Maximum Likelihood: The theoretical basis for our objective functions

Garrison W. Cottrell
Gary's Unbelievable Research Unit (GURU)
Computer Science and Engineering Department
Temporal Dynamics of Learning Center
Institute for Neural Computation
UCSD

Reading

- This lecture is based on Bishop Chapter 6, sections 6.1, 6.7, and 6.9.

This lecture

- What is maximum likelihood?
- How does it lead to Sum Squared Error for regression?
- How does it lead to cross-entropy for logistic regression?
- How does it lead to cross-entropy for multinomial regression?

- Next lecture: Other approaches to objective functions
- Summary

This lecture

- ***What is maximum likelihood?***
- How does it lead to Sum Squared Error for regression?
- How does it lead to cross-entropy for logistic regression?
- How does it lead to cross-entropy for multinomial regression?

Maximum Likelihood

- Main Idea:
- What we ***really*** want to know is, given the data D, ***which parameters W of our model are most likely?***
- That is, we want the parameters W that *maximize* $P(W|D)$.
- This is hard.
- But recall Bayes' Rule: $$P(W \mid D) = \frac{P(D \mid W)P(W)}{P(D)}$$

Maximum Likelihood

- Recall Bayes' Rule:

$$P(W \mid D) = \frac{P(D \mid W)P(W)}{P(D)} = \frac{\text{Likelihood*Prior}}{\text{normalizing constant}}$$

- Note that:
  - The probability of the data is just a normalizing constant.
  - We have no reason, *a priori,* to assume some *W*'s are better than others, so we assume that without any data, the prior is flat – all *W*'s are equally likely.
  - So $P(W)$, the prior, is a constant.

Maximum Likelihood

- So, given the assumption of a *uniform prior*:

$$P(W \mid D) = \frac{P(D \mid W)P(W)}{P(D)} = \frac{\text{Likelihood*Prior}}{\text{normalizing constant}}$$

$$\max_W P(W \mid D) = \max_W \frac{P(D \mid W)P(W)}{P(D)} = \max_W P(D \mid W)$$

- Hence the name, *Maximum Likelihood*.
- Specifically, we want our network to make the observed data as likely as possible.
- How we model the distribution of the data is key.

Maximum Likelihood

- Example: Gaussian distributions

$$p(x \mid \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

- Now, assuming the data points are independently identically distributed (iid), the *Likelihood of the data is:*

$$\mathcal{L} = \prod_{n=1}^{N} p(x^n) = \frac{1}{\sqrt{2\pi\sigma^2}^N} \prod_{p=1}^{N} e^{-\frac{(x^n-\mu)^2}{2\sigma^2}}$$

Maximum Likelihood

- Finally, Maximum Likelihood says we should pick μ and σ such that:

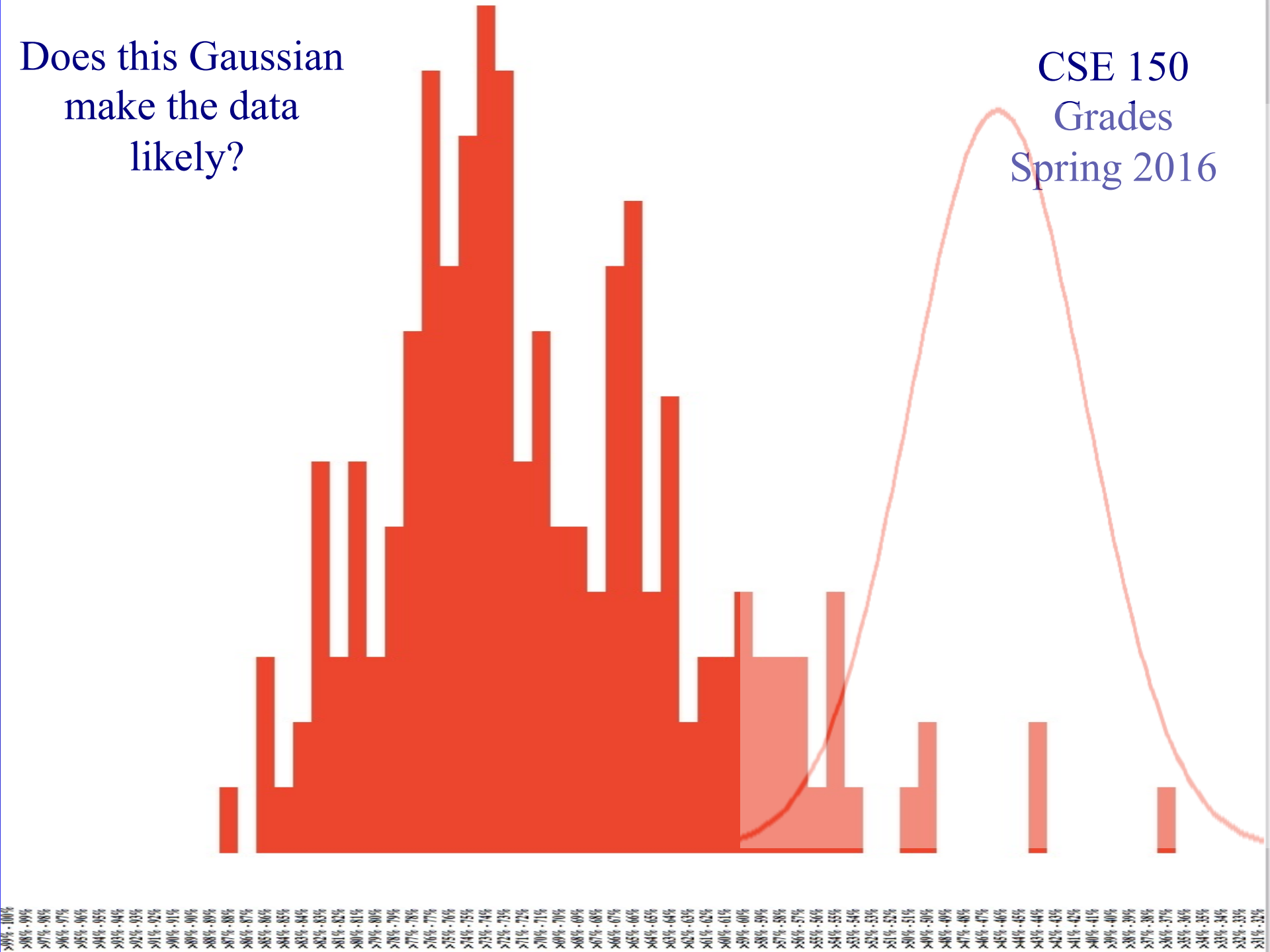
$$(\mu,\sigma) = \arg\max_{\mu,\sigma} \mathcal{L}$$

