Probabilistic ML II

CS771: Introduction to Machine Learning

Purushottam Kar

Recap of Last Lecture

Conditional expectation, conditional (co)variance, conditional mode

All rules continue to hold (but if conditioned uniformly across)

Be careful to condition consistently: conditioning should be on the same random variable(s) throughout, as well as be on the same values for those r.v.

Probabilistic ML: instead of predicting just one value of the label, predict a PMF over all possible values the label could take (i.e. support)

Can use this to make a single prediction e.g. $\hat{y} = \arg\max_{y \in \mathcal{Y}} \mathbb{P}[Y = y \mid \mathbf{x}, \mathbf{w}]$

Can use expectation/median as well – will see later

Can use this to find out if ML model is confident in its prediction or not

Bernoulli Distributions

These are probability distributions over the support $\{0,1\}$

Very useful in binary classification as labels are often named $\{0,1\}$

Arguably the simplest of all distributions. PMF of a r.v. Y with Bernoulli distribution is uniquely specified by just specifying $\mathbb{P}[Y = 1] = p$

Using complement rule we automatically get $\mathbb{P}[Y = 0] = 1 - p$

p *called "success probability" or "bias"*

Do not confuse this with the bias of linear model – not the same thing!

Mean: p

Mode: 1 if $p > 0.5$, 0 if $p < 0.5$, $\{0,1\}$ if $p = 0.5$

Variance: $p(1 - p)$

Rademacher Distributions

These are probability distributions over the support $\{-1,1\}$

- *Very similar to Bernoulli distributions except that support is different*
- *If X is distributed as Bernoulli then $2X - 1$ is distributed as Rademacher*
- *If Y is distributed as Bernoulli then $(Y + 1)/2$ is distributed as Bernoulli*

Also extremely simple distribution. PMF of a r.v. Y with Rademacher distribution is uniquely specified by just specifying $\mathbb{P}[Y = 1] = p$

- *Using complement rule we automatically get $\mathbb{P}[Y = -1] = 1 - p$*
- *Often, papers refer to Rademacher distribution only in special case $p = 0.5$*

Mean: $2p - 1$ (**Hint**: use scaling and sum rules for expectation)

Mode: 1 if $p > 0.5$, -1 if $p < 0.5$, $\{-1,1\}$ if $p = 0.5$

Variance: $4 \cdot p(1 - p)$ (**Hint**: use scaling and shift rules for variance)

Probabilistic Binary Classification

Find a way to map every data point $\mathbf{x}$ to a Rademacher distribution

Another way of saying this: map every data point $\mathbf{x}$ *to a prob* $p_{\mathbf{x}} \in [0,1]$

Will give us a PMF $[1 - p_{\mathbf{x}}, p_{\mathbf{x}}]$ *i.e.* $\mathbb{P}[-1 \mid \mathbf{x}] = 1 - p_{\mathbf{x}}, \mathbb{P}[+1 \mid \mathbf{x}] = p_{\mathbf{x}}$

If using mode predictor i.e. $\hat{y} = \arg\max_{y \in \mathcal{Y}} \mathbb{P}[Y = y \mid \mathbf{x}]$ then this PMF will give us the correct label only if the following happens

When the true label of $\mathbf{x}$ *is* $+1$, $p_{\mathbf{x}} > 1 - p_{\mathbf{x}}$, *in other words* $p_{\mathbf{x}} > 0.5$

When the true label of $\mathbf{x}$ *is* -1, $1 - p_{\mathbf{x}} > p_{\mathbf{x}}$, *in other words* $p_{\mathbf{x}} < 0.5$

Note that if $p_{\mathbf{x}} = 0.5$, *it means ML model is totally confused about label of* $\mathbf{x}$

Data points for whom $p_{\mathbf{x}} = 0.5$ *or even* $p_{\mathbf{x}} \approx 0.5$ *are on decision boundary!!*

Of course, as usual we want a healthy margin

If true label of the data point $\mathbf{x}$ *is* $+1$, *then we want* $p_{\mathbf{x}} \gg 0.5$ *i.e.* $p_{\mathbf{x}} \approx 1$

If true label of the data point $\mathbf{x}$ *is* -1, *then we want* $p_{\mathbf{x}} \ll 0.5$ *i.e.* $p_{\mathbf{x}} \approx 0$

Probabilistic Binary Classification

How to map feature vectors $\mathbf{x}$ to probability values $p_{\mathbf{x}} \in [0,1]$?

