

CPSC 340: Machine Learning and Data Mining

Feature Selection

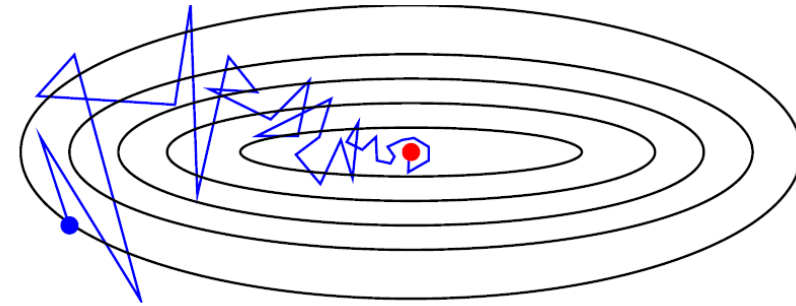
Admin

- hw4 posted
 - Due March 19 (you should probably do it earlier)
 - We will cover everything needed this week
- Midterm regrade requests
 - Must be submitted by **this Friday**
 - You can ask me before/after class today/Wednesday/Friday
- Final exam: April 25 at 8:30am

Last Time: Stochastic Gradient

- **Stochastic gradient** minimizes average of smooth functions:

$$f(w) = \frac{1}{n} \sum_{i=1}^n f_i(w)$$



- Function $f_i(w)$ is error for example 'i'.
- Iterations perform **gradient descent on one random example 'i'**:

$$w^{t+1} = w^t - \alpha^t \nabla f_i(w^t)$$

- Very cheap iterations even when 'n' is large.
- Doesn't always decrease 'f'.
- But solves **problem if α^t goes to 0** at an appropriate rate.

Last Time: Stochastic Gradient

- **Practical tricks** when using stochastic gradient:
 - Constant step-sizes, binary search for step, stop using validation error.
- Stochastic gradient **converges very slowly**:
 - But if your dataset is too big, there may not be much you can do.
 - Improved by “mini-batches” or “variance-reduced” methods (SAG, SVRG).

Motivation: Discovering Food Allergies

- Recall the food allergy example:

Egg	Milk	Fish	Wheat	Shellfish	Peanuts	...		Sick?
0	0.7	0	0.3	0	0		→	1
0.3	0.7	0	0.6	0	0.01		→	1
0	0	0	0.8	0	0		→	0
0.3	0.7	1.2	0	0.10	0.01		→	1

- Instead of predicting “sick”, we want to do **feature selection**:
 - Which foods are “relevant” for predicting “sick”.

Feature Selection

- General **feature selection** problem:

A hand-drawn diagram illustrating the feature selection problem. It shows a matrix X with a vertical column highlighted by a blue oval. A blue arrow points from this column to the text "feature 'j'". To the right of X is a vector y , both represented by hand-drawn brackets.

- Find the features (columns) of 'X' that are important for predicting 'y'.
 - “What are the relevant factors?”
 - “What is the right basis?”
 - “What types of new data should I collect?”
 - “How can I speed up computation?”
- One of most important problems in ML/statistics:
 - But it's **very very messy**...

Is “Relevance” Clearly Defined?

- Consider a supervised classification task:

