Normal Naïve Bayes**

  $$\boldsymbol{x}|y \sim \mathcal{N}(\boldsymbol{\mu}_c, \boldsymbol{\Sigma}^c_{diag})$$

  Note that here we use $\boldsymbol{x}$ vector and not a specific feature. Since we impose that $\boldsymbol{\Sigma}^c$ is a diagonal matrix, we have no covariance between features, so this comes down to having an independent normal for each feature (or dimension) of the output. This is also required by the "Naïve" assumption of conditional independence. We would use MVN Naïve Bayes when the features take continuous values in $\mathbb{R}^n$.

5.17 General Naïve Bayes

We consider the data points $\{(\boldsymbol{x}_n, y_n)\}$, without specifying a particular generative model. The likelihood of each data point is:

$$\begin{aligned} p(\boldsymbol{x}_n, y_n|\text{param}) &= p(y_n|\boldsymbol{\pi}) \prod_j p(x_{nj}|y_n, \text{param}) && (5.1) \\ &= \prod_c \pi_c^{(y_n=c)} \prod_j \prod_c p(x_{nj}|y_n)^{(y_n=c)} && (5.2) \end{aligned}$$

where in equation (5.2) we assume conditional independence (the "Naïve" assumption). The term $p(x_{nj}|y_n)$ depends on the generative model used for x and also on the class y_n.

We can then solve for the parameters maximizing the likelihood, which is equivalent to maximizing the log likelihood.

$$\begin{aligned} (\pi_{\text{MLE}}, \mu_{\text{MLE}}) &= \underset{(\boldsymbol{\pi},\boldsymbol{\mu})}{\text{argmax}} \sum_n \log p(\boldsymbol{x}_n, y_n|\text{param}) && (5.3) \\ &= \underset{(\boldsymbol{\pi},\boldsymbol{\mu})}{\text{argmax}} \sum_c N_c \log \pi_c + \sum_i \sum_c \sum_{n:y_n=c} \log p(x_{nj}|y_n) && (5.4) \\ &= \left(\underset{(\boldsymbol{\pi},\boldsymbol{\mu})}{\text{argmax}} \sum_c N_c \log \pi_c \right) + \left(\underset{(\boldsymbol{\pi},\boldsymbol{\mu})}{\text{argmax}} \sum_i \sum_c \sum_{n:y_n=c} \log p(x_{nj}|y_n) \right) && (5.5) \end{aligned}$$

Where $N_c = \sum_n \mathbb{1}(y_n = c)$, and $N =$ the number of data points.

This factors into two parts (5.10), the first only depending on $\boldsymbol{\pi}$ the other is the MLE for the class condition distribution on each feature or dimension of the input. This factorization allows to solve for the maximizing $\boldsymbol{\pi}$ and the maximizing parameters for the class conditional separately.

For example, if we use a Multivariate Bernoulli Naïve Bayes generative model we would get the following parameters from MLE:

$$\begin{aligned} \pi_c &= \frac{N_c}{N} && (5.6) \\ \mu_{jc} &= \frac{\sum_{n:y_n=c} x_{nj}}{N_c} = \frac{N_{cj}}{N_c} && (5.7) \end{aligned}$$

Again, where $N_c = \sum_n \mathbb{1}(y_n = c)$, $N_{cj} = \sum_n \mathbb{1}(y_n = c)x_{nj}$ and $N =$ number of data points.

5.18 Bayesian Naive Bayes: Add a Prior

Here, instead of working with a single distribution, we are working with multiple distributions. For simplicity, let's use the following factored **prior**:

$$p(\pi, \mu) = p(\pi) \prod_j \prod_c p(\mu_{jc})$$

where $p(\pi)$ represents the prior on class distribution and $\prod_j \prod_c p(\mu_{jc})$ represents prior on class conditional distribution.

Now, **what prior should we use?**

1. π: Dirichlet (goes with Categorical)
2. μ_{jc}:
   (a) Beta (goes with Bernoulli)
   (b) Dirichlet (goes with Categorical)
   (c) Normal (goes with Normal)

   Here, what distribution we choose depends on our choice of class conditional distribution.