Could treat it as a regression problem since prob values $\in \mathbb{R}$ after all

Will need to modify the training set a bit to do this (basically change all -1 *labels to* 0 *since we want* $p_{\mathbf{x}} = 0$ *if the label is* -1

Could use kNN, DT etc to solve this regression problem

Using linear models to do this presents a challenge

If we learn a linear model $\mathbf{w}$ *using ridge regression it may happen that for some data point* $\mathbf{x}$*, we have* $\mathbf{w}^\top\mathbf{x} < 0$ *or else* $\mathbf{w}^\top\mathbf{x} > 1$

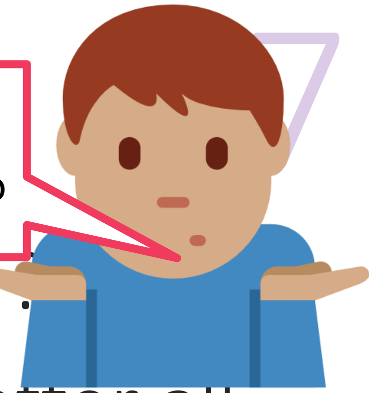
$p_{\mathbf{x}}$ *wont make sense in this case – not a valid PMF!!*

kNN, DT don't suffer from this problem since they always predict a $p_{\mathbf{x}} \in [0,1]$

kNN, DT use averages of a bunch of train labels to obtain test prediction – the average of a bunch of 0s and 1s is always a value in the range $[0,1]$

Probabilistic Binary Class

So can we never use linear models to do probabilistic ML?



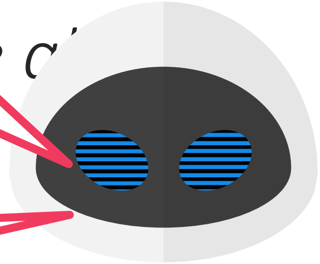
How to map feature vectors $\mathbf{x}$ to probability values $p_{\mathbf{x}} \in [0,1]$

Could treat it as a regres after all

We can – one way to solve the problem of using linear methods to map $x \mapsto [0,1]$ is called *logistic regression* – have seen it before

Will need to modify the t ge o

labels to 0 since we want

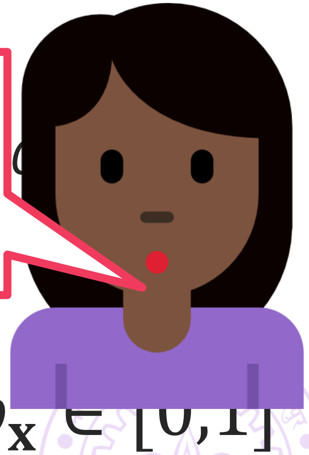


Yes, but there is a trick involved. Let us take a look at it

Could use kNN, DT etc to solve this regression problem

Using linear models to do this presents a challenge

Ah! The name makes sense now – logistic regression is used to solve binary classification problems but since it does so by mapping $x \mapsto [0,1]$, experts thought it would be cool to have the term “regression” in the name



If

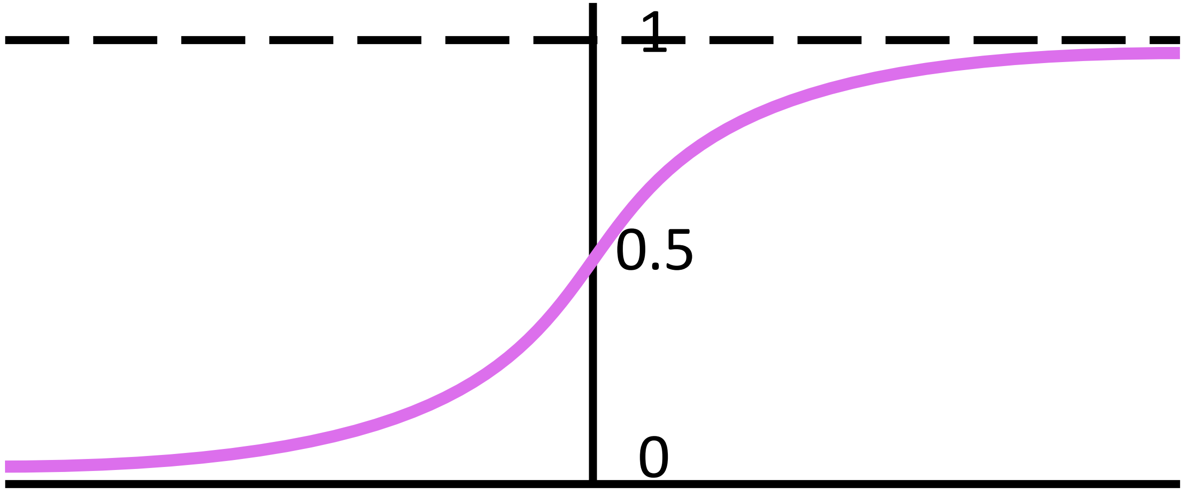
so

$p_{\mathbf{x}}$ wont make sense in this case – not a valid PMF!!

