# CPSC 540: Machine Learning

## Density Estimation

Mark Schmidt

University of British Columbia

Winter 2020

# Admin

- Registration forms:
  - I will sign them at the end of class (need to submit prereq form first).
- Website/Piazza:
  - `https://www.cs.ubc.ca/~schmidtm/Courses/540-W20`.
  - `https://piazza.com/ubc.ca/winterterm22019/cpsc540`.
- Assignment 1 due tonight.
  - Gradescope submissions posted on Piazza.
  - Prereq form submitted separately on Gradescope.
  - Make sure to use your student number as your "name" in Gradescope.
  - You can use late days to submit next week if needed.

- Today is the last day to add or drop the course.

# Last Time: Structure Prediction

- "Classic" machine learning: models $p(y^i \mid x^i)$, where $y^i$ was a single variable.
  - In 340 we used simple distributions like the Gaussian and sigmoid.
- Structured prediction: $y^i$ could be a vector, protein, image, dependency tree,....
  - This requires defining more-complicated distributions.
- Before considering $p(y^i \mid x^i)$ for complicated $y^i$:
  - We'll first consider just modeling $p(x^i)$, without worrying about conditioning.

# Density Estimation

- The next topic we'll focus on is density estimation:

$$X = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix} \quad \tilde{X} = \begin{bmatrix} ? & ? & ? & ? \\ ? & ? & ? & ? \\ ? & ? & ? & ? \\ ? & ? & ? & ? \\ ? & ? & ? & ? \end{bmatrix}$$

  - What is probability of $[1\ 0\ 1\ 1]$?
  - Want to estimate probability of feature vectors $x^i$.

- For the training data this is easy:
  - Set $p(x^i)$ to "number of times $x^i$ is in the training data" divided by $n$.

- We're interested in the probability of test data,
  - What is probability of seeing feature vector $\tilde{x}^i$ for a new example $i$.

# Density Estimation Applications

- Density estimation could be called a "master problem" in machine learning.
  - Solving this problem lets you solve a lot of other problems.

- If you have $p(x^i)$ then:
  - Outliers could be cases where $p(x^i)$ is small.
  - Missing data in $x^i$ can be "filled in" based on $p(x^i)$.
  - Vector quantization can be achieved by assigning shorter code to high $p(x^i)$ values.
  - Association rules can be computed from conditionals $p(x^i_j \mid x^i_k)$.

- We can also do density estimation on $(x^i, y^i)$ jointly:
  - Supervised learning can be done by conditioning to give $p(y^i \mid x^i)$.
  - Feature relevance can be analyzed by looking at $p(x^i \mid y^i)$.

- If features are continuous, we are estimating the "probability density function".
  - I'll sloppily just say "probability" though.

# Unsupervised Learning

- Density estimation is an unsupervised learning method.
  - We only have $x^i$ values, but no explicit target labels.
  - You want to do "something" with them.

- Some unsupervised learning tasks from CPSC 340 (depending on semester):
  - Clustering: what types of $x^i$ are there?
  - Association rules: which $x_j$ and $x_k$ occur together?
  - Outlier detection: is this a "normal" $x^i$?
  - Latent-factors: what "parts" are $x^i$ made from?
  - Data visualization: what do the high-dimensional $x^i$ look like?
  - Ranking: which are the most important $x^i$?

- You can probably address all these if you can do density estimation.

# Bernoulli Distribution on Binary Variables

- Let's start with the simplest case: $x^i \in \{0, 1\}$ (e.g., coin flips),

$$X = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}.$$

- For IID data the only choice is the Bernoulli distribution:

$$p(x^i = 1 \mid \theta) = \theta, \quad p(x^i = 0 \mid \theta) = 1 - \theta.$$

- We can write both cases as

$$p(x^i \mid \theta) = \theta^{\mathcal{I}[x^i=1]}(1-\theta)^{\mathcal{I}[x^i=0]}, \text{ where } \mathcal{I}[y] = \begin{cases} 1 & \text{if } y \text{ is true} \\ 0 & \text{if } y \text{ is false} \end{cases}.$$

