COMPSCI 689
Lecture 15: Probabilistic Supervised Learning II

Benjamin M. Marlin

College of Information and Computer Sciences
University of Massachusetts Amherst

Slides by Benjamin M. Marlin (marlin@cs.umass.edu).

Probabilistic Supervised Learning

- In probabilistic supervised learning, our goal is to model the true probability distribution of the outputs $y \in \mathcal{Y}$ given the inputs $\mathbf{x} \in \mathcal{X}$.
- If $\mathcal{Y}$ is discrete, our goal is to model $P_*(Y = y|\mathbf{X} = \mathbf{x})$ with a conditional parametric probability mass function $P(Y = y|\mathbf{X} = \mathbf{x}, \theta)$.
- If $\mathcal{Y}$ is uncountable, our goal is to model $p_*(Y = y|\mathbf{X} = \mathbf{x})$ with a conditional parametric probability density function $p(Y = y|\mathbf{X} = \mathbf{x}, \theta)$.

Generating Supervised Probabilistic Models

- We can very flexibly generate probabilistic supervised learning models by combining unconditional probability models with regression models that predict their parameter values.
- We can select different probability models to provide distributions over different output spaces.
- We can use any type of regression model including both linear and non-linear models.
- We may need to apply an invertible transformation to the regression model outputs to ensure that the predicted parameter values always fall in the parameter space Φ of the unconditional model.

Learning Supervised Probabilistic Models

- So long as all model components are differentiable functions, we can learn the model parameters θ by minimizing the conditional negative log likelihood function given a data set $\mathcal{D}$:

$$nll(\mathcal{D}, \theta) = -\sum_{n=1}^{N} \log P(Y = y | \mathbf{X} = \mathbf{x}, \theta)$$

when comp^n is intractable ?!
pseudo likelihood

$$nll(\mathcal{D}, \theta) = -\sum_{n=1}^{N} \log p(Y = y | \mathbf{X} = \mathbf{x}, \theta)$$

- This is referred to as *maximum likelihood estimation*.

Example: Probabilistic Logistic Regression

→ INVERTIBLE
→ DIFF. & CONTINUOUS

Bernoulli Model

- Suppose that $\mathcal{Y} = \{-1, 1\}$ and $\mathcal{X} \in \mathbb{R}^D$.
- Base Model: $P(Y = y|\phi) = \phi^{[y=1]}(1-\phi)^{[y=-1]}$
- Conditional Model:

$$P(Y = y|\mathbf{X} = \mathbf{x}, \theta) = \phi(\mathbf{x})^{[y=1]}(1-\phi(\mathbf{x}))^{[y=-1]}$$

assuming sigmoid a logistic fnc

- Parameter Prediction Function: $\phi(\mathbf{x}) = \sigma(\mathbf{x}\theta) = \frac{1}{1+\exp(-\mathbf{x}\theta)}$
- Parameter Transformation Function: $\sigma(a) = \frac{1}{1+\exp(-a)}$
- Model Parameters: θ
- Negative Log Likelihood:

$$nll(\mathcal{D}, \theta) = -\sum_{n=1}^{N} \log P(Y = y_n|\mathbf{X} = \mathbf{x}_n, \theta)$$

Example: Probabilistic Logistic Regression

$$
\begin{aligned}
nll(\mathcal{D}, \theta) &= -\sum_{n=1}^{N} \log P(Y = y_n | \mathbf{X} = \mathbf{x}_n, \theta) \\
&= -\sum_{n=1}^{N} \log \left(\phi(\mathbf{x}_n)^{[y_n=1]} (1 - \phi(\mathbf{x}_n))^{[y_n=-1]} \right) \\
&= -\sum_{n=1}^{N} \left([y_n = 1] \log \phi(\mathbf{x}_n) + [y_n = -1] \log(1 - \phi(\mathbf{x}_n)) \right)
\end{aligned}
$$

Example: Probabilistic Logistic Regression

- Note that $\phi(\mathbf{x}) = \frac{1}{1+\exp(-\mathbf{x}\theta)}$ so:
  $\log(\phi(\mathbf{x})) = -\log(1 + \exp(-\mathbf{x}\theta))$.

- This means that:
  $[y = 1] \log(\phi(\mathbf{x})) = -[y = 1] \log(1 + \exp(-y\mathbf{x}\theta))$.

  LOGISTIC LOSS FUNCTION

- Note that: $(1 - \phi(\mathbf{x})) = 1 - \frac{1}{1+\exp(-\mathbf{x}\theta)} = \frac{\exp(-\mathbf{x}\theta)}{1+\exp(-\mathbf{x}\theta)}$
  $= \frac{1}{\exp(\mathbf{x}\theta)+1} = \frac{1}{1+\exp(\mathbf{x}\theta)}$.

