6.S079 MACHINE LEARNING 3

MARCH 12, 2024
MIKE CAFARELLA

THANKS TO TIM KRASKA FOR SLIDES
AGENDA

1. More Supervised Learning
2. Bias/Variance
3. Cross-Validation
4. Quality Metrics
5. Embeddings
AGENDA

1. More Supervised Learning
2. Bias/Variance
3. Cross-Validation
4. Quality Metrics
5. Embeddings
## MACHINE LEARNING PROBLEMS

<table>
<thead>
<tr>
<th>Supervised Learning</th>
<th>Unsupervised Learning</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Discrete</strong></td>
<td><strong>Continuous</strong></td>
</tr>
<tr>
<td>classification or categorization</td>
<td>clustering</td>
</tr>
<tr>
<td>regression</td>
<td>dimensionality reduction</td>
</tr>
</tbody>
</table>
The maximum margin linear classifier is the linear classifier with the maximum margin. This is the simplest kind of SVM (Called an LSVM).
SUPPOSE WE’RE IN 1-DIMENSION

What would SVMs do with this data?
SUPPOSE WE’RE IN 1-DIMENSION

$x=0$

Positive “plane”

Negative “plane”
HARDER 1-DIMENSIONAL DATASET

$\chi=0$
HARDER 1-DIMENSIONAL DATASET

Permitting non-linear basis functions

\[ z_k = (x_k, x_k^2) \]

Slides from Andrew W. Moore
Because the kernel seems to be the object of interest, and not the mapping the kernel for every pair of data points, thus using same letter both for the function and its matrix. We don't even need to know how to compute the modified inner product. And because "modified inner product" is a long name, leaving the mapping completely implicit.

What happens in our example when we first map our data via some function which is the equation of an ellipse. So that's interesting, we can use our linear algorithm on a transformed space using the form

$$K(x, y) = \langle x, y \rangle + b,$$

where$$ K$$ is the kernel, of the data, in the form

$$++\quad \cdot\cdot\cdot$$

and if we try to linearly separate the mapped data, our decision boundaries will be hyperplanes in space using the form

$$K(x, y) = \langle x, y \rangle + b,$$

where$$ K$$ is the kernel, of the data, in the form

$$++\quad \cdot\cdot\cdot$$

However when people refer to SVM, they generally refer to the enhanced and more general version that we obtained using several of them.

Now, at the optimum, pick an $$\alpha_i$$ such that

$$\alpha_i^* > 0, i.e. as a function of x, y$$

Therefore it will be on the boundary, where $$\alpha_i^* > 0$$ and computing the inner product, we can do it in one operation, $$\langle x, y \rangle + b, \forall x, y \in \mathbb{R}^n$$

Clearly, the data on the left in figure 1 is not linearly separable. Yet if we map it to a three-dimensional space using the form

$$K(x, y) = \langle x, y \rangle + b,$$

where$$ K$$ is the kernel, of the data, in the form

$$++\quad \cdot\cdot\cdot$$

The data is now linearly separable.
THE KERNEL TRICK

\[ \phi : \mathbb{R}^2 \rightarrow \mathbb{R}^3 \]
\[ (x_1, x_2) \mapsto (z_1, z_2, z_3) = (x_1^2, \sqrt{2}x_1x_2, x_2^2) \]

SVM with a polynomial Kernel visualization

Created by: Udi Aharoni

https://www.youtube.com/watch?v=3liCbRZPrZA
How would you draw the expected decision boundary for
- Random Forest
- SVM w/ kernel and regularization
- 1-KNN
The decision boundary looks like the one of:

a) Random Forest
b) SVM w/ kernel and regularization
c) 1-KNN
The decision boundary looks like the one of:
a) Random Forest
b) SVM w/ kernel and regularization
c) 1-KNN
The decision boundary looks like the one of:

a) Random Forest
b) SVM w/ kernel and regularization
c) 1-KNN
If the training data is not linearly separable, slack variables $\xi_i$ (a regularization parameter) can be added to allow misclassification of difficult or noisy examples.

Still, try to minimize training set errors, and to place hyperplane “far” from each class (large margin).

“Overfitting” means memorizing the dataset instead of generalizing.