gender	mom	dad	SNP
F	1	0	1
M	0	1	0
F	0	0	0
F	1	1	1

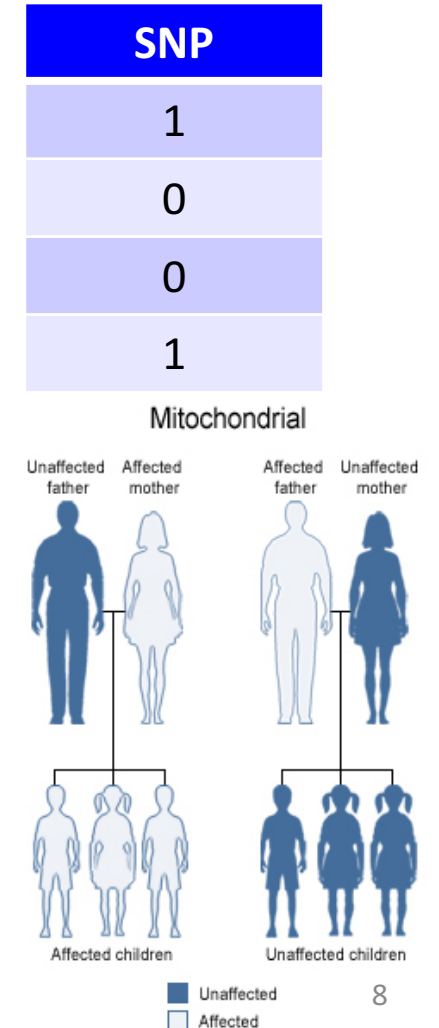
- Predict whether someone has a particular genetic variation (SNP).
 - Location of mutation is in “mitochondrial” DNA.
 - “You almost always have the same value as your mom”.

Is “Relevance” Clearly Defined?

- Consider a supervised classification task:

gender	mom	dad
F	1	0
M	0	1
F	0	0
F	1	1

- True model:
 - (SNP = mom) with very high probability.
 - (SNP != mom) with some very low probability.
- What are the “relevant” features for this problem?
 - Mom is relevant and {gender, dad} are not relevant.



Is “Relevance” Clearly Defined?

- What if “mom” feature is repeated?

gender	mom	dad	mom2
F	1	0	1
M	0	1	0
F	0	0	0
F	1	1	1

SNP
1
0
0
1

- Are “mom” and “mom2” relevant?
 - Should we pick them both?
 - Should we pick one because it lets predict the other?
- General problem (“dependence”, “collinearity” for linear models):
 - If features can be predicted from features, don’t know one(s) to pick.

Neither of these is “correct”, but not picking either is incorrect.

Detour: a bit more on Collinearity

- Imagine we are fitting a linear model with 2 identical features.
- The elements of the weight vector must add up to a **total weight**, but we can shuffle weight between the two elements **without changing the loss**.
 - In fact, we can even make one negative and one positive, so long as the total is correct. In terms of interpretation (and probably generalization), this is very worrying!
- With L2-regularization, we get a unique solution with two equal weights.
 - This is because squaring “amplifies” big values, so we want to make the weights small.
 - (I had claimed uniqueness earlier but not elaborated, this was the elaboration.)
- But with our feature selection mindset we’re still unsure which ones to select.
- You can also have pairs of features that are almost collinear, with similar implications.
- There are linear algebraic interpretations of all this: if X has two identical columns, then $X^T X$ will have two identical rows: thus it will be **singular**.

Is “Relevance” Clearly Defined?

- What if we add “grandma”?

gender	mom	dad	grandma
F	1	0	1
M	0	1	0
F	0	0	0
F	1	1	1

SNP
1
0
0
1

- Is “grandma” relevant?
 - You can predict SNP very accurately from “grandma” alone.
 - But “grandma” is irrelevant if I know “mom”.
- General problem (conditional independence):
 - “Relevant” features may be irrelevant given other features.

Is “Relevance” Clearly Defined?

- What if we don’t know “mom”?

gender	grandma	dad	SNP
F	1	0	1
M	0	1	0
F	0	0	0
F	1	1	1

- Now is “grandma” is relevant?
 - Without “mom” variable, using “grandma” is the best you can do.
- General problem:
 - Features can be **relevant due to missing information**.

Is “Relevance” Clearly Defined?

- What if we don’t know “mom” or “grandma”?

gender	dad
F	0
M	1
F	0
F	1

SNP
1
0
0
1

- Now there are no relevant variables, right?
 - But “dad” and “mom” must have some common maternal ancestor.
 - “Mitochondrial Eve” estimated to be ~200,000 years ago.
- General problem (effect size):
 - “Relevant” features may have small effects.

Is “Relevance” Clearly Defined?