Recall that we want to use conjugate priors to have a natural update (that's why we pair them up!). By using conjugate priors, we will have:

$$p(\pi|\text{data}) = Dir(N_1 + \alpha_1, \cdots, N_c + \alpha_c)$$
$$p(\mu_{jc}|\text{data}) = \beta((N_c - N_{jc}) + \beta_0, N_c + \beta_1)$$

5.18.1 Intuition

You can think of the α_i above as initial pseudocounts. Those pseudocounts give nonzero probability to features we haven't seen before, which is crucial for NLP. For unseen features, you could have a pseudocount of 1 or 0.5 (Laplace term) or something.

Because of this property, a Bayesian model helps prevents overfitting by introducing such priors: consider the spam email classification problem mentioned before. Say the word "subject" (call it feature j) always occurs in both classes ("spam" and "not spam"), so we estimate $\hat{\theta}_{jc} = 1$ (we overfit!) What will happen if we encounter a new email which does not have this word in it? Our algorithm will crash and burn! This is another manifestation of the black swan paradox discussed in Book Section 3.3.4.1. Note that this will not happen if we introduce pseudocounts to all features!

5.19 Posterior Predictive

$$p(\hat{y}, \hat{x} \mid \text{data}) = (\text{integrate over parameters})$$
$$\pi_c{}^{\text{MAP}} = \frac{N_c + \alpha_c}{N + \sum_c \alpha_c} (\text{Dirichlet MAP})$$
$$\mu_{jc}{}^{\text{MAP}} = \frac{N_{jc} + \beta_1}{N_c + \beta_1 + \beta_0} (\text{Beta MAP})$$

(How to derive this? Good exercise!)

5.20 More on Predictive

Now, let's consider a little bit more about what's happening in our predictive. Consider the email spam classification problem: given some features of an email, we want to predict if the email is a spam or not a spam. We have:

$$\begin{aligned} p(y = c|x, data) &\propto \pi_c \prod_j p(x_j|y) \text{ (try to generate observations from class)} \\ &= \pi_c \prod_j \mu_{jc}^{x_j} (1 - \mu_{jc})^{(1-x_j)} \text{ (informal parametrization)} \\ &= \exp(\log \pi_c + \sum_j x_j \log \mu_{jc} + (1 - x_j) \log(1 - \mu_{jc})) \text{ (take exp of log)} \\ &= \exp\left(\log \pi_c + \sum_j \log(1 - \mu_{jc}) + \sum_j x_j \log \frac{\mu_{jc}}{1 - \mu_{jc}}\right) \end{aligned}$$

where the first two term $log \pi_c + \sum_j log(1 - \mu_{jc})$ is a constant (we call it b for bias), and the last term $\sum_j x_j log \frac{\mu_{jc}}{1-\mu_{jc}}$ is linear (we call it θ).

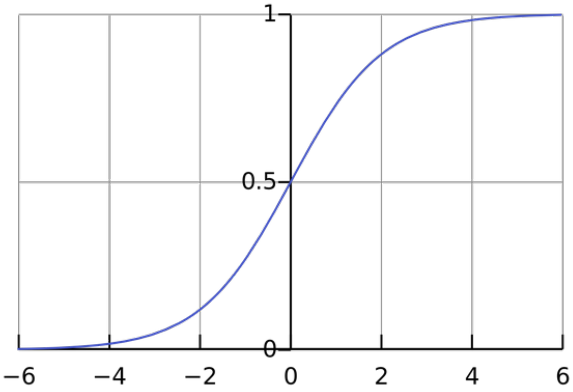


Figure 5.2: The Sigmoid Function

5.21 Multivariate Bernoulli Naive Bayes

For Multivariate Bernoulli NB, we will have:

$$\theta_{jc} = \log \frac{\mu_{jc}}{1-\mu_{jc}}$$
$$b_c = \log \pi_c + \log(1-\mu_{jc})$$

So, we have:

$$p(y = c \mid x) \propto \exp(\theta_c^T x + b_c)$$

Thus, in order to determine which class (”spam” or ”not spam”), for each class we simply compute a linear function with respect to x, and compare the two. Our $\theta x + b$ is going to be associated with a linear separator of the data. Even better, for prediction, we can simply compute θ and β (as shown above) using closed form for both MAP and MLE cases.