kNN, DT don’t suffer from this problem since they always predict a $p_{\mathbf{x}} \in [0,1]$

kNN, DT use averages of a bunch of train labels to obtain test prediction – the average of a bunch of 0s and 1s is always a value in the range $[0,1]$

Sigmoid Function



$$\sigma(t) = \frac{1}{1+\exp(-t)} = \frac{\exp(t)}{\exp(t)+1}$$

$$1 - \sigma(t) = \frac{1}{1+\exp(t)} = \sigma(-t)$$

Trick: learn a linear model $\mathbf{w}$ and map $\mathbf{x} \mapsto \sigma(\mathbf{w}^\top \mathbf{x})$

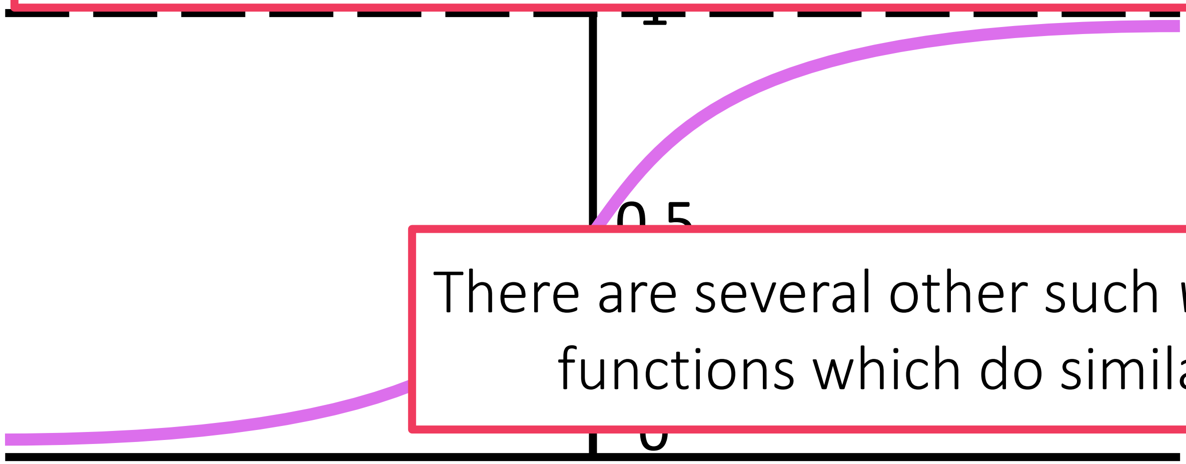
May have an explicit/hidden bias term as well

This will always give us a value in the range $[0,1]$*, hence give a valid PMF*

Note that $\sigma(t) > 0.5$ if $t > 0$ and $\sigma(t) < 0.5$ if $t < 0$ and also that $\sigma(t) \rightarrow 1$ as $t \rightarrow \infty$ and $\sigma(t) \rightarrow 0$ as $t \rightarrow -\infty$

This means that our sigmoidal map will predict $p_{\mathbf{x}} \approx 1$ *if* $\mathbf{w}^\top \mathbf{x} \gg 0$ *and* $p_{\mathbf{x}} \approx 0$ *if* $\mathbf{w}^\top \mathbf{x} \ll 0$

Nice! So I want to learn a linear model $\mathbf{w}$ such that once I do this sigmoidal map, data points with label $+1$ get mapped to a probability value close to 1 whereas data points with label -1 get mapped to a probability value close to 0



$$\sigma(t) = \frac{1}{1+\exp(-t)} = \frac{\quad}{\exp(t)+1}$$

There are several other such *wrapper/quashing/link/activation* functions which do similar jobs e.g. tanh, ramp, ReLU

Trick: learn a linear model $\mathbf{w}$ and map $\mathbf{x} \mapsto \sigma(\mathbf{w}^\top\mathbf{x})$

May have an explicit/hidden bias terr

This will always give us a value in the range [0,1], hence give a valid

How do I learn such a model $\mathbf{w}$?