- What if we don’t know “mom” or “grandma”?

gender	dad
F	0
M	1
F	0
F	1

SNP
1
0
0
1

- Now there are no relevant variables, right?
 - What if “mom” likes “dad” because he has the same SNP as her?
- General problem (**confounding**):
 - Hidden effects can **make “irrelevant” variables “relevant”**.

Is “Relevance” Clearly Defined?

- What if we add “sibling”?

gender	dad	sibling	SNP
F	0	1	1
M	1	0	0
F	0	0	0
F	1	1	1

- Sibling is “relevant” for predicting SNP, but it’s not the cause.
- General problem (non-causality or reverse **causality**):
 - A “relevant” feature **may not be causal**, or may be an effect of label.

Is “Relevance” Clearly Defined?

- What if we add “baby”?

gender	dad	baby
F	0	1
M	1	1
F	0	0
F	1	1

SNP
1
0
0
1

- “Baby” is relevant when (gender == F).
 - “Baby” is relevant (though causality is reversed).
 - Is “gender” relevant?
 - If we want to find relevant factors, “gender” is not relevant.
 - If we want to predict SNP, “gender” is relevant.
- General problems (context-specific relevance or conditional relevance):
 - Adding a feature can make an “irrelevant” feature “relevant”.

Is “Relevance” Clearly Defined?

- Warnings about feature selection:
 - A feature is **only “relevant” in the context of available features.**
 - Adding/removing features can make other features relevant/irrelevant.
 - Confounding factors can **make “irrelevant” variables the most “relevant”.**
 - If features can be predicted from features, **you can’t know which to pick.**
 - A “relevant” feature may have a **tiny effect.**
 - “Relevance” for prediction does **not imply a causal relationship.**

Is this hopeless?

- In the end, we often want to do feature selection we so have to try!
- We **won't be able to resolve causality or confounding**.
 - So “relevance” could mean “affected by confounding” or “affected by label”.
 - This can sometimes be addressed by the **way you collect data** (see bonus slides).
- Different methods will behave differently with respect to:
 - **Tiny effects**.
 - **Context-specific relevance** (is “gender” relevant if given “baby”?).
 - **Variable dependence** (“mom” and “mom2” have same information).
 - **Conditional independence** (“grandma” is irrelevant given “mom”).

“Association” Approach to Feature Selection

- A simple/common way to do feature selection:

for $j = 1:d$

compute “similarity” between $X(:,j)$ and y

Say ‘ j ’ is “relevant” if “similarity” is above a threshold.

- “Similarity” could be correlation, mutual information, etc.
- Ignores tiny effects.
- Reasonable for variable dependence: it will take “mom” and “mom2”.
- Not reasonable for conditional independence:
 - It will take “grandma”, “great-grandma”, “great-great grandma”, etc.
- Not reasonable for context-specific relevance:
 - If two features aren’t relevant on their own, then both set as “irrelevant”.
(This method will say “gender” is “irrelevant” given “baby”.)

} Systematically
includes
irrelevant
variables.

“Regression Weight” Approach to Feature Selection

- A simple/common approach to feature selection:

Fit 'w' using least squares

Take all Features 'j' where $|w_j|$ is greater than some threshold.

- Deals **very badly with variable dependence**:

- If can take two irrelevant collinear variables:

- Set one w_j hugely positive and the other hugely negative.

} → Systematically adds irrelevant variables

- Means it can **allow tiny effects**.

- It **could take any subset** of {"mom", "mom2", "mom3"}, including none.

- It should **address conditional independence**:

- Should take “mom” but not “grandma” if you get enough data.

- It addresses **context-specific relevance**, **if effect is linear**.

- Because we aren’t looking at each feature independently.

- This one says “gender” is “relevant”.


↓
This is bad

“Regression Weight” Approach to Feature Selection

- A simple/common approach to feature selection:

Fit ' w ' using least squares with L_2 -regularization.

Take all features ' j ' where $|w_j|$ is greater than some threshold.

