Machine Learning: Foundations

Features, missing values, and some basics

SUVRIT SRA

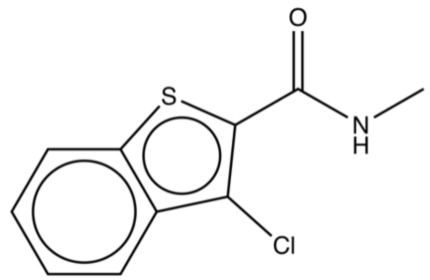
Massachusetts Institute of Technology

ml.mit.edu

Slides adapted from: Tommi Jaakkola, Stefanie Jegelka

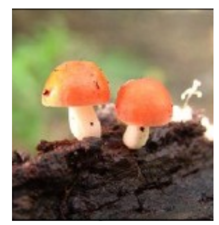
Our focus: modeling

- E.g., supervised learning



→ potency, toxicity, etc

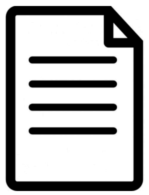
→ sentiment, etc.



→ category

- Learn to formulate problems as learning tasks, matching problems and methods, understand how to encode information effectively
- Learn a toolbox of machine learning methods so as to be able to see what's possible, and how
- Understand when things work, how to evaluate and revise

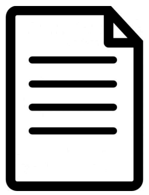
Where we are



x

input object
for prediction

Where we are



x

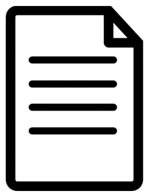
input object
for prediction

$$\begin{bmatrix} 1 \\ 0 \\ \vdots \\ 1 \end{bmatrix}$$

$\phi(x)$

feature vector

Where we are



x

input object
for prediction

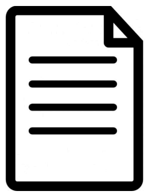
$$\longrightarrow \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 1 \end{bmatrix} \longrightarrow f = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_d \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 1 \end{bmatrix} + \theta_0$$

$\phi(x)$

feature vector

linear predictor $f = \theta \cdot \phi(x) + \theta_0$

Where we are



x

input object
for prediction

$\longrightarrow$

$$\begin{bmatrix} 1 \\ 0 \\ \vdots \\ 1 \end{bmatrix}$$

$\phi(x)$

feature vector

$\longrightarrow$

$$f = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_d \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 1 \end{bmatrix} + \theta_0$$

linear predictor $f = \theta \cdot \phi(x) + \theta_0$

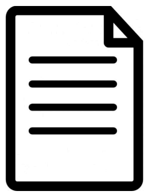
linear regression $f = \theta \cdot \phi(x) + \theta_0 \quad \in \mathbb{R}$

linear classification $h = \text{sign}(\theta \cdot \phi(x) + \theta_0) \quad \in \{-1,1\}$

logistic regression $p = g(\theta \cdot \phi(x) + \theta_0) \quad \in [0,1]$ $\qquad g(z) = \dfrac{1}{1 + \exp(-z)}$

etc.

Our goal today



x

input object
for prediction

$$\begin{bmatrix} 1 \\ 0 \\ \vdots \\ 1 \end{bmatrix}$$

$\phi(x)$

feature vector

$$f = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_d \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 1 \end{bmatrix} + \theta_0$$

linear predictor $f = \theta \cdot \phi(x) + \theta_0$

(1) what should the feature vector be?

(2) what if some examples have missing features?