Regularization exists to prevent overfitting in the face of difficult/noisy data.
THE IMPACT OF REGULARIZATION

No regularization

Right amount

Too much
# MACHINE LEARNING PROBLEMS

<table>
<thead>
<tr>
<th>Supervised Learning</th>
<th>Unsupervised Learning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discrete</td>
<td>Clustering</td>
</tr>
<tr>
<td>Classification or categorization</td>
<td></td>
</tr>
<tr>
<td>Continuous</td>
<td>Regression</td>
</tr>
<tr>
<td></td>
<td>Dimensionality reduction</td>
</tr>
</tbody>
</table>
LINEAR REGRESSION

Datapoints
Regression
POLYNOMIAL REGRESSION

\[ h_\theta(x) = \theta_0 + \theta_1 x_1 \]

\[ h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_1^2 \]

\[ h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_1^2 + \theta_3 x_1^3 \]

\[ h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_1^2 + \ldots + \theta_7 x_1^7 \]
DECISION TREE - REGRESSION

### Predictors

<table>
<thead>
<tr>
<th>Outlook</th>
<th>Temp</th>
<th>Humidity</th>
<th>Windy</th>
<th>Hours Played</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rainy</td>
<td>Hot</td>
<td>High</td>
<td>False</td>
<td>25</td>
</tr>
<tr>
<td>Rainy</td>
<td>Hot</td>
<td>High</td>
<td>True</td>
<td>30</td>
</tr>
<tr>
<td>Overcast</td>
<td>Hot</td>
<td>High</td>
<td>False</td>
<td>46</td>
</tr>
<tr>
<td>Sunny</td>
<td>Mild</td>
<td>High</td>
<td>False</td>
<td>46</td>
</tr>
<tr>
<td>Sunny</td>
<td>Cool</td>
<td>Normal</td>
<td>False</td>
<td>62</td>
</tr>
<tr>
<td>Sunny</td>
<td>Cool</td>
<td>Normal</td>
<td>True</td>
<td>23</td>
</tr>
<tr>
<td>Overcast</td>
<td>Cool</td>
<td>Normal</td>
<td>True</td>
<td>43</td>
</tr>
<tr>
<td>Rainy</td>
<td>Mild</td>
<td>High</td>
<td>False</td>
<td>56</td>
</tr>
<tr>
<td>Rainy</td>
<td>Cool</td>
<td>Normal</td>
<td>False</td>
<td>88</td>
</tr>
<tr>
<td>Sunny</td>
<td>Mild</td>
<td>Normal</td>
<td>False</td>
<td>46</td>
</tr>
<tr>
<td>Rainy</td>
<td>Mild</td>
<td>Normal</td>
<td>True</td>
<td>46</td>
</tr>
<tr>
<td>Overcast</td>
<td>Mild</td>
<td>High</td>
<td>True</td>
<td>62</td>
</tr>
<tr>
<td>Overcast</td>
<td>Hot</td>
<td>Normal</td>
<td>False</td>
<td>44</td>
</tr>
<tr>
<td>Sunny</td>
<td>Mild</td>
<td>High</td>
<td>True</td>
<td>30</td>
</tr>
</tbody>
</table>

### Target

```
<table>
<thead>
<tr>
<th>Hours Played</th>
</tr>
</thead>
<tbody>
<tr>
<td>46.3</td>
</tr>
</tbody>
</table>
```

### Decision Tree

- **Outlook**
  - Sunny
  - Windy
    - FALSE: Hours Played = 47.7
    - TRUE: Hours Played = 26.5
  - Overcast
    - Temp.
      - Cool: Hours Played = 38
      - Hot: Hours Played = 27.5
      - Mild: Hours Played = 41.5
  - Rainy
AGENDA

1. More Supervised Learning
2. Bias/Variance
3. Cross-Validation
4. Quality Metrics
5. Embeddings
Bias and Variance

- There are technical definitions but also used informally

- **Bias** measures one kind of error
  - Difference between the answer and expected answer
  - Your pre-data model is “too strong”
  - Often, your model is too simple to capture the target domain, so you get the answer wrong a lot
  - Can be remedied by building a more flexible or higher-parameter model
  - A high bias model reflects strong assumptions about the domain
  - If you don’t have much training data, a high bias model might be your only option
Bias and Variance

- **Variance** is another kind of error
  - Measures spread of your answers around mean
  - Your model is “underfitting” or “overfitting”
    - (Put another way, you are not correctly sensitive to the training data)
  - Can be remedied by building a less flexible or lower-parameter model
  - Most variance bugs are due to high variance (that is, overfitting, which usually means you are too sensitive to the data)
Bias and Variance

Error

Training set M

++ - + - ++ - -+ - - + ++ -- + - -
Bias and Variance

Error

Training Set M

Training Set (m)

Test-Set (ts)

Training Set

Validation Set

++ -+ -++-- -- ++

- + ++-- + -- -

+ + ++- + - -

- - - - - - -
Bias and Variance

- Training Error

Training set M

Training Set (m)

Training Set

Validation Set
Bias and Variance

Error

Training Error

Training set M

Training Set (m)