Note that $\sigma(t) > 0.5$ if $t > 0$ and $\sigma(t) < 0.5$ if $t < 0$ and als $\sigma(t) \to 1$ as $t \to \infty$ and $\sigma(t) \to 0$ as $t \to -\infty$

This means that our sigmoidal map will predict $p_\mathbf{x} \approx 1$ *if* $\mathbf{w}^\top\mathbf{x} \gg 0$ *and* $p_\mathbf{x} \approx 0$ *if* $\mathbf{w}^\top\mathbf{x} \ll 0$

Likelihood

Suppose we have a linear model $\mathbf{w}$ (assume bias is hidden for now)

Given a data point $(\mathbf{x}^t, y^t)$, $\mathbf{x}^t \in \mathbb{R}^d$ and $y^t \in \{-1,1\}$, the use of the sigmoidal map gives us a Rademacher PMF $\mathbb{P}[\, y \mid \mathbf{x}^t, \mathbf{w}]$

The probability that this PMF gives to the correct label i.e. $\mathbb{P}[\, y^t \mid \mathbf{x}^t, \mathbf{w}]$ is called the *likelihood* of this model with respect to this data point

It easy to show that $\mathbb{P}[\, y^t \mid \mathbf{x}^t, \mathbf{w}] = \sigma(y^t \cdot \mathbf{w}^\top \mathbf{x}^t)$

Hint*: use the fact that* $\sigma(-t) = 1 - \sigma(t)$ *and that* $y^t \in \{-1,1\}$

If we have several points $(\mathbf{x}^1, y^1), \dots, (\mathbf{x}^n, y^n)$ then we define the likelihood of w w.r.t entire dataset as $\mathbb{P}[y^1, \dots, y^n \mid \mathbf{x}^1, \dots, \mathbf{x}^n, \mathbf{w}]$

Usually we assume data points are independent so we use product rule to get

$$\mathbb{P}[y^1, \dots, y^n \mid \mathbf{x}^1, \dots, \mathbf{x}^n, \mathbf{w}] = \prod_{i=1}^n \mathbb{P}\left[\, y^i \mid \mathbf{x}^i, \mathbf{w}\right] = \prod_{i=1}^n \sigma\left(y^i \cdot \mathbf{w}^\top \mathbf{x}^i\right)$$

Li

Data might not actually be independent e.g. my visiting a website may not be independent from my friend visiting the same website if I have found an offer on that website and posted about it on social website. However, often we nevertheless assume independence to make life simple

Su

Given a data point $(\mathbf{x}^t, y^t)$, $\mathbf{x}^t \in \mathbb{R}^d$ and $y^t \in \{-1,1\}$, the use of the sigmoidal map gives us a Rademacher PMF $\mathbb{P}[\, y \mid \mathbf{x}^t, \mathbf{w}]$

The probability that this PMF gives to the correct label i.e. $\mathbb{P}[\, y^t \mid \mathbf{x}^t, \mathbf{w}]$ is called the *likelihood* of this model with respect to this data point

It easy to show that $\mathbb{P}[\, y^t \mid \mathbf{x}^t, \mathbf{w}] = \sigma(y^t \cdot \mathbf{w}^\top \mathbf{x}^t)$

Hint*: use the fact that* $\sigma(-t) = 1 - \sigma(t)$ *and that* $y^t \in \{-1,1\}$

If we have several points $(\mathbf{x}^1, y^1), \ldots, (\mathbf{x}^n, y^n)$ then we define the likelihood of w w.r.t entire dataset as $\mathbb{P}[y^1, \ldots, y^n \mid \mathbf{x}^1, \ldots, \mathbf{x}^n, \mathbf{w}]$

Usually we assume data points are independent so we use product rule to get

$$\mathbb{P}[y^1, \ldots, y^n \mid \mathbf{x}^1, \ldots, \mathbf{x}^n, \mathbf{w}] = \prod_{i=1}^n \mathbb{P}\big[\, y^i \mid \mathbf{x}^i, \mathbf{w}\big] = \prod_{i=1}^n \sigma\big(y^i \cdot \mathbf{w}^\top \mathbf{x}^i\big)$$

Maximum Likelihood

The expression $\mathbb{P}\left[\, y^i \mid \mathbf{x}^i, \mathbf{w}\right]$ tells us if the model $\mathbf{w}$ thinks the label y^i is a very likely label given the feature vector $\mathbf{x}^i$ or not likely at all!

- $\mathbb{P}[y^1, \ldots, y^n \mid \mathbf{x}^1, \ldots, \mathbf{x}^n, \mathbf{w}]$ *similarly tells us how likely does the model* $\mathbf{w}$ *think the labels* $y^1, \ldots, y^n$ *are, given the feature vectors* $\mathbf{x}^1, \ldots, \mathbf{x}^n$

Since we trust our training data as clean and representative of reality, we should look for a $\mathbf{w}$ that considers train labels to be very likely

- *E.g. in RecSys example, if* $y^t = 1$ *if customer makes a purchase and* $y^t = 0$ *otherwise then if we trust that these labels do represent reality i.e. what our customers like and dislike, then we should learn a model* $\mathbf{w}$ *accordingly*
- *Totally different story if we mistrust our data – different techniques for that*