$$= \arg\max_{\mu,\sigma} \prod_{n=1}^{N} p(x^n) = \arg\max_{\mu,\sigma} \frac{1}{\sqrt{2\pi\sigma^2}^N} \prod_{p=1}^{N} e^{-\frac{(x^n-\mu)^2}{2\sigma^2}}$$

- Again, in English: We should choose the parameters that maximize the likelihood of the data.
- How do we do that?

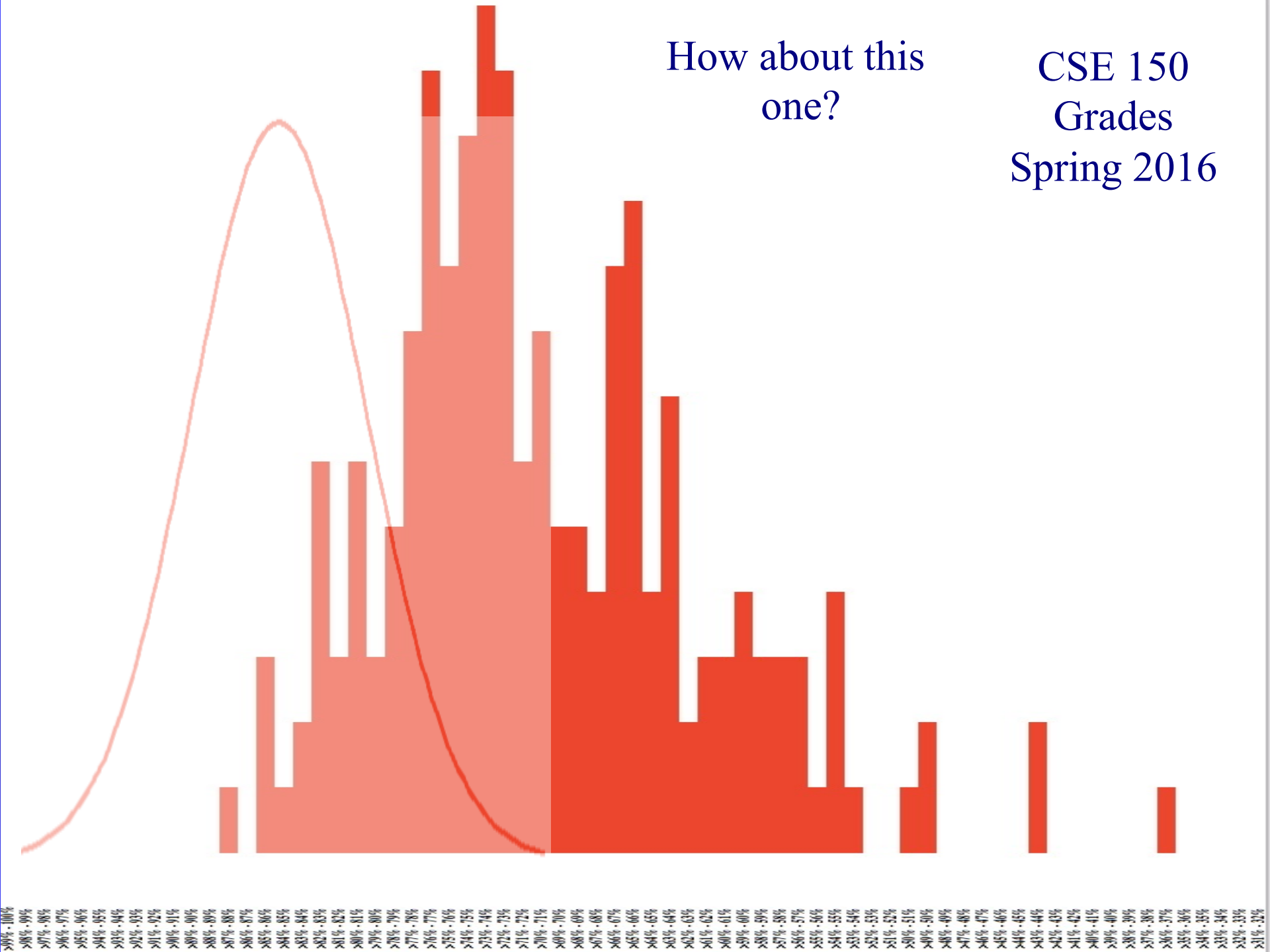
Does this Gaussian make the data likely?