Training Set

Validation Set
Bias and Variance

Training set M

Error

Training Error

Training Set (m)

Training Set

Validation Set
Bias and Variance

Error

Training set M

Training Error

Training Set (m)

Training Set

Validation Set

+++ --

+ +++- + - -
Bias and Variance

Training Error

Error

Training set M

Training Set (m)

Training Set

Validation Set
Bias and Variance

Training Error

Training set M

Training Set (m)

Training Set

Validation Set
Bias and Variance

Error

Training set M

Training Error

Training Set (m)

Training Set

Validation Set
Bias and Variance

Error

Training set M

- Training Error
- Test Error

Training Set (m)

Test error
a) decreases with M
b) increases with M
c) stays constant

Training Set

Validation Set
Bias and Variance

Error

Training set M

Training Error
Test Error

Training Set (m)

Training Set
Validation Set
High Bias

![Graph showing training set M, error, training error, test error.](Image)
If you have high-bias, does more data help?

a) No
b) Yes
If you have high-variance, does more data help?

a) No
b) Yes
Ideas for improving quality

1. Get more training examples
2. Try smaller sets of features
3. Try getting additional features
4. Try adding polynomial features (kernels)
5. Try increase regularization
6. Try decrease regularization
What would you do?

1. Get more training examples
2. Try smaller sets of features
3. Try getting additional features
4. Try adding polynomial features (kernels)
5. Try increase regularization
6. Try decrease regularization

Helps with
A. High Variance
B. High Bias
C. Both
D. None
What would you do?

1. Get more training examples
2. **Try smaller sets of features**
3. Try getting additional features
4. Try adding polynomial features (kernels)
5. Try increase regularization
6. Try decrease regularization

Helps with
A. High Variance
B. High Bias
C. Both
D. None
What would you do?

1. Get more training examples
2. Try smaller sets of features
3. **Try getting additional features**
4. Try adding polynomial features (kernels)
5. Try increase regularization
6. Try decrease regularization

Helps with

A. High Variance
B. High Bias
C. Both
D. None
What would you do?

1. Get more training examples
2. Try smaller sets of features
3. Try getting additional features
4. **Try adding polynomial features (kernels)**
5. Try increase regularization
6. Try decrease regularization

Helps with

A. High Variance
B. High Bias
C. Both
D. None
What would you do?

1. Get more training examples
2. Try smaller sets of features
3. Try getting additional features
4. Try adding polynomial features (kernels)
5. **Try increase regularization**
6. Try decrease regularization

Helps with
A. High Variance
B. High Bias
C. Both
D. None
What would you do?

1. Get more training examples
2. Try smaller sets of features
3. Try getting additional features
4. Try adding polynomial features (kernels)
5. Try increase regularization
6. Try decrease regularization

Helps with
A. High Variance
B. High Bias
C. Both
D. None
AGENDA

1. More Supervised Learning
2. Bias/Variance
3. Cross-Validation
4. Quality Metrics
5. Embeddings
Testing, Training, Validation

- **Training (~80%)**: the core data that allows a learning system to find good parameters. A typical training procedure may view this data repeatedly.

- **Validation (~10%)**: data that lets you estimate the success of training. Based on validation results, you might adjust hyperparameters or terminate training. Not all procedures use validation data.

- **Test (~10%)**: data that gives you a “final” and clean measure of your model’s accuracy.
Cross-validation k-fold: split the data into k groups, train on every group except for one, which you test on. Repeat for all groups.

**One iteration of a 5-fold Cross-Validation:**

1st Fold:
- Test set: Red
- Train set: Green

2nd Fold:
- Test set: Red
- Train set: Green

3rd Fold:
- Test set: Red
- Train set: Green

4th Fold:
- Test set: Red
- Train set: Green

5th Fold:
- Test set: Red
- Train set: Green

Bitsearch blog
AGENDA

1. More Supervised Learning
2. Bias/Variance
3. Cross-Validation
4. Quality Metrics
5. Embeddings
There are LOTS of error metrics

Classification:
• Accuracy
• F-score
• F1-micro
• F1-macro
• ROC AUC (micro, macro)

Ranking:
• Kendall’s Tau
• Mean Reciprocal Rank

Regression
• Mean-Squared Error
• Root-Mean Squared Error
• Mean absolute Error
• R²
• Cohen Kappa
PRECISSION, RECALL, ACCURACY

<table>
<thead>
<tr>
<th>Predicted Label</th>
<th>True Label</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>True</td>
<td>False</td>
</tr>
<tr>
<td>True</td>
<td>tp</td>
<td>fp</td>
</tr>
<tr>
<td>False</td>
<td>fn</td>
<td>tn</td>
</tr>
</tbody>
</table>

- **Precision**: correctly identified positive cases
  
  Precision $P = \frac{tp}{tp + fp}$

- **Recall**: correctly identified positive cases from all the actual positive cases.
  