# Maximum Likelihood with Bernoulli Distribution

- MLE for Bernoulli likelihood with IID data is

$$\begin{aligned}
\operatorname*{argmax}_{0 \le \theta \le 1} p(X \mid \theta) &= \operatorname*{argmax}_{0 \le \theta \le 1} \prod_{i=1}^{n} p(x^i \mid \theta) \\
&= \operatorname*{argmax}_{0 \le \theta \le 1} \prod_{i=1}^{n} \theta^{\mathcal{I}[x^i=1]} (1-\theta)^{\mathcal{I}[x^i=0]} \\
&= \operatorname*{argmax}_{0 \le \theta \le 1} \underbrace{\theta^1 \theta^1 \cdots \theta^1}_{\text{number of } x_i = 1} \underbrace{(1-\theta)(1-\theta)\cdots(1-\theta)}_{\text{number of } x_i = 0} \\
&= \operatorname*{argmax}_{0 \le \theta \le 1} \theta^{n_1} (1-\theta)^{n_0},
\end{aligned}$$

  where $n_1$ is count of number of $1$ values and $n_0$ is the number of $0$ values.
- If you equate the derivative of the log-likelihood with zero, you get $\theta = \frac{n_1}{n_1+n_0}$.
- So if you toss a coin $50$ times and it lands heads $24$ times, your MLE is $24/50$.

# Multinomial Distribution on Categorical Variables

- Consider the multi-category case: $x^i \in \{1, 2, 3, \ldots, k\}$ (e.g., rolling di),

$$X = \begin{bmatrix} 2 \\ 1 \\ 1 \\ 3 \\ 1 \\ 2 \end{bmatrix}.$$

- The categorical distribution is

$$p(x^i = c \mid \theta_1, \theta_2, \ldots, \theta_k) = \theta_c,$$

where $\sum_{c=1}^{k} \theta_c = 1$.

- We can write this for a generic $x$ as

$$p(x^i \mid \theta_1, \theta_2, \ldots, \theta_k) = \prod_{c=1}^{k} \theta_c^{\mathcal{I}[x^i = c]}.$$

# Multinomial Distribution on Categorical Variables

- Using Lagrange multipliers (bonus) to handle constraints, the MLE is

$$\theta_c = \frac{n_c}{\sum_{c'} n_{c'}}. \quad (\text{"fraction of times you rolled a 4"})$$

- If we never see category $4$ in the data, should we assume $\theta_4 = 0$?
  - If we assume $\theta_4 = 0$ and we have a $4$ in test set, our test set likelihood is 0.
- To leave room for this possibility we often use "Laplace smoothing",

$$\theta_c = \frac{n_c + 1}{\sum_{c'}(n_{c'} + 1)}.$$

- This is like adding a "fake" example to the training set for each class.

# MAP Estimation with Bernoulli Distributions

- In the binary case, a generalization of Laplace smoothing is

$$\theta = \frac{n_1 + \alpha - 1}{(n_1 + \alpha - 1) + (n_0 + \beta - 1)},$$

- We get the MLE when $\alpha = \beta = 1$, and Laplace smoothing with $\alpha = \beta = 2$.
- This is a MAP estimate under a beta prior,

$$p(\theta \mid \alpha, \beta) = \frac{1}{B(\alpha, \beta)} \theta^{\alpha-1}(1-\theta)^{\beta-1},$$

where the beta function $B$ makes the probability integrate to one.

We want $\int_\theta p(\theta \mid \alpha, \beta) d\theta = 1$, so define $B(\alpha, \beta) = \int_\theta \theta^{\alpha-1}(1-\theta)^{\beta-1} d\theta$.

- Note that $B(\alpha, \beta)$ is constant in terms of $\theta$, it doesn't affect MAP estimate.
  - Above formula assumes $n_1 + \alpha > 1$ and $n_0 + \beta > 1$ (other cases in bonus).