CSE 150
Grades
Spring 2016



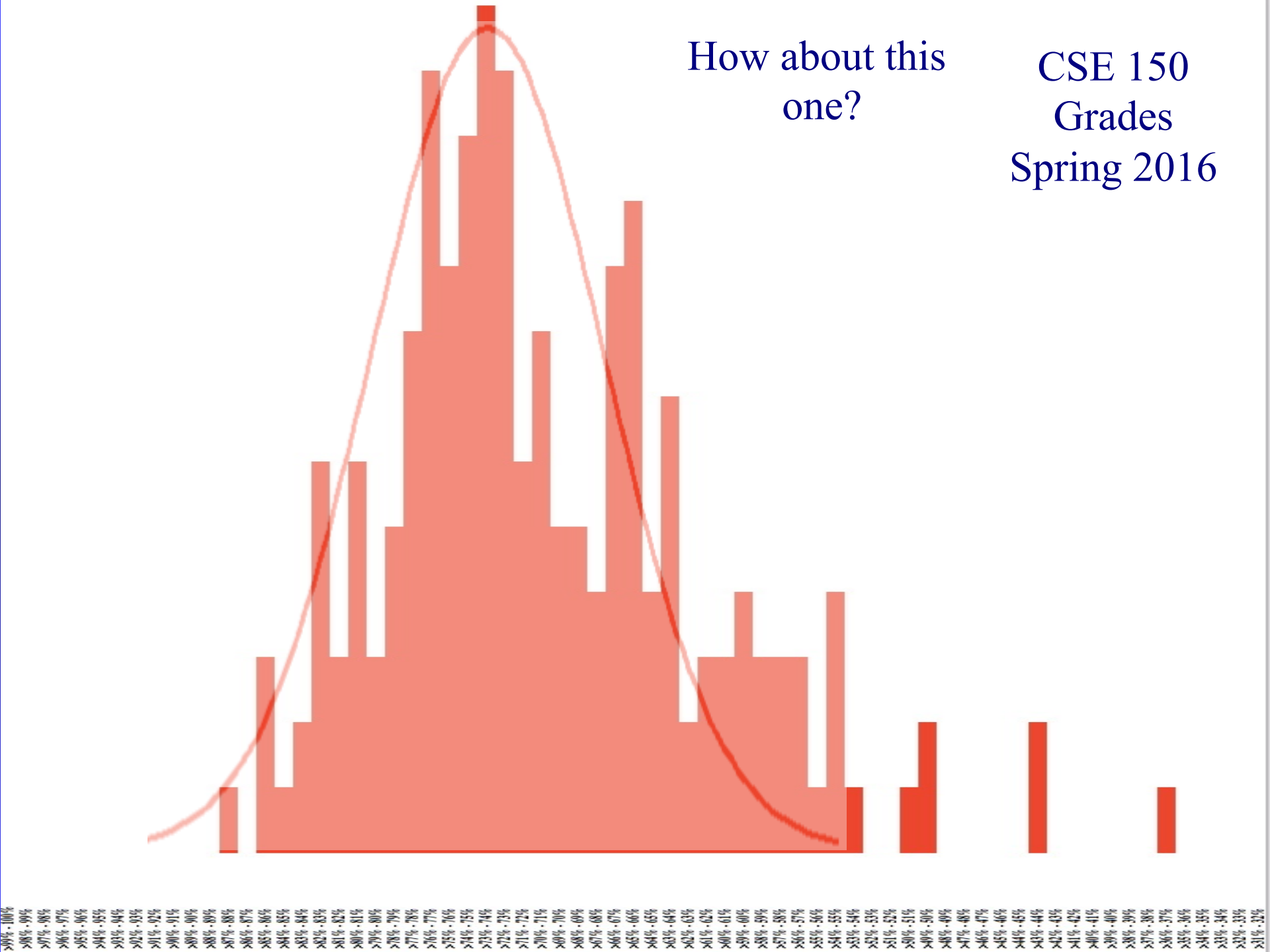
How about this one?