  Recall $R = \frac{tp}{tp + fn}$

- **F-Score**: is the harmonic mean of precision and recall
  
  $F = \frac{2}{\frac{1}{P} + \frac{1}{R}} = \frac{tp}{tp + \frac{1}{2}(fp + fn)}$
Precision and recall

• Generally we trade precision vs. recall
  – How to get a system with high recall?
• Recall is a non-decreasing function of the # of docs retrieved
  – Precision *usually* decreases with more docs retrieved
• Drawbacks
  – Binary relevance (for search results)
  – Need human judgments
  – Must average over large corpus
  – Alternatively, skewed by corpus/author selection
Exercise

• Consider a search engine that always returns all documents

• Do you expect high or low precision?

• Do you expect high or low recall?
Exercise

• Consider a search engine that always returns all documents

• Do you expect high or low precision?
  – Low. If all docs are returned, then many non-relevant docs are included, which will decrease the percentage of returned docs that are relevant.

• Do you expect high or low recall?
  – High. If all docs are returned, then all relevant docs must be returned.

• Do you, personally, want a high-precision or high-recall search engine?
• Who might want the opposite?
Precision-recall curves

• A search engine will create a total ordering on all documents
• The top $k$ are returned to the user
• We can calculate precision and recall for several values of $k$
• This creates a precision-recall curve
P/R CURVES

Thanks https://www.datacamp.com/tutorial/precision-recall-curve-tutorial
Take Ranking Into Account

• Precision at fixed recall
  – Precision of top k results, for k=1,10,50,…
  – Critical for Web Search
• Use Kendall’s Tau for comparing sort orders
Kendall's Tau

• Use a real ordering of documents, not just binary "relevant/not relevant"
• The correct document ordering is:
  – 1, 2, 3, 4
• Search Engine A outputs:
  – 1, 2, 4, 3
• Search Engine B outputs:
  – 4, 3, 1, 2
• Intuitively, A is better. How do we capture this numerically?
Measuring Rank Correlation

• Kendall's Tau has some nice properties:
  – If agreement between 2 ranks is perfect, then KT = 1
  – If disagreement is perfect, then KT = -1
  – If rankings are uncorrelated, then KT = 0 on average

• Intuition: Compute fraction of pairwise orderings that are consistent
Kendall's Tau

The non-normalized version is called Kendall's Tau Distance

Also called bubble-sort distance

\[ \tau = \frac{n_c - n_d}{\frac{1}{2} n(n - 1)} \]

- The non-normalized version is called Kendall's Tau Distance
- Also called bubble-sort distance
Try it out

• Correct ordering:
  – 1, 2, 3, 4

• Search Engine A:
  – 1, 2, 4, 3

\[
\tau = \frac{5 - 1}{\frac{1}{2} \cdot 4(4 - 1)} = \frac{4}{6} = 0.666
\]

• Search Engine B:
  – 4, 3, 2, 1

\[
\tau = \frac{0 - 6}{\frac{1}{2} \cdot 4(4 - 1)} = \frac{-6}{6} = -1
\]
ROC AUC
(usually used for models with a threshold)

What would be the ideal ROC curve?
How would a random guess look like?
ROC AUC

(usually used for models with a threshold)

AUC: Area under the ROC Curve
Very similar, but not quite the same

What’s same? What’s different?

Which one would you prefer to use?
Evaluation:
Accuracy isn’t always enough

• Is 90% accuracy good or bad?
Evaluation:
Accuracy isn’t always enough

• Is 90% accuracy good or bad?
  – It depends on the problem
• Need a baseline:
  – Base Rate
    • Accuracy of trivially predicting the most-frequent class
  – Random Rate
    • Accuracy of making a random class assignment
    • Might apply prior knowledge to assign random distribution
  – Naïve Rate
    • Accuracy of some simple default or pre-existing model
    • Ex: “All females survived”
Why Baselines?

\[
\frac{38}{40} = 95\%
\]

\[
\frac{38}{40} = 95\%
\]
AGENDA

1. More Supervised Learning
2. Bias/Variance
3. Cross-Validation
4. Quality Metrics
5. Embeddings
Feature Engineering
How would you predict the unemployment rate before the official numbers come out?