5.22 The Sigmoid Function

Before proceeding, we should name our variables to speak about them more easily.

We call μ the ”informal parameters” and θ the ”scores.” In the case of a Multivariate Bernouli model, we have the map $\theta_{jc} = \log \frac{\mu_{jc}}{1-\mu_{jc}}$, which we call the ”log odds.” We may also invert this relationship to find μ as a function of θ:

$$\theta = \log \frac{\mu}{1-\mu} \implies \mu = \frac{e^\theta}{1+e^\theta} = \frac{1}{1+e^{-\theta}} = \sigma(\theta)$$

We denote this function $\sigma(\theta)$ as the sigmoid function. The sigmoid function is a map from the real line to the interval $[0, 1]$, so is useful as a representation of probability. It is also a common building block in constructing neural networks, as we will see later in the course.

5.23 The Softmax Function

We will now return to the predictive $p(y = c|x) \propto \exp(\theta_c^T x + c_c)$ to try to compute the normalizer Z:

$$p(y = c|x) = \frac{1}{Z} \exp(\theta_c^T x + b_c)$$

In general, we can compute the normal by summing over all our classes.

$$Z(\theta) = \sum_{c'} = \exp(\theta_{c'}^T x + b_{c'})$$

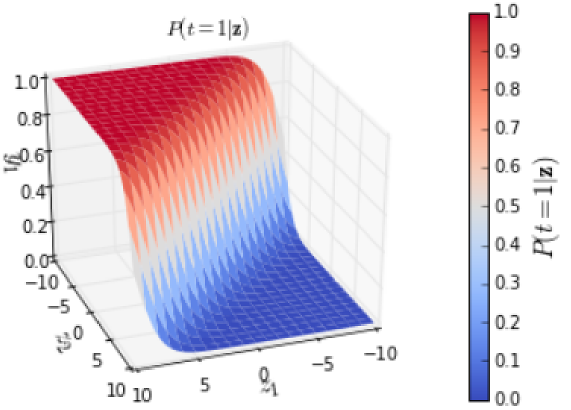


Figure 5.3: The Softmax Function

In practice, this summation is often computationally expensive. However, it is not necessary to compute this sum if we are only interested in the most likely class label given an input.

We call the resulting probability density function the *softmax*:

$$\text{softmax}(z)_i = \frac{\exp(z_i)}{\sum_{i'} \exp(z_{i'})}$$

This function generalizes the sigmoid function to multiple classes/dimensions. We call it the ”softmax” because we may think of it as a smooth, differentiable version of the function which simply returns 1 for the most likely class (or argmax).

5.24 Discriminative Classification

We may apply the mathematical tools developed in the generative classification setting discussed above to perform discriminative classification. In discriminative classification, we assume that our inputs x are fixed, rather than coming from some probability distribution.

We take the maximum likelihood estimate, as in linear regression, given that $p(y = c|x) \propto \exp(\theta_c^T x)$:

$$\text{MLE} : \underset{\theta}{\text{argmax}}\, p(y|x, \theta) = \underset{\theta}{\text{argmax}} \sum_n \log \text{softmax}(\theta_c^T x_n) c_n$$

What are the advantages and disadvantages of this approach? The primary disadvantage compared to methods we have seen earlier is that this maximum likelihood estimate has *no closed form*. It is also not clear how we might incorporate our prior (although there is recent work in this area). On the other hand, this equation is convex and it is easy (at least mathematically, not necessarily computationally) to compute gradients, so we may use gradient descent.

$$\frac{d(\cdot)}{d\theta_c} = \sum_n x_n \cdot \begin{cases} 1 - \text{softmax}(\theta_c^T x) & \text{if } y_n = c \\ \text{softmax}(\theta_c^T x) & \text{otherwise} \end{cases} \tag{5.8}$$

This model is known as **logistic regression** (even though it is used for classification, not regression) and is widely used in practice.

More Resources on Optimization

- Convex Optimization by Lieven Vandenberghe and Stephen P. Boyd