CSE 150
Grades
Spring 2016



How about this one?

CSE 150
Grades
Spring 2016



How do we find the ML parameters?

- Usually, it is easier to minimize the negative log likelihood, i.e.,

$$
\begin{aligned}
(\mu,\sigma) &= \arg\max_{\mu,\sigma} \mathcal{L} \\
&= \arg\max_{\mu,\sigma} \ln \prod_{n=1}^{N} p(x^n) \\
&= \arg\min_{\mu,\sigma} \; -\ln \prod_{n=1}^{N} p(x^n) \\
&= \arg\min_{\mu,\sigma} \; -\sum_{n=1}^{N} \ln p(x^n)
\end{aligned}
$$

How do we find the ML parameters?

- First, a little terminology:

$$L = \prod_{n=1}^{N} p(x^n) \qquad \text{"Likelihood"}$$

$$\ln L = \ln \prod_{n=1}^{N} p(x^n) \qquad \text{"Log Likelihood"}$$

$$-\ln L = -\ln \prod_{n=1}^{N} p(x^n) \qquad \text{"Error"}$$

- The negative log likelihood is called the *error*.

How do we find the ML parameters?

- These are all equivalent:

$$(\mu,\sigma) = \arg\max_{\mu,\sigma} \ln \prod_{n=1}^{N} p(x^n)$$

$$= \arg\min_{\mu,\sigma} -\ln \prod_{n=1}^{N} p(x^n)$$

$$= \arg\min_{\mu,\sigma} -\sum_{n=1}^{N} \ln p(x^n)$$

$$= \arg\min_{\mu,\sigma} -\sum_{n=1}^{N} \frac{-(x^n-\mu)^2}{2\sigma^2} - N \ln \sqrt{2\pi\sigma^2}$$

- Where the last bit assumes a gaussian

How do we find the ML parameters?

- These are all equivalent:

$$(\mu,\sigma) = \arg\max_{\mu,\sigma} \ln \prod_{n=1}^{N} p(x^n)$$

$$= \arg\min_{\mu,\sigma} \; -\ln \prod_{n=1}^{N} p(x^n)$$

$$= \arg\min_{\mu,\sigma} \; -\sum_{n=1}^{N} \ln p(x^n)$$

$$= \arg\min_{\mu,\sigma} \; -\sum_{n=1}^{N} \frac{-(x^n-\mu)^2}{2\sigma^2} - N \ln \sqrt{2\pi\sigma^2}$$

- Exercise for the reader: Show that this leads to setting μ to the empirical mean!

How do we find the ML parameters?

- Exercise for the reader: Show that this leads to setting μ to the empirical mean!

- *Hint:* take the derivative with respect to μ, set it to zero, and solve.

Clicker time!

Q1: Maximum Likelihood Estimation

a. Maximizes parameters of a distribution

b. Maximizes the loss function

c. Maximizes the probability of the data

d. Maximizes the log likelihood of the loss

e. Minimizes the probability of the data

Clicker time!

Q1: Maximum Likelihood Estimation

a. Maximizes parameters of a distribution

b. Maximizes the loss function

c. Maximizes the probability of the data

d. Maximizes the log likelihood of the loss

e. Minimizes the probability of the data

Clicker time!

Q2: The following is an expression for the results of MLE for a one-D Gaussian:

A. $(\mu,\sigma) = \arg\max_{\mu,\sigma} \ln \prod_{n=1}^{N} p(x^n)$

B. $(\mu,\sigma) = \arg\min_{\mu,\sigma} -\ln \prod_{n=1}^{N} p(x^n)$

C. $(\mu,\sigma) = \arg\max_{\mu,\sigma} \prod_{n=1}^{N} p(x^n)$

D. A & C

E. You can't fool me, these are all the same!

Clicker time!