Feature Engineering

• Dropping features
  – Remove duplicates
  – Highly correlated values (Zip code, Lon/Lat)

• Feature creation
  – Feature crosses: Cost per square feet
  – Creating special features (“I lost my job”)
  – Row statistics
    • Number of 0, nulls, negative value, mean, max, min,…
  – Projection to circle
    • Turn a single feature (like day_of_week) into two coordinates on a circle
    • Ensures that distance between Monday and Sunday etc is the same
  – Spatial
    • GPS encoding
    • Categorized locations (e.g., close to city, rural, nearby hospital, etc.)
  – Use embeddings from other models (more on that later)
  – Discretization (date \(\rightarrow\) weekend/weekday)
  – …
Transformations

• Rounding
  – Lossy
  – Precision can just be noise -> might improve training
  – Log transform before rounding often useful
• Binning
  – Removes information
  – Can work gracefully with variables outside of ranges seen in the train set
• Scaling
  – Standard (Z) Scaling
  – MinMax Scaling
  – Root Scaling
  – Log Scaling
• Outlier removal
• Imputation (mean, median, …)
• Interaction encoding: Specifically encodes the interaction between two numerical variables
  – Subtraction, Addition, Multiplication....
  – Polynomial encoding
    • Linear algorithms can not solve XOR problem
    • A polynomial kernel can solve XOR
Encodings

• **One-hot**
  • NaN, null, etc → create explicit encoding

• **Hash-encoding** (careful might introduce collisions)

• **Count encoding**: replace categorical value with their count
  – Useful for both linear and non-linear algorithms
  – Sensitive to outliers
  – Might create collisions

• **Rank encoding**: Rank categorical variables by count in train set
  – Useful for both linear and non-linear algorithms
  – Not sensitive to outliers
  – Won’t give same encoding to different variables

• **Target encoding**: Encode categorical variables by their ratio of target (binary classification) in train set
  – Be careful to avoid overfit
  – Add smoothing to avoid setting variable encoding to 0
  – Add random noise?
  – Can work extremely well when done right

• **Consolidation/expansion encoding**: map different categorical variables to the same
  – Spelling errors, slightly different job descriptions, abbreviations
Dear Home Owner,

Your credit doesn’t matter to us! If you own real estate and want IMMEDIATE cash to spend ANY way you like, or simply wish to LOWER your monthly payments by one third or more, here are the deals we have today:

$488,000.00 at 3.67% fixed rate  
$372,000.00 at 3.90% variable-rate  
$492,000.00 at 3.21% interest-only  
$248,000.00 at 3.36% fixed rate  
$198,000.00 at 3.55% variable rate

Hurry, when these deals are gone, they’re gone! Simple fill out the 1 minute form.

Don’t worry about approval, credit is not a matter!

CLICK HERE AND FILL THE 60 SECS FORM!

Bag of Words

- Urgent: 1
- money: 1
- Herbel: 2
- Pills: 2
- Are: 1
- ...

N-Grams

- herbel pills: 1
- pills for: 1
- for Hair: 2
- Hair growth: 1
- surgeries: 2
- ...

Spam

Not Spam
One-Hot Encoding

Bag of Words

\[
\begin{pmatrix}
Urgent \\
Money \\
Herbel \\
Pills \\
Are \\
\ldots
\end{pmatrix}
\]

<table>
<thead>
<tr>
<th>ID</th>
<th>Urgent</th>
<th>Money</th>
<th>Herbel</th>
<th>Pills</th>
<th>Are</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mail1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>...</td>
</tr>
<tr>
<td>Mail2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Word embeddings

• **Idea:** learn a high-dimensional representation of each word
  Cat: \{0.002, 0.244, 0.546, \ldots, 0.345\}

• Need to have a function \(W(\text{word})\) that returns a vector encoding that word.

• **Applications:** ???
Word embeddings: properties

Relationships between words correspond to difference between vectors.

\[ W(“woman”) - W(“man”) \approx W(“aunt”) - W(“uncle”) \]

\[ W(“woman”) - W(“man”) \approx W(“queen”) - W(“king”) \]

Word embeddings: questions

• How big should the embedding space be?
  – Trade-offs like any other machine learning problem – greater capacity versus efficiency and overfitting.

• How do we find $W$?
  – Often as part of a prediction or classification task involving neighboring words.
Learning word embeddings

• First attempt:
  – Input data is sets of 5 words from a meaningful sentence. E.g., “one of the best places”. Modify half of them by replacing middle word with a random word. “one of function best places”
  – $W$ is a map (depending on parameters, $Q$) from words to 50 dim’l vectors.
  – Feed 5 embeddings into a module $R$ to determine ‘valid’ or ‘invalid’
  – Optimize over $Q$ to predict better