# MAP Estimation with Categorical Distributions

- In the categorical case, a generalization of Laplace smoothing is

$$\theta_c = \frac{n_c + \alpha_c - 1}{\sum_{c'=1}^{k}(n_{c'} + \alpha_{c'} - 1)},$$

  which is a MAP estimate under a Dirichlet prior,

$$p(\theta_1, \theta_2, \ldots, \theta_k \mid \alpha_1, \alpha_2, \ldots, \alpha_k) = \frac{1}{B(\alpha)} \prod_{c=1}^{k} \theta_c^{\alpha_c - 1},$$

  where $B(\alpha)$ makes the multivariate distribution integrate to 1 over $\theta$,

$$B(\alpha) = \int_{\theta_1} \int_{\theta_2} \cdots \int_{\theta_{k-1}} \int_{\theta_k} \prod_{c=1}^{k} \left[\theta_c^{\alpha_c - 1}\right] d\theta_k d\theta_{k-1} \cdots d\theta_2 d\theta_1.$$

- Because of MAP-regularization connection, Laplace smoothing is regularization.

# Outline

# General Discrete Distribution

- Now consider the case where $x^i \in \{0,1\}^d$:
  - Words in e-mails, pixels in binary image, locations of cancers, and so on.

$$X = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix}.$$

- Now there are $2^d$ possible values of vector $x^i$.
  - General discrete distribution would consider $\theta_{0000}$, $\theta_{0001}$, $\theta_{0010}$, $\theta_{0011}$, $\theta_{0100}$,...
  - You can compute the MLE of this distribution in $O(nd)$.
    - See at most $n$ unique $x^i$ values, and using a hash data structure.
  - But unless we have a small number of repeated $x^i$ values, we'll hopelessly overfit.
- With finite dataset, we'll need to make assumptions...

# Product of Independent Distributions

- A common assumption is that the variables are independent:

$$p(x_1^i, x_2^i, \ldots, x_d^i \mid \Theta) = \prod_{j=1}^{d} p(x_j^i \mid \theta_j).$$

- Now we just need to model each column of $X$ as its own dataset:

$$X = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix} \quad \rightarrow \quad X_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \quad X_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \quad X_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}, \quad X_4 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}.$$

- A big assumption, but now you can fit Bernoulli for each variable.
  - We used a similar independence assumption in CPSC 340 for naive Bayes.

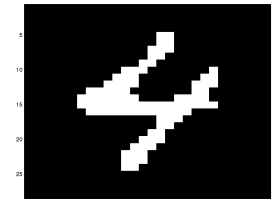
# Big Picture: Training and Inference

- Density estimation training phase:
  - Input is a matrix $X$.
  - Output is a model.
- Density estimation prediction phase:
  - Input is a model, and possibly test data $\tilde{X}$
  - Many possible prediction tasks:
    - Measure probability of test examples $\tilde{x}^i$.
    - Generate new samples $x$ according to the distribution.
    - Find configuration $x$ maximizing $p(x)$.
    - Compute marginal probability like $p(x_j = c)$ for somr variable $j$ and value $c$.
    - Compute conditional queries like $p(x_j = c \mid x_{j'} = c')$.
- We call these inference tasks.
  - More complicated than superised learning.
    - In supervised learning, inference was "find $\hat{y}^i$" or "compute $p(y = c \mid w, x)$.

# Example: Independent vs. General Discrete on Digits

- Consider handwritten images of digits:

$$x^i = \text{vec}\left( \right),$$



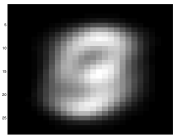

  so each row of $X$ contains all pixels from one image of a 0, 1, 2, . . . , or a 9.

- Previously we had labels and wanted to recognize that this is a 4.
- In density estimation we want probability distribution over images of digits.

- Given an image, what is the probability that it's a digit?
- Sampling from the density estimator should generate images of digits.