Feature Engineering for Spam Filtering

```
Delivered-To: suvrit@gmail.com
Received: by 10.157.56.186 with SMTP id p55csp703174otc;
        Sat, 30 Sep 2017 07:34:01 -0700 (PDT)
X-Google-Smtp-Source: AOwi7QCK9y8KOdBTvMLfrbOcOxH51i9LN6lkw+lhc/IkwZLmb8UrH7HW7JKfqLt7xj1TwDsf3dX1
X-Received: by 10.80.193.26 with SMTP id l26mr13268612edf.97.1506782040979;
        Sat, 30 Sep 2017 07:34:00 -0700 (PDT)
ARC-Seal: i=1; a=rsa-sha256; t=1506782040; cv=none;
        d=google.com; s=arc-20160816;
        b=BxVulFx86G2rfNXnEsv2kNx5pFgiU0jkZKmJ6TwkNBg/dr5MrDXJsGEihegHN7u0IU
         6J1J5kn6q8sMk84OaGfeeDAOatE37P2Strqc4c4BivUY2YKD3wxp31wstrE5hLYDw1od
         lmzTjTzfWZKv1AB1fRuKDCgn+bnpIvWyL03aEo5dMiuztpvA0HId90leNsxJo442Cn3J
         0s3kVvw4nJZVoDUO5zsLwD8XrHFP8M6k739NXdTLfVyci78BjQseVXrlf8a/ya/fHfvL
         xYl07foHl7nXUvVprfx/D9EooB+8v0e/IjiBHFf1lnPNAi6fcx1p53S70dNQ0VJss/GM
         rFdA==
ARC-Message-Signature: i=1; a=rsa-sha256; c=relaxed/relaxed; d=google.com; s=arc-20160816;
        h=feedback-id:date:to:reply-to:from:subject:message-id:mime-version
         :dkim-signature:dkim-signature:arc-authentication-results;
        bh=R4godP4US7Q4tcEB7kBYBKAfq6im2AktH8nd+AKvSNY=;
        b=oJa78ArtTLtDeN2+Byqt5hil7chs4JrD2djnrXA5t6u7Jab5SfRXYWY9y9YNOl6Cqu
         3a58btCCKgs1e84Rf2GNlAv7+zdV2rpGmI79cQ6dGKPQPItshGRzGMCl39tprnAUOGvd
         jLYV8/o/RCejaHDelJ/UKPRaLjrAtTYjaDOL5wJSTkjkjBgiR8dP2lea7xmLnPZ+cB2H
         FkZIGrEOpG4in01bfx1yvapCv9vrwtR4+DFAOGbZxE4NRdq7CUu9ewTTpjkFmaFZMXB8
         SwrDrkh8P86DUjcCsOlIYUp7EVQdw7Lc0p7AMP1ZaFkWoIuTcwShzDUe6Q1MAZudto0B
         xGsQ==
ARC-Authentication-Results: i=1; mx.google.com;
       dkim=pass header.i=@axolbio.com header.s=yeey7jbyihrzecpea7e3xbglnfzh65yg header.b=N75Hyb7v;
       dkim=pass header.i=@amazonses.com header.s=shh3fegwg5fppqsuzphvschd53n6ihuv header.b=WkaTV2p1;
       spf=pass (google.com: domain of
0102015ed3354feb-392b28ee-3ca2-485c-8b4d-87aff57c8b67-000000@amazonses.axolbio.com designates 54.240.4.3
as permitted sender)
smtp.mailfrom=0102015ed3354feb-392b28ee-3ca2-485c-8b4d-87aff57c8b67-000000@amazonses.axolbio.com;
       dmarc=pass (p=NONE sp=NONE dis=NONE) header.from=axolbio.com
Return-Path: <0102015ed3354feb-392b28ee-3ca2-485c-8b4d-87aff57c8b67-000000@amazonses.axolbio.com>
Received: from a4-3.smtp-out.eu-west-1.amazonses.com (a4-3.smtp-out.eu-west-1.amazonses.com.
[54.240.4.3])
        by mx.google.com with ESMTPS id x13si6722343edx.227.2017.09.30.07.34.00
        for <suvrit@gmail.com>
        (version=TLS1 cipher=ECDHE-RSA-AES128-SHA bits=128/128);
        Sat, 30 Sep 2017 07:34:00 -0700 (PDT)
Received-SPF: pass (google.com: domain of
0102015ed3354feb-392b28ee-3ca2-485c-8b4d-87aff57c8b67-000000@amazonses.axolbio.com designates 54.240.4.3
as permitted sender) client-ip=54.240.4.3;
Authentication-Results: mx.google.com;
       dkim=pass header.i=@axolbio.com header.s=yeey7jbyihrzecpea7e3xbglnfzh65yg header.b=N75Hyb7v;
       dkim=pass header.i=@amazonses.com header.s=shh3fegwg5fppqsuzphvschd53n6ihuv header.b=WkaTV2p1;
       spf=pass (google.com: domain of
0102015ed3354feb-392b28ee-3ca2-485c-8b4d-87aff57c8b67-000000@amazonses.axolbio.com designates 54.240.4.3
as permitted sender)
smtp.mailfrom=0102015ed3354feb-392b28ee-3ca2-485c-8b4d-87aff57c8b67-000000@amazonses.axolbio.com;
       dmarc=pass (p=NONE sp=NONE dis=NONE) header.from=axolbio.com
DKIM-Signature: v=1; a=rsa-sha256; q=dns/txt; c=relaxed/simple; s=yeey7jbyihrzecpea7e3xbglnfzh65yg;
d=axolbio.com; t=1506782040; h=Content-Type:MIME-Version:Message-Id:Subject:From:Reply-To:To:Date;
bh=R4godP4US7Q4tcEB7kBYBKAfq6im2AktH8nd+AKvSNY=;
b=N75Hyb7vZ3s13rX+JqDoN5U5Yoo9cwUTbNRLI9KUrG07oGpokoLxYWPhL8WcTw21
IkTsnw2S1M3jPmX5gVrcqehxtE7CkjnkaX0z4P+q31nSYOoeOvmxN35SjB6tNBPGMvh
lczINQOPSKtSkd2y812VB4hrBg6ToPL2dP3xYtcg=
DKIM-Signature: v=1; a=rsa-sha256; q=dns/txt; c=relaxed/simple; s=shh3fegwg5fppqsuzphvschd53n6ihuv;
d=amazonses.com; t=1506782040; h=Content-Type:MIME-Version:Message-Id:Subject:From:Reply-
To:To:Date:Feedback-ID; bh=R4godP4US7Q4tcEB7kBYBKAfq6im2AktH8nd+AKvSNY=; b=WkaTV2p1v987maaZaWS/
6JbrUq13AC7JNw12Jq0jP/pDyF20WALENqNXB1Er+4Wj AXU3ARVMePb+hHSAyqKghaWIo6iCH1u7XWV9kbHz1k69LK+/
cg2Dci8zIyhWZ8PsywW +x4qLfirwM2uQTqYnArzLe1sq/lcl6Wtfn3AulVg=
Content-Type: multipart/mixed; boundary="===============5772367175723715400=="
MIME-Version: 1.0
Message-ID: <0102015ed3354feb-392b28ee-3ca2-485c-8b4d-87aff57c8b67-000000@eu-west-1.amazonses.com>
Subject: Come meet us at Drug Discovery 2017 in Liverpool!
From: Axol Bioscience <query@axolbio.com>
Reply-To: Axol Bioscience <query@axolbio.com>
To: suvrit@gmail.com
Date: Sat, 30 Sep 2017 14:34:00 +0000
X-SES-Outgoing: 2017.09.30-54.240.4.3
Feedback-ID: 1.eu-west-1.dsZhP2MFwA9Gu8y1ynbiR1E1xEwjZOlznrjLL2AvoJg=:AmazonSES
```