Q2: The following is an expression for the results of MLE for a one-D Gaussian:

A. $(\mu,\sigma) = \arg\max_{\mu,\sigma} \ln \prod_{n=1}^{N} p(x^n)$

B. $(\mu,\sigma) = \arg\min_{\mu,\sigma} -\ln \prod_{n=1}^{N} p(x^n)$

C. $(\mu,\sigma) = \arg\max_{\mu,\sigma} \prod_{n=1}^{N} p(x^n)$

D. A & C

E. You can't fool me, these are all the same!

This lecture

- What is maximum likelihood?
- ***How does it lead to Sum Squared Error for regression?***
- How does it lead to cross-entropy for logistic regression?
- How does it lead to cross-entropy for multinomial regression?

Modeling input-output data

- In the previous slides, we just assumed one-dimensional data.
- In applications of neural networks (and regression), we have inputs and outputs, i.e., now the likelihood looks like:

$$= \prod_{n=1}^{N} p(x^n, t^n) = \prod_{n=1}^{N} p(t^n \mid x^n) p(x^n)$$

- And taking the negative log, we get:

$$-\ln L = -\sum_{n=1}^{N} \left(\ln p(t^n \mid x^n) + \ln p(x^n) \right)$$

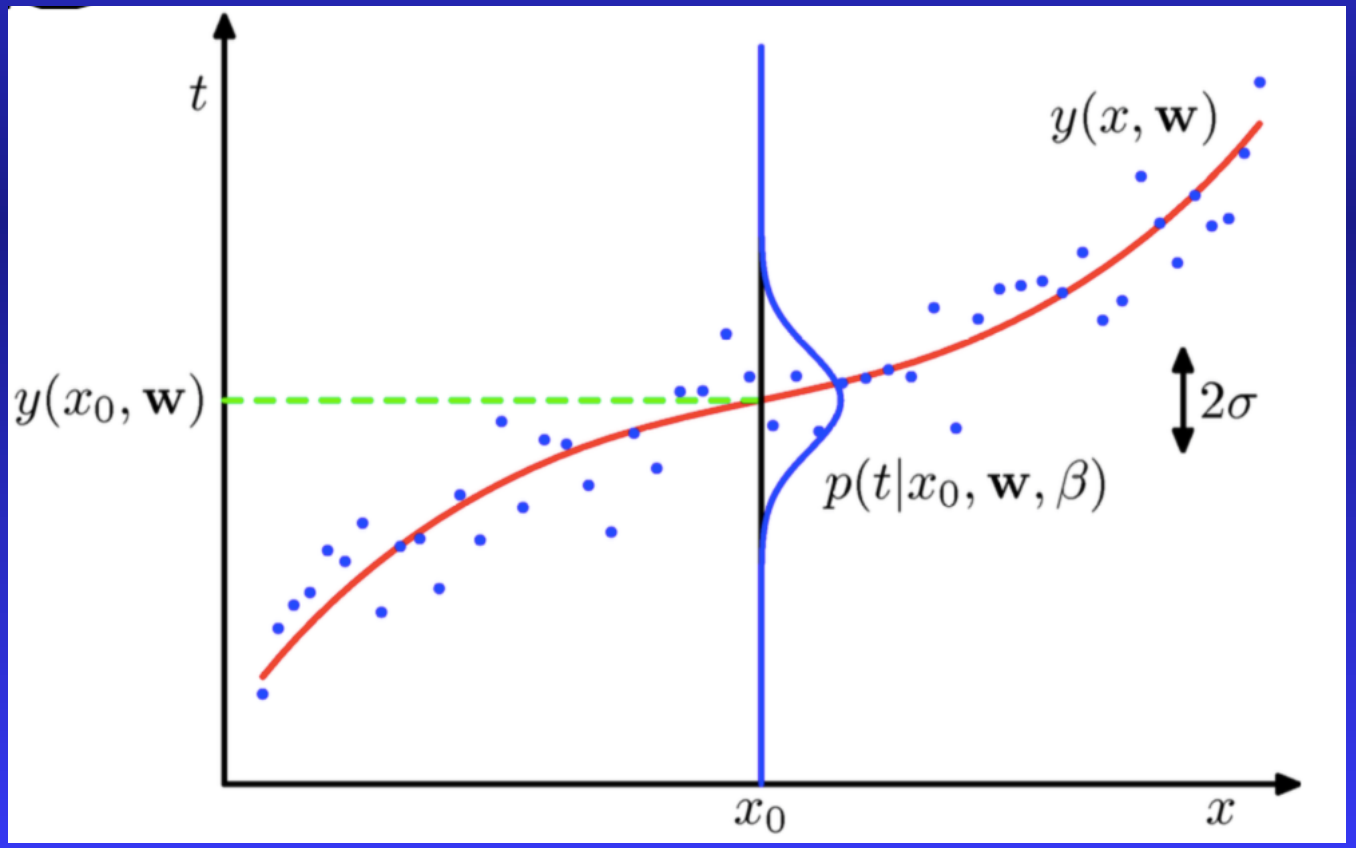
Modeling input-output data

$$-\ln L = -\sum_{n=1}^{N}\left(\ln p(t^n \mid x^n) + \ln p(x^n)\right)$$

- Since we are modeling the mapping from x to t, when minimizing this with respect to our parameters, the second term doesn't change, so we can drop it.