# Example: Independent vs. General Discrete on Digits

- Fitting independent Bernoullis to this data gives a parameter $\theta_j$ for each pixel $j$.
  - MLE is "fraction of times we have a $1$ at pixel $j$":

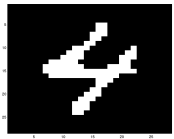


- Samples generated from independent Bernoulli model:

  - Flip a coin that lands hands with probability $\theta_j$ for each pixel $j$.
- This is clearly a terrible model: misses dependencies between pixels.

# Example: Independent vs. General Discrete on Digits

- Here is a sample from the MLE with the general discrete distribution:



- Here is an image with a probability of 0:

- This model memorized training images and doesn't generalize.
  - MLE puts probability at least $1/n$ on training images, and $0$ on non-training images.

# Density Estimation and Fundamental Trade-off

- "Product of independent" distributions (with $d$ parameters):
  - Easily estimate each $\theta_c$ but can't model many distributions.
- General discrete distribution (with $2^d$ parameters):
  - Hard to estimate $2^d$ parameters but can model any distribution.

- An unsupervised version of the fundamental trade-off:
  - Simple models often don't fit the data well but don't overfit much.
  - Complex models fit the data well but often overfit.

- We'll consider models that lie between these extremes:
  1. Mixture models.
  2. Markov models.
  3. Graphical models.
  4. Boltzmann machines.
  5. Variational autoencoders.
  6. Generative adversarial networks.

# Summary

- Density estimation: unsupervised modelling of probability of feature vectors.
- Categorical distribution for modeling discrete data.
- MAP estimation with beta and Dirichlet priors ("Laplace smoothing").
- Product of independent distributions is simple/crude density estimation method.
- Next time: more about the normal distribution than you ever wanted to know.

# Lagrangian Function for Optimization with Equality Constraints

- Consider minimizing a differentiable $f$ with linear equality constraints,

$$\operatorname*{argmin}_{Aw=b} f(w).$$

- The Lagrangian of this problem is defined by

$$L(w, v) = f(w) + v^T(Aw - b),$$

  for a vector $v \in \mathbb{R}^m$ (with $A$ being $m$ by $d$).
- At a solution of the problem we must have

$$\nabla_w L(w, v) = \nabla f(w) + A^T v = 0 \quad \text{(gradient is orthogonal to constraints)}$$
$$\nabla_v L(w, v) = Aw - b = 0 \quad \text{(constraints are satisfied)}$$

- So solution is stationary point of Lagrangian.

# Lagrangian Function for Optimization with Equality Constraints

- Scans from Bertsekas discussing Lagrange multipliers (also see CPSC 406).

### 3.1 NECESSARY CONDITIONS FOR EQUALITY CONSTRAINTS

In this section we consider problems with equality constraints of the form

$$\begin{aligned} &\text{minimize } f(x) \\ &\text{subject to } h_i(x) = 0, \qquad i = 1, \ldots, m. \end{aligned} \tag{ECP}$$

We assume that $f : \Re^n \mapsto \Re$, $h_i : \Re^n \mapsto \Re$, $i = 1, \ldots, m$, are continuously differentiable functions. *All the necessary and the sufficient conditions of this chapter relating to a local minimum can also be shown to hold if $f$ and $h_i$ are defined and are continuously differentiable within just an open set containing the local minimum.* The proofs are essentially identical to those given here.

For notational convenience, we introduce the constraint function $h : \Re^n \mapsto \Re^m$, where

$$h = (h_1, \ldots, h_m).$$

We can then write the constraints in the more compact form

$$h(x) = 0. \tag{3.1}$$

Our basic Lagrange multiplier theorem states that for a given local minimum $x^*$, there exist scalars $\lambda_1, \ldots, \lambda_m$, called *Lagrange multipliers*, such that

$$\nabla f(x^*) + \sum_{i=1}^{m} \lambda_i \nabla h_i(x^*) = 0. \tag{3.2}$$

There are two ways to interpret this equation:

(a) The cost gradient $\nabla f(x^*)$ belongs to the subspace spanned by the constraint gradients at $x^*$. The example of Fig. 3.1.1 illustrates this interpretation.