- This means that:
  $[y = -1] \log(1 - \phi(\mathbf{x})) = -[y = -1] \log(1 + \exp(-y\mathbf{x}\theta))$.

Example: Probabilistic Logistic Regression

JUST COMBINE FROM BEFORE, The unconditional model comes back to the original logistic regression problem. CONVEX DIFF.

$$nll(\mathcal{D}, \theta) = -\sum_{n=1}^{N} \log\left(\phi(\mathbf{x}_n)^{[y_n=1]}(1-\phi(\mathbf{x}_n))^{[y_n=-1]}\right)$$

REFER COLAB.

$$= -\sum_{n=1}^{N} \left([y_n = 1]\log\phi(\mathbf{x}_n) + [y_n = -1]\log(1-\phi(\mathbf{x}_n))\right)$$

DECISION BOUNDARY

≥ 0.5

$$= -\sum_{n=1}^{N} \Big(-[y_n = 1]\log(1+\exp(-y_n\mathbf{x}_n\theta)) - [y_n = -1]\log(1+\exp(-y_n\mathbf{x}_n\theta))\Big)$$

Once we just got the value how close is to its category. Now we get an actual prob. for that.

$$= \sum_{n=1}^{N} \log(1+\exp(-y_n\mathbf{x}_n\theta))$$

Example: Probabilistic Logistic Regression

- Suppose that instead of defining $\mathcal{Y} = \{-1, 1\}$, we choose $\mathcal{Y} = \{0, 1\}$.
- In this case, the negative log likelihood can be written as:

$$
\begin{aligned}
nll(\mathcal{D}, \theta) &= -\sum_{n=1}^{N} \log \left(\phi(\mathbf{x}_n)^{y_n} (1 - \phi(\mathbf{x}_n))^{1-y_n} \right) \\
&= -\sum_{n=1}^{N} \left(y_n \log \phi(\mathbf{x}_n) + (1 - y_n) \log(1 - \phi(\mathbf{x}_n)) \right)
\end{aligned}
$$

- This function is referred to as the *binary cross entropy loss*.
- Minimizing the logistic loss under ERM and either version of the probabilistic logistic regression NLL function lead to equivalent optimization problems.

Regularizer can be interpreted as another probabilistic perspective? Bayesian.....

Example: Non-Linear Probabilistic Classification

- If we want to build a non-linear probabilistic binary classifier we can use the probabilistic logistic regression model with a basis expansion or a kernel.
- We can also model the parameter prediction function $\phi(\mathbf{x})$ using the logistic transform applied to an arbitrary neural network model.

1 LOGISTIC LOSS
2 HINGE LOSS
3 PROBABILISTIC PERSP. + MAX. LOG LIKELIHOOD

Categorical Random Variables

Suppose we have a die with C sides. Each side potentially comes up with a different probability. How can we model this with a random variable?

- Values: $\mathcal{Z} = \{1, 2, ..., C\}$
- Parameters: For each c we have $\phi_c \geq 0$. We also have $\sum_{c=1}^{C} \phi_c = 1$
- Parameter Space: $\Phi = \mathcal{S}$
- Mass Function: $P(Z = z|\phi) = \prod_{c=1}^{C} \phi_c^{[z=c]}$

MLE for Categorical Random Variables

- Suppose we have a data set $\mathcal{D} = \{z_1, ..., z_N\}$ such that $z_n \in \{1, 2, ..., C\}$ for all N.

- $nll(\mathcal{D}, \phi_{1:C}) = -\sum_{n=1}^{N} \sum_{c=1}^{C} [z_n = c] \log \phi_c$

- To find the MLE we need to minimize $nll(\mathcal{D}, \phi_{1:C})$ while enforcing the equality constraint $\sum_{c=1}^{C} \phi_c = 1$.

- We obtain $\hat{\phi}_c = \frac{\sum_{n=1}^{N} [z_n = c]}{N}$.

Multiclass Logistic Regression

- Suppose that $\mathcal{Y} = \{1, ..., C\}$ and $\mathcal{X} \in \mathbb{R}^D$.
- Base Model: $P(Y = y|\phi) = \prod_{c=1}^{C} \phi_c^{[y=c]}$
- Conditional Model: $P(Y = y|\mathbf{X} = \mathbf{x}, \theta) = \prod_{c=1}^{C} \phi_c(\mathbf{x})^{[y=c]}$
- Parameter Prediction Function: $\phi_c(\mathbf{x}) = \text{softmax}(\mathbf{x}, c, \theta)$
- Parameter Transformation Function:

$$\text{softmax}(\mathbf{x}, c, \theta) = \frac{\exp(\mathbf{x}\mathbf{w}_c)}{\sum_{k=1}^{C} \exp(\mathbf{x}\mathbf{w}_k)}$$