ML and regression

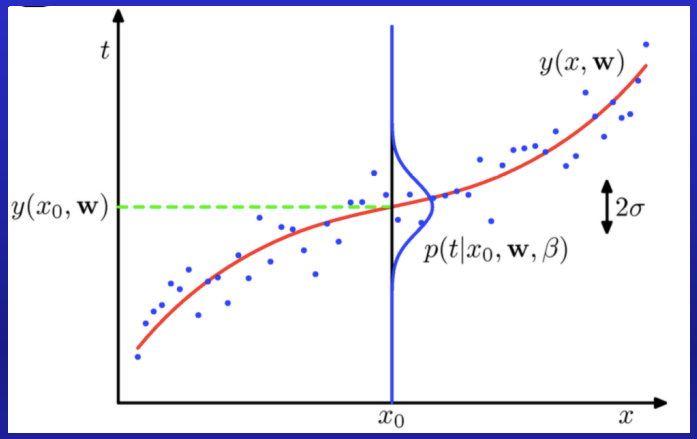
- Main idea: ***Assume*** gaussian noise in the targets t^n (don't take the expressions in the figure here too seriously – not in our notation!)



ML and regression

- Main idea: ***Assume*** gaussian distribution of the targets
- In other words:

$$p(t^n \mid x^n) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(t^n - h(x^n))^2}{2\sigma^2}}$$



- Here, we assume that there is an underlying *deterministic* function *h* (the red line), with some additive, 0 mean gaussian noise ε:

$$t^n = h(x^n) + \varepsilon \quad \text{or,} \quad \varepsilon = t^n - h(x^n)$$

ML and regression

So, the likelihood of *all* of the data looks like:

$$L = \prod_{n=1}^{N} p(t^n \mid x^n) = \frac{1}{\sqrt{2\pi\sigma^2}^N} \prod_{p=1}^{N} e^{-\frac{(t^n - y(x^n;w))^2}{2\sigma^2}}$$

Where we have replaced the deterministic function $h(x^n)$ by our model of it, $y(x^n;w)$ (this notation emphasizes that the network is parameterized by the weights, w.)

And the negative log likelihood, or error is:

$$E = -\ln \frac{1}{\sqrt{2\pi\sigma^2}^N} \prod_{p=1}^{N} e^{-\frac{(t^n - y(x^n;w))^2}{2\sigma^2}}$$

ML and regression

The negative log likelihood, or error is:

$$E = -\ln \frac{1}{\sqrt{2\pi\sigma^2}^N} \prod_{p=1}^{N} e^{-\frac{(t^n - y(x^n;w))^2}{2\sigma^2}}$$

$$= \frac{1}{2\sigma^2} \sum_{n=1}^{N} (t^n - y(x^n;w))^2 + \ln\left(\sqrt{2\pi\sigma^2}^N\right)$$

Removing the second term, which is constant with respect to the weights, and the factor of $1/\sigma^2$, which doesn't affect the minimum, we get:

$$\frac{1}{2} \sum_{n=1}^{N} (t^n - y(x^n;w))^2$$

ML and regression

To recap:

When doing regression, if we

1) assume the targets are Gaussian distributed, and
2) maximize the likelihood of the data by minimizing the negative log likelihood of the data,

We find that we need to minimize the Sum Squared Error!

This lecture

- What is maximum likelihood?
- How does it lead to Sum Squared Error for regression?
- ***How does it lead to cross-entropy for logistic regression?***
- How does it lead to cross-entropy for multinomial regression?

Logistic regression should output the probability of category 1

- Here, we want the network to produce the probability that the input is in category 1:

$$y(x^n)=P(C_1|x^n)$$

Note: This is basically categorization!

Modeling two categories is like modeling a coin flip

- Here, we want the network to produce the probability that the input is in category 1:

$$y(x^n)=P(\mathrm{C}_1|x^n)$$

This will work if we use $t^n = 1$ for category 1, and $t^n = 0$ for category 2.