(b) The cost gradient $\nabla f(x^*)$ is orthogonal to the subspace of *first order feasible variations*

$$V(x^*) = \{\Delta x \mid \nabla h_i(x^*)'\Delta x = 0,\ i = 1, \ldots, m\}.$$

This is the subspace of variations $\Delta x$ for which the vector $x = x^* + \Delta x$ satisfies the constraint $h(x) = 0$ up to first order. Thus, according to the Lagrange multiplier condition of Eq. (3.2), at the local minimum $x^*$, the first order cost variation $\nabla f(x^*)'\Delta x$ is zero for all variations

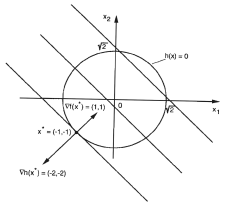



**Figure 3.1.1.** Illustration of the Lagrange multiplier condition (3.1) for the problem

$$\begin{aligned} &\text{minimize } x_1 + x_2 \\ &\text{subject to } x_1^2 + x_2^2 = 2. \end{aligned}$$

At the local minimum $x^* = (-1, -1)$, the cost gradient $\nabla f(x^*)$ is normal to the constraint surface and is therefore, collinear with the constraint gradient $\nabla h(x^*) = (-2, -2)$. The Lagrange multiplier is $\lambda = 1/2$.

$\Delta x$ in this subspace. This statement is analogous to the "zero gradient condition" $\nabla f(x^*) = 0$ of unconstrained optimization.

Here is a formal statement of the main Lagrange multiplier theorem:

**Proposition 3.1.1: (Lagrange Multiplier Theorem – Necessary Conditions)** Let $x^*$ be a local minimum of $f$ subject to $h(x) = 0$, and assume that the constraint gradients $\nabla h_1(x^*), \ldots, \nabla h_m(x^*)$ are linearly independent. Then there exists a unique vector $\lambda^* = (\lambda_1^*, \ldots, \lambda_m^*)$, called a *Lagrange multiplier vector*, such that

$$\nabla f(x^*) + \sum_{i=1}^{m} \lambda_i^* \nabla h_i(x^*) = 0. \tag{3.3}$$

If in addition $f$ and $h$ are twice continuously differentiable, we have

# Lagrangian Function for Optimization with Equality Constraints

- We can use these optimality conditions,

$$\nabla_w L(w, v) = \nabla f(w) + A^T v = 0 \quad \text{(gradient is orthogonal to constraints)}$$
$$\nabla_v L(w, v) = Aw - b = 0 \quad \text{(constraints are satisfied)}$$

  to solve some constrained optimization problems.

- A typical approach might be:
  1. Solve for $w$ in the equation $\nabla_w L(w, v) = 0$ to get $w = g(v)$ for some function $g$.
  2. Plug this $w = g(v)$ into the the equation $\nabla_v L(w, v) = 0$ to solve for $v$.
  3. Use this $v$ in $g(v)$ to get the optimal $w$.

- But note that these are necessary conditions (may need to check it's a min).

# Beta Distribution with $\alpha < 1$ and $\beta < 1$

- Wikipedia has a rather extensive article on the beta distribution: `https://en.wikipedia.org/wiki/Beta_distribution`
- In their picture of the beta distribution with $\alpha = \beta = 0.5$, you see that it's "U"-shaped, with modes at the extreme values of 0 or 1. I think of this as regularizing towards the coin being biased, but you're not sure whether the coin is biased towards heads or tails.
- Also, the MAP formula given in class only works in $n_1 + \alpha$ and $n_0 + \alpha$ are both greater than 1. This trivial holds for Laplace smoothing and the MLE case, but doesn't hold if you haven't seen both heads and tails when $\alpha$ and $\beta$ are less than 1. In that case, the MAP will be either 0 or 1 or both depending on the precise values.