Maximum Likelihood Estimator (**MLE**): the model which thinks observed labels are most likely $\hat{\mathbf{w}}_{\text{MLE}} = \underset{\mathbf{w} \in \mathbb{R}^d}{\arg\max} \prod_{i=1}^n \mathbb{P}\left[\, y^i \mid \mathbf{x}^i, \mathbf{w}\right]$

Logistic Regression

Suppose we learn a model as the MLE while using sigmoidal map

$$\hat{\mathbf{w}}_{\mathrm{MLE}} = \arg\max_{\mathbf{w}\in\mathbb{R}^d} \prod_{i=1}^{n} \sigma\left(y^i \cdot \mathbf{w}^\top \mathbf{x}^i\right)$$

Working with products can be numerically unstable

Since $\sigma(\cdot) \in [0,1]$*, product of several such values can be extremely small*

***Solution**: take logarithms and exploit that* $\max_x f(x) = \max_x \ln(f(x))$

$$\hat{\mathbf{w}}_{\mathrm{MLE}} = \arg\max_{\mathbf{w}\in\mathbb{R}^d} \ln\left(\prod_{i=1}^{n} \sigma\left(y^i \cdot \mathbf{w}^\top \mathbf{x}^i\right)\right)$$

$$= \arg\min_{\mathbf{w}\in\mathbb{R}^d} \sum_{i=1}^{n} \ln\left(1 + \exp\left(-y^i \cdot \mathbf{w}^\top \mathbf{x}^i\right)\right)$$

Also called *negative log-likelihood*

Thus, the logistic loss function pops out automatically when we try to learn a model that maximizes the likelihood function

Probabilistic Multiclassification

Suppose we have C classes, then for every data point we would have to output a PMF over the support $[C]$

Popular way: assign a positive score to all classes and normalize so that the scores form a proper probability distribution

***Common trick**: converting any score to a positive score – exponentiate!!*

Learn C models $\mathbf{w}^1, \ldots, \mathbf{w}^C$, given a point $(\mathbf{x}^t, y^t)$, $\mathbf{x}^t \in \mathbb{R}^d$, $y^t \in [C]$

Assign a positive score per class $\eta_c = \exp(\langle \mathbf{w}^c, \mathbf{x}^t \rangle)$

Normalize to obtain a PMF $\mathbb{P}[\, y \mid \mathbf{x}^t, \{\mathbf{w}^c\}] = \eta_y / \sum_{c=1}^{C} \eta_c$ *for any* $y \in [C]$

Likelihood in this case is $\mathbb{P}[\, y^t \mid \mathbf{x}^t, \{\mathbf{w}^c\}] = \eta_{y^t} / \sum_{c=1}^{C} \eta_c$

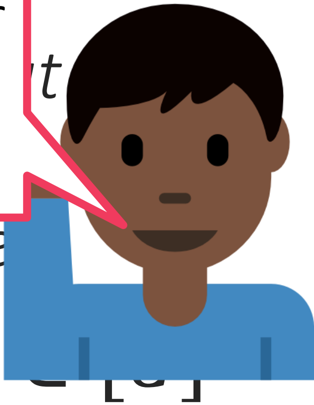
Log-likelihood in this case is $\ln\left(\eta_{y^t} / \sum_{c=1}^{C} \eta_c\right)$

Proba

Just as we had the Bernoulli distributions over the support $\{0,1\}$, if the support instead has $C > 2$ elements, then the distributions are called either *Multinoulli distributions* or *Categorical distributions*

Suppose we have C classes, then for every data point we wou o output a PMF over the support $[C]$

To specify a multinoulli distribution over C labels, we need to specify C non-negative numbers that add up to one



- ***Popular way**: assign a positive scor t scores form a proper probability d*
- ***Common trick**: converting any score to a positive score – exponentiate*

Learn C models $\mathbf{w}^1, \ldots, \mathbf{w}^C$, given a point $(\mathbf{x}^t, y^t)$, $\mathbf{x}^t \in \mathbb{R}^d$, $y^t \in [C]$

- *Assign a positive score per class* $\eta_c = \exp(\langle \mathbf{w}^c, \mathbf{x}^t \rangle)$
- *Normalize to obtain a PMF* $\mathbb{P}[\, y \mid \mathbf{x}^t, \{\mathbf{w}^c\}] = \eta_y / \sum_{c=1}^{C} \eta_c$ *for any* $y \in [C]$

Likelihood in this case is $\mathbb{P}[\, y^t \mid \mathbf{x}^t, \{\mathbf{w}^c\}] = \eta_{y^t} / \sum_{c=1}^{C} \eta_c$

Log-likelihood in this case is $\ln\left(\eta_{y^t} / \sum_{c=1}^{C} \eta_c\right)$