- What is the likelihood of the data now?

$$\mathcal{L} = \prod_{n=1}^{N} p(t^n \mid x^n) = \prod_{n=1}^{N} (y^n)^{t^n} (1-y^n)^{(1-t^n)}$$

Modeling two categories is like modeling a coin flip

- What is the likelihood of the data now?
- Again, the network is modeling the *probability distribution* of the data.
- By assumption last time, it was a Gaussian distribution; now, it is a *Bernoulli distribution* (same as for a coin flip!).
- $\mathcal{L} = \prod_{n=1}^{N} p(t^n \mid x^n) = \prod_{n=1}^{N} (y^n)^{t^n} (1 - y^n)^{(1-t^n)}$

Modeling two categories is like modeling a coin flip

$$\mathcal{L} = \prod_{n=1}^{N} p(t^n \mid x^n) = \prod_{n=1}^{N} (y^n)^{t^n} (1-y^n)^{(1-t^n)}$$

$$-\ln \mathcal{L} = -\ln \prod_{n=1}^{N} (y^n)^{t^n} (1-y^n)^{(1-t^n)}$$

$$= -\sum_{n=1}^{N} \ln (y^n)^{t^n} (1-y^n)^{(1-t^n)}$$

$$= -\sum_{n=1}^{N} t^n \ln(y^n) + (1-t^n)\ln(1-y^n)$$

Cross entropy error!

ML and two-category classification

To recap:

When doing two-category classification (AKA logistic regression), if we

1) assume the targets are Bernoulli distributed, and
2) maximize the likelihood of the data by minimizing the negative log likelihood,

we find that we need to minimize the Cross Entropy Error!

This lecture

- What is maximum likelihood?
- How does it lead to Sum Squared Error for regression?
- How does it lead to cross-entropy for logistic regression?
- ***How does it lead to cross-entropy for multinomial regression?***

Multinomial Regression

- Now we have more than two categories – so we need multiple outputs.
- Let's assume there are c outputs, one for each category.
- Now we want: $P(C_k|x^n) = y_k(x^n)$ – the k^{th} output is the probability that the input is in category k...
- And $t_k^n = 1$ if example n is from category k, and 0 otherwise (one hot encoding)
- *How to write the likelihood?*

Multinomial Regression

- Now: $P(C_k|x^n) = y_k(x^n)$ and $t_k^n = 1$ if example n is from category k, and 0 otherwise (one hot encoding)
- We can therefore write the probability of *one pattern* as:

$$p(t^n \mid x^n) = \prod_{k=1}^{c} (y_k^n)^{t_k^n}$$

- E.g., suppose the input is from category 3 out of 4 total categories. Then:

$$p(t^n \mid x^n) = \prod_{k=1}^{4} (y_k^n)^{t_k^n}$$

$$= (y_1^n)^0 (y_2^n)^0 (y_3^n)^1 (y_4^n)^0$$

$$= y_3^n$$

Multinomial Regression

- Now: $P(C_k|x^n)=y_k(x^n)$ and $t_k^n = 1$ if example n is from category k, and 0 otherwise (one hot encoding)
- We can therefore write the probability of *one pattern* as:

$$p(t^n \mid x^n) = \prod_{k=1}^{c} (y_k^n)^{t_k^n}$$

- So we can write the entire likelihood as:

$$\mathcal{L} = \prod_{n=1}^{N} p(t^n \mid x^n) = \prod_{n=1}^{N} \prod_{k=1}^{c} (y_k^n)^{t_k^n}$$

Multinomial Regression

So we can write the entire likelihood as:

$$\mathcal{L} = \prod_{n=1}^{N} p(t^n \mid x^n) = \prod_{n=1}^{N} \prod_{k=1}^{c} (y_k^n)^{t_k^n}$$

And the negative log likelihood as:

$$-\ln \mathcal{L} = -\ln \prod_{n=1}^{N} \prod_{k=1}^{c} (y_k^n)^{t_k^n} = -\sum_{n=1}^{N} \sum_{k=1}^{c} t_k^n \ln y_k^n$$

ML and c-way classification

To recap:

When doing c-way classification (AKA multinomial regression), if we

1) assume the targets are multinomially distributed, and
2) maximize the likelihood of the data by minimizing the negative log likelihood,

we find that we need to minimize the Cross Entropy Error!

Next lecture

- What is maximum likelihood?
- How does it lead to Sum Squared Error for regression?
- How does it lead to cross-entropy for logistic regression?
- How does it lead to cross-entropy for multinomial regression?
- ***Other approaches to objective functions***
- ***Summary***
- ***Some tricks of the trade***