- Model Parameters: $\theta = [\mathbf{w}_1, ..., \mathbf{w}_C]$
- Negative Log Likelihood: $nll(\mathcal{D}, \theta) = -\sum_{n=1}^{N} \log P(Y = y_n|\mathbf{X} = \mathbf{x}_n, \theta)$

Multiclass Logistic Regression

- Putting all of this together, we have the model:

$$P(Y = y|\mathbf{x}, \theta) = \prod_{c=1}^{C} \left(\frac{\exp(\mathbf{x}\mathbf{w}_c)}{\sum_{k=1}^{C} \exp(\mathbf{x}\mathbf{w}_k)} \right)^{[y=c]}$$

- There is one weight vector $\mathbf{w}_c$ per class (assuming bias absorption).
- Note that this parameterization is actually redundant due to the normalization constraint. This redundancy can be removed by asserting that $\mathbf{w}_c = 0$ for one of the C classes.

Multiclass Logistic Regression

- Putting all of this together, we have the model:

$$P(Y=y|\mathbf{x},\theta)=\prod_{c=1}^{C}\left(\frac{\exp(\mathbf{x}\mathbf{w}_c)}{\sum_{k=1}^{C}\exp(\mathbf{x}\mathbf{w}_k)}\right)^{[y=c]}$$

- The NLL function simplifies to:

$$\begin{aligned}
nll(\mathcal{D},\theta) &= -\sum_{n=1}^{N}\log P(Y=y_n|\mathbf{X}=\mathbf{x}_n,\theta)\\
&= -\sum_{n=1}^{N}\sum_{c=1}^{C}[y_n=c]\left(\mathbf{x}_n\mathbf{w}_c-\log\left(\sum_{k=1}^{C}\exp(\mathbf{x}_n\mathbf{w}_k)\right)\right)\\
&= -\sum_{n=1}^{N}\sum_{c=1}^{C}\left([y_n=c]\mathbf{x}_n\mathbf{w}_c-\log\left(\sum_{k=1}^{C}\exp(\mathbf{x}_n\mathbf{w}_k)\right)\right)
\end{aligned}$$

Poisson Random Variables

- Suppose we have a process that produces data such that $z \in \mathbb{Z}^{\geq 0}$.
- One distribution that matches the support of z is the Poisson distribution:

$$P(Y = y|\lambda) = \frac{\lambda^y \exp(-\lambda)}{y!}$$

- This distribution has the constraint that $\lambda \in \mathbb{R}^{>0}$.

Poisson Regression

- Suppose that $\mathcal{Y} = \mathbb{Z}^{\geq 0}$ and $\mathcal{X} \in \mathbb{R}^D$.
- Base Model: $P(Y = y|\lambda) = \frac{\lambda^y \exp(-\lambda)}{y!}$
- Conditional Model: $P(Y = y|\mathbf{X} = \mathbf{x}, \theta) = \frac{\lambda(\mathbf{x})^y \exp(-\lambda(\mathbf{x}))}{y!}$
- Parameter Prediction Function: $\lambda(\mathbf{x}) = \exp(\mathbf{x}\theta)$
- Model Parameters: θ
- Negative Log Likelihood:

$$nll(\mathcal{D}, \mathbf{w}) = -\sum_{n=1}^{N} \log P(Y = y_n|\mathbf{X} = \mathbf{x}_n, \theta)$$

Poisson Regression

- This gives us the model:

$$P(Y = y|\mathbf{X} = \mathbf{x}, \theta) = \frac{\exp(\mathbf{x}\theta)^y \exp(-\exp(\mathbf{x}\theta))}{y!}$$

- And the NLL simplifies to:

$$\begin{aligned} nll(\mathcal{D}, \mathbf{w}) &= -\sum_{n=1}^{N} \log P(Y = y_n|\mathbf{X} = \mathbf{x}_n, \theta) \\ &= -\sum_{n=1}^{N} (y_n \mathbf{x}_n \theta - \exp(\mathbf{x}_n \theta) - \log(y_n!)) \end{aligned}$$

Making Predictions

- Given a probabilistic supervised model, we can produce an estimate of the conditional probability of y given $\mathbf{x}$ by plugging the estimated parameters $\hat{\theta}$ into the model.
- In the case of discrete y, when we need to issue a prediction, we typically predict the value that achieves the maximum conditional probability given $\mathbf{x}$ and $\hat{\theta}$:

$$\hat{y} = \underset{y \in \mathcal{Y}}{\arg\max}\, P(Y = y | \mathbf{X} = \mathbf{x}, \hat{\theta})$$

- In the case of continuous y, we can predict different functions of the conditional distribution. The most commonly used prediction is the conditional mean of y:

$$\hat{y} = E_{p(Y=y|\mathbf{X}=\mathbf{x},\hat{\theta})}[y] = \int_{\mathcal{Y}} yp(Y = y|\mathbf{X} = \mathbf{x}, \hat{\theta}dy$$