- Same good properties with respect to independence/context.
- Deals less badly with collinearity:
 - If you have two irrelevant collinear variables, doesn't take them.
 - No longer allows tiny affects.
 - But it could say “mom” and “mom2” are both irrelevant.  Bad!
 - Sum of their weights could be above threshold, with neither weight above threshold.
 - This effect is even worse if you have {mom, mom2, mom3, ..., mom10}

Common Approaches to Feature Selection

- 3 main “advanced” approaches to feature selection:
 1. Search and score (today)
 2. L1-Regularization (next class)
 3. Hypothesis testing (see bonus slides)
- None is ideal, but good to know advantages/disadvantages.

Feature Selection Approach 1: Search and Score

- Two components behind **search and score** methods:
 - Define a **score function $f(s)$** that says how “good” a set of variables ‘s’ are:
 - Now **search** for the variables ‘s’ with the best value of $f(s)$.
- Under usual score functions, very **hard to find the best ‘s’**.
- Usual greedy approach is **forward selection**:
 - Start with ‘s’ empty, add variable that increase score the most, repeat.
- Many variations like “backward” and “stagewise” selection.

Feature Selection Approach 1: Search and Score

- Two components behind **search and score** methods:
 - Define a **score function $f(s)$** that says how “good” a set of variables ‘s’ are:
 - Now **search** for the variables ‘s’ with the best value of $f(s)$.
- **Can’t use training error as the score**: you’ll just add all features.
- Usual score functions:
 - **Validation/cross-validation**:
 - Good if your goal is prediction.
 - Tends to give false positives because you search over many subsets.
 - **L0-“norm”**:
 - Balance training error and number of non-zero variables.
 - You will implement this on Assignment 4

L0-Norm

- In linear models, setting $w_j = 0$ is the same as removing feature 'j':

$$y_i = w_1 x_{i1} + w_2 x_{i2} + w_3 x_{i3} + \dots + w_d x_{id}$$

\downarrow set $w_2 = 0$

$$y_i = w_1 x_{i1} + \underbrace{0}_{\text{ignore } x_{i2}} + w_3 x_{i3} + \dots + w_d x_{id}$$

- The L0-“norm” is the number of non-zero values.

$$\text{If } w = \begin{bmatrix} 1 \\ 0 \\ 2 \\ 0 \\ 3 \end{bmatrix} \text{ then } \|w\|_0 = 3 \qquad \text{If } w = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \text{ then } \|w\|_0 = 0.$$

- Not actually a true norm.
- A vector with many elements set to 0 is called a **sparse vector**.

L0-Norm

- L0-norm regularization for feature selection:

$$f(w) = \frac{1}{2} \|Xw - y\|^2 + \lambda \|w\|_0$$

- Balances between training error and number of features.
- Different values of λ give common feature selection scores:
 - Akaike information criterion (AIC).
 - Bayesian information criterion (BIC).
- To we use $f(w)$ to score features 's':
 - Solve least squares problem using only features 's'.
 - Compute $f(w)$ above with all other w_j set to zero.

Search and Score Issues

- Advantages:
 - Deals with conditional independence (if linear).
 - Sort of **deals with collinearity**:
 - Cross-validation picks at least one of “mom” and “mom2”.
 - L0-norm will pick only one of “mom” or “mom2”.
- Disadvantages:
 - Difficult to define ‘correct’ score:
 - Cross-validation often selects too many.
 - L0-norm selects too few/many depending on λ .
 - Under most scores, it’s hard to find optimal features (use greedy approximation).
- Neither good nor bad:
 - Does not take small effects.
 - Says “gender” is relevant if we know “baby”.
 - **This approach is better for prediction than the previous approaches.**

Summary

- **Feature selection** is task of choosing the relevant features.
 - Hard to define “relevant” and many problems that can have.
 - Obvious approaches have obvious problems.
- **Hypothesis testing**: find sets that make y_i and x_{ij} independent.
- **Search and score**: find features that optimize some score.
- Next time:
 - L1-regularization.