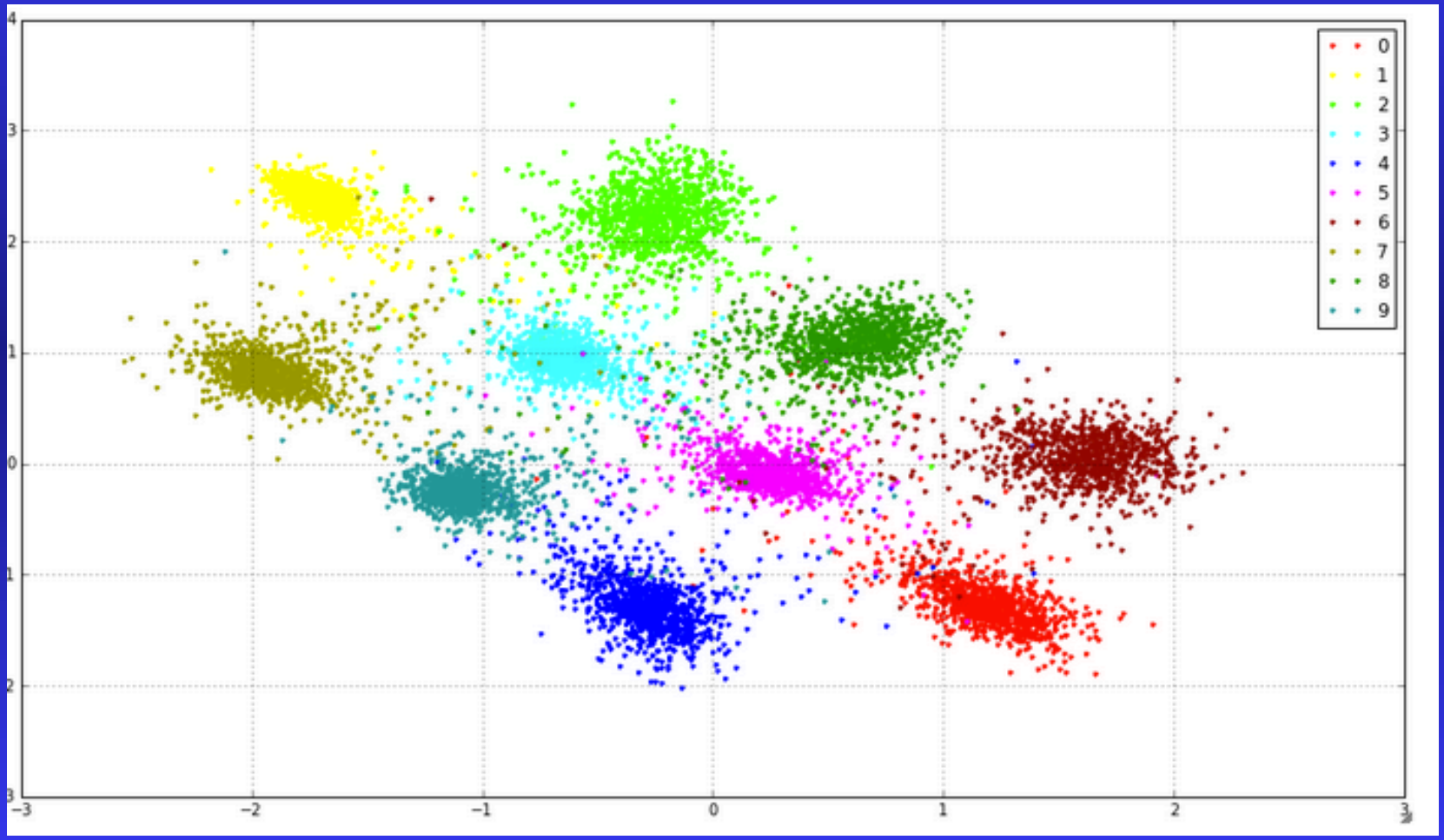
Another approach to objective functions

- Sometimes we aren't trying to fit a distribution!
- Suppose we have two networks, with output vectors of arbitrary but equal dimension – but their meaning is unspecified.
- This output vector can represent some arbitrary transformation of the input vector.
- E.g., we may want all pictures of the same person to have the same representation, while all pictures of different people to have different representations.

Another approach to objective functions

- E.g., we may want all pictures of the same person to have the same representation, while all pictures of different people to have different representations.
- This is *clustering*
- For example, supposed we only knew some inputs are from the same category, and some inputs are from different categories.
- The next slide is an example of the result of this using MNIST

MNIST clusters in 2 dimensions (the network has two outputs)



What kind of objective function could we use?

We want representations of things from the same category to be pushed together, and representations of things from different categories to be pushed apart.

We could take the two outputs, and minimize the distance between them when they are the same (using, e.g., squared error), and put a negative sign on that when they are different (push them apart).

But other ways to "push them apart" are possible.

These kinds of objective functions are sometimes called *energy functions*.

We will see more of this later in the quarter.

Summary

- *Maximum Likelihood* is a kind of "meta-objective" function:
  - overall, you want to adjust your parameters to maximize the likelihood of the data you see.
  - But the particular form of the function you get varies with the kind of distribution you are modeling
- A gaussian distribution leads to SSE
- A bernoulli distribution leads to cross entropy
- A multinomial distribution, with the right target coding, leads to cross entropy.
- Other kinds of objective functions are possible, depending on what you are trying to do.