Softmax Regression

If we now want to learn the MLE, we would have to find

$$\left\{\hat{\mathbf{w}}^1_{\mathrm{MLE}}, \ldots, \hat{\mathbf{w}}^C_{\mathrm{MLE}}\right\} = \arg\max_{\mathbf{w}^1,\ldots,\mathbf{w}^C \in \mathbb{R}^d} \prod_{i=1}^n \mathbb{P}\left[\, y^i \mid \mathbf{x}^i, \mathbf{w}\right]$$

$$= \arg\max_{\mathbf{w}^1,\ldots,\mathbf{w}^C \in \mathbb{R}^d} \prod_{i=1}^n \eta^i_{y^t} / \sum_{c=1}^C \eta^i_c \text{ where } \eta^i_c = \exp\left(\left\langle \mathbf{w}^c, \mathbf{x}^i \right\rangle\right)$$

Using the negative log-likelihood for numerical stability

$$= \arg\min_{\mathbf{w}^1,\ldots,\mathbf{w}^C \in \mathbb{R}^d} \sum_{i=1}^n - \ln\left(\eta^i_{y^t} / \sum_{c=1}^C \eta^i_c\right)$$

Note: this is nothing but the softmax loss function we saw earlier, also known as the *cross entropy loss function*

> ***Reason for the name***: *it corresponds to something known as the* cross entropy *between the PMF given by the model and the true label of the data point*

Softm

I may find other ways to assign a PMF over $[C]$ to each data point by choosing some function other than $\exp(\cdot)$ e.g. ReLU $[t]_+$ to assign positive scores i.e. let $\eta_c = [\langle \mathbf{w}^c, \mathbf{x}^t \rangle]_+$, let $\mathbb{P}[\, y \mid \mathbf{x}^t, \{\mathbf{w}^c\}] = \eta_y / \sum_{c=1}^{C} \eta_c$ and then proceed to obtain an MLE. Something similar to this is indeed used in deep learning

$$\mathbf{W}_{\text{MLE}} \ldots \mathbf{w}^1,\ldots,\mathbf{w}^C \in \mathbb{R}^d$$

$$= \arg \max_{\mathbf{w}^1,\ldots,\mathbf{w}^C \in \mathbb{R}^d}$$

It should be noted that this is not the only way to do probabilistic multiclassification. It is just that this way is simple to understand, implement and hence popular

Using the negative log likelihood for numerical stability

However, be warned that generating a PMF using DT/kNN need not necessarily be an MLE since we have not explicitly maximized any likelihood function here

$$= \arg \min_{\mathbf{w}^1,\ldots,\mathbf{w}^C \in \mathbb{R}^d} \sum$$

Note: this is nothing but the softmax loss function we saw earlier, also

I could do also kNN or DT and invoke the “probability as proportions” interpretation to assign a test data point to a PMF that simply gives the proportion of each label in the neighbourhood/leaf of that data point!!

entropy

int

General Recipe for MLE Algorithms

Given a problem with label set $\mathcal{Y}$, find a way to map data features $\mathbf{x}$ to PMFs $\mathbb{P}[\cdot \mid \mathbf{x}, \mathbf{m}]$ with support $\mathcal{Y}$

The notation $\mathbf{m}$ *captures parameters in the model (e.g. vectors, bias terms)*

For binary classification, $\mathcal{Y} = \{-1,1\}$ *and* $\mathbf{m} = \mathbf{w}$

For multiclassification, $\mathcal{Y} = [C]$ *and* $\mathbf{m} = \{\mathbf{w}^1, \dots, \mathbf{w}^C\}$

The function $\mathbb{P}[\cdot \mid \mathbf{x}, \mathbf{m}]$ is often called the *likelihood function*

The function $-\ln \mathbb{P}[\cdot \mid \mathbf{x}, \mathbf{m}]$ called *negative log likelihood function*

Given data $\left\{\left(\mathbf{x}^i, y^i\right)\right\}_{i=1}^{n}$, find the model parameters that maximize likelihood function i.e. think that the training labels are very likely

$$\hat{\mathbf{m}}_{\text{MLE}} = \arg\min_{\mathbf{m}} \sum_{i=1}^{n} -\ln \mathbb{P}\left[y^i \mid \mathbf{x}^i, \mathbf{m}\right]$$

Probabilistic Regression??

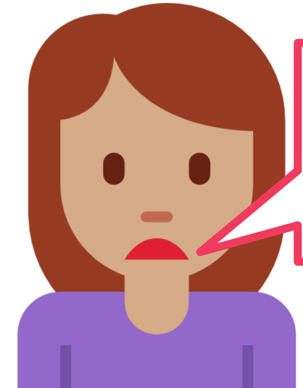
To perform probabilistic binary classification, our ML model was told to output a PMF over $\{-1,1\}$ for every data point $\mathbf{x}$

We could specify this PMF by just specifying 2 real numbers that add up to 1

To perform probabilistic multiclassification with C classes, our ML model was told to output a PMF over $[C]$ for every data point $\mathbf{x}$

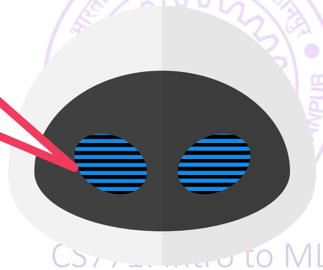
We could specify this PMF by just specifying C *real numbers that add up to* 1

However, if we want to perform probabilistic regression over $\mathbb{R}$, if we go the PMF route, we would need to specify infinitely many numbers



We can't do that. Is there a way out?

Indeed, this is not the correct way to specify a probability distribution over a continuum such as $\mathbb{R}$. Let us see the proper way to do this



Continuous Random Variables

These are r.v. that can take infinitely many possibly values that are not discrete but continuous i.e. support is $\mathbb{R}$ or some subset of $\mathbb{R}$

The notion of PMF which tells us with what probability does the r.v. take this value or that value does not make sense when support is continuous

Instead of PMF, we use a PDF (probability density function) in such cases

Consider a r.v. X which takes value in the interval $[-2,2]$

***Warning**: $[-2,2]$ is very different from $\{-2,2\}$*

To specify the probability distribution of X we use a PDF $f_X: S_X \to \mathbb{R}_+$

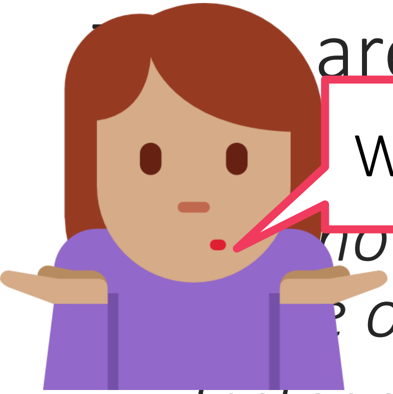
The PDF takes a value in support of r.v. and spits out a non-negative number

***Note**: f_X can take values greater than 1 as well but they must not be negative*

***Interpretation**: For any $x \in S_X$, the value $f_X(x)$ tells us how likely is X to take a value* around *x i.e. for some teeny $\delta > 0$, we have*

$$\mathbb{P}\big[X \in [x-\delta, x+\delta]\big] \approx f_X(x) \cdot 2\delta$$

Continuous Random Variables



Why ≈ why not = ?

… are r.v. that can take infinitely many possibly values that are not … s i.e. support is ℝ or some subset of ℝ

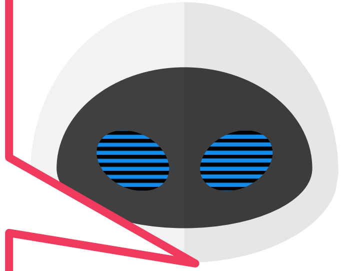
… notion of PMF which tells us with what probability does the r.v. take this … or that value does not make sense when support is continuous

Instead of PMF, we use a PDF (probability density function) in such cases

Co…

We do have an exact formula too $\mathbb{P}\big[X \in [x-\delta, x+\delta]\big] = \int_{x-\delta}^{x+\delta} f_X(t)dt$

In general, if the r.v. X has a PDF f_X, then for any interval within its support $[a, b] \subseteq S_X$, we have $\mathbb{P}\big[X \in [a, b]\big] = \int_a^b f_X(t)dt$



… $\mathbb{R}_+$

The PD… … ive number

… *t be negative*

… *is X to take a*

However, if the interval is "small", then we can often get a good and simple approximation $\mathbb{P}\big[X \in [a, b]\big] \approx f_X(c) \cdot (b - a)$ where $c = \frac{(b+a)}{2}$. How small is "small" enough depends on the PDF f_X

$$\mathbb{P}\big[X \in [x-\delta, x+\delta]\big] \approx f_X(x) \cdot 2\delta$$

Continuous R.V.s– the Rules Revisited

PDF f_X of a r.v. X satisfies $f_X(x) \geq 0$ for all $x \in S_X$ and $\int_{S_X} f_X(t)dt = 1$

Expectation of a continuous R.V. X is $\mathbb{E}X \triangleq \int_{S_X} t \cdot f_X(t)dt$

***LOTUS**:* $\mathbb{E}[g(X)] = \int_{S_X} g(t) \cdot f_X(t)dt$

Variance of a continuous R.V. X is $\mathbb{V}X \triangleq \int_{S_X} (t - \mathbb{E}X)^2 \cdot f_X(t)dt$

$\mathbb{V}X = \mathbb{E}[X^2] - (\mathbb{E}X)^2 = \int_{S_X} t^2 \cdot f_X(t)dt - (\mathbb{E}X)^2$

Joint PDFs make sense too $f_{X,Y}$: suppose $[a, b] \subseteq S_X$ and $[p, q] \subseteq S_Y$

$$\mathbb{P}\big[X \in [a, b], Y \in [p, q]\big] = \int_p^q \int_a^b f_{X,Y}(s, t)\, ds\, dt$$

They make sense even if X is continuous and Y is discrete and vice versa

Details of these constructions, however, are beyond the scope of CS771

Continuous R.V.s– the Rules Revisited

Marginal probabilities continue to make sense

$$f_X(x) = \int_{S_Y} f_{X,Y}(x,t)dt$$

Conditional probabilities also make sense

When X, Y are both are continuous $\mathbb{P}\left[X \in [a,b] \mid Y \in [p,q]\right]$

$\mathbb{P}\left[X \in [a,b] \mid Y \in [p,q]\right] \triangleq \mathbb{P}\left[X \in [a,b], Y \in [p,q]\right] / \mathbb{P}\left[Y \in [p,q]\right]$

Actually, even $\mathbb{P}[X \in [a,b] \mid Y = r]$ makes sense in this case – details beyond CS771

When X is discrete but Y is continuous $\mathbb{P}\left[X = c \mid Y \in [p,q]\right]$

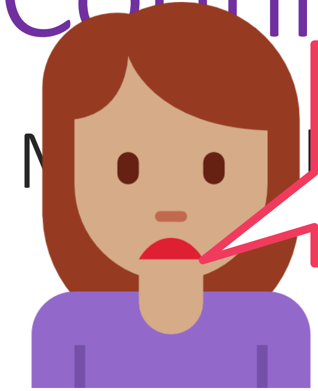
Actually, even $\mathbb{P}[X = c \mid Y = r]$ makes sense in this case – details beyond CS771

When X is continuous but Y is discrete $\mathbb{P}[X \in [a,b] \mid Y = r]$

Conditional expectations, (co)variances also defined similarly

Tricky to define in some cases as above – details beyond scope of CS771

Continuous R.V.s– the Rules Revisited



Wait! If Y is continuous, even if $r \in S_Y$, what is $\mathbb{P}[Y = r]$?

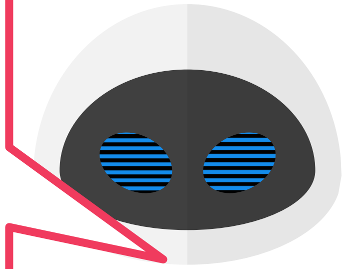
N to make sense

$f_X(x) = \int f_{X,Y}(x,t)dt$

Conditional pro

When X, Y are

In general $\mathbb{P}[Y = r] = 0$ in such cases and you would be right to suspect a divide-by-zero problem here. However, it is possible to still define $\mathbb{P}[X \in [a, b] \mid Y = r]$ using limits or a powerful technique called the *Radon-Nikodym derivative*



$\mathbb{P}\left[X \in [a, b] \mid Y \in [p, q]\right] \triangleq \mathbb{P}\left[X \in [a, b], Y \in [p, q]\right] / \mathbb{P}\left[Y \in [p, q]\right]$

Actually, even $\mathbb{P}[X \in [a, b] \mid Y = r]$ makes sense in this case – details beyond CS771

When X is discrete but Y is continuous $\mathbb{P}\left[X = c \mid Y \in [p, q]\right]$

Actually, even $\mathbb{P}[X = c \mid Y = r]$ makes sense in this case – details beyond CS771

When X is continuous but Y is discrete $\mathbb{P}[X \in [a, b] \mid Y = r]$

Conditional expectations, (co)variances also defined similarly

Tricky to define in some cases as above – details beyond scope of CS771

Continuous R.V.s– the Rules Revisited

Rules of Probability: All rules Sum, Product, Chain, Bayes, Complement, Union continue to hold

If X, Y are independent continuous R.V. then $f_{X,Y}(x, y) = f_X(x) \cdot f_Y(y)$

For independent continuous R.V. we continue to have

$$\mathbb{P}\left[X \in [a, b] \mid Y \in [p, q]\right] = \mathbb{P}\left[X \in [a, b]\right]$$
$$\mathbb{P}[X \in [a, b] \mid Y \in r] = \mathbb{P}\left[X \in [a, b]\right]$$

Rules of Expectation: All rules Linearity, Scaling, Product still hold

Rules of (co)Variance: All rules Constant, Scaling, Shift, Sum still hold