Constructing Features: naive OCR system

	1	2	3	4	5	6	7	8	9	0
Loops	0	0	0	1	0	1	0	2	1	1
3 Joints	0	0	0	0	0	1	0	0	1	0
4 Joints	0	0	0	1	0	0	0	1	0	0
Angles	0	1	1	1	1	0	1	0	0	0
Ink	1	2	2	2	2	2	1	3	2	2

Feature engineering

- Handwritten Japanese Character Recognition
  - Break down the images into strokes and recognize it
  - Lookup based on stroke order
- Medical Diagnosis
  - Physician's comments
  - Blood status / ECG / height / weight / temperature ...
  - Medical knowledge
- Preprocessing
  - Zero mean, unit variance to fix scaling (e.g. weight vs. income)
  - Probability integral transform (inverse CDF) as alternative
- Click through rate (CTR)
  - One-hot, or one-of-K encoding
  - But first need to decide "what raw features" to collect!

Difficult, expensive, create feature, hope it has discriminatory power
Can be very valuable in practice (**yes, even these days!**)

Feature encoding

Part of ‘preprocessing’ in ML

Reading:

[https://scikit-learn.org/stable/modules/preprocessing.html]

(1) Strategies of feature encoding

- The information is typically not available in the form that should be used directly in (say) a linear regression model
- E.g., predicting house (selling) price
  - type of house x_1: 1,2,3,4
  - number of bathrooms x_2: 1,2,3,4 or more
  - size in sqft: x_3
  - etc.

(1) Strategies of feature encoding

- The information is typically not available in the form that should be used directly in (say) a linear regression model
- E.g., predicting house (selling) price
  - type of house x_1: 1,2,3,4
  - number of bathrooms x_2: 1,2,3,4 or more
  - size in sqft: x_3
  - etc.

- We would like to select an effective feature vector $\phi(x)$ for linear regression
- For simplicity, let's consider each $x_i, i = 1,2,3$ separately (as if they were the only feature)

(1) Strategies of feature encoding

- The information is typically not available in the form that should be used directly in (say) a linear regression model
- E.g., predicting house (selling) price
  - type of house x_1: 1,2,3,4
  - number of bathrooms x_2: 1,2,3,4 or more
  - size in sqft: x_3
  - etc.

(1) Strategies of feature encoding

- The information is typically not available in the form that should be used directly in (say) a linear regression model
- E.g., predicting house (selling) price
  - type of house x_1: 1,2,3,4
  - number of bathrooms x_2: 1,2,3,4 or more
  - size in sqft: x_3
  - etc.

e.g., $\phi(x_1) = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \quad \text{if } x_1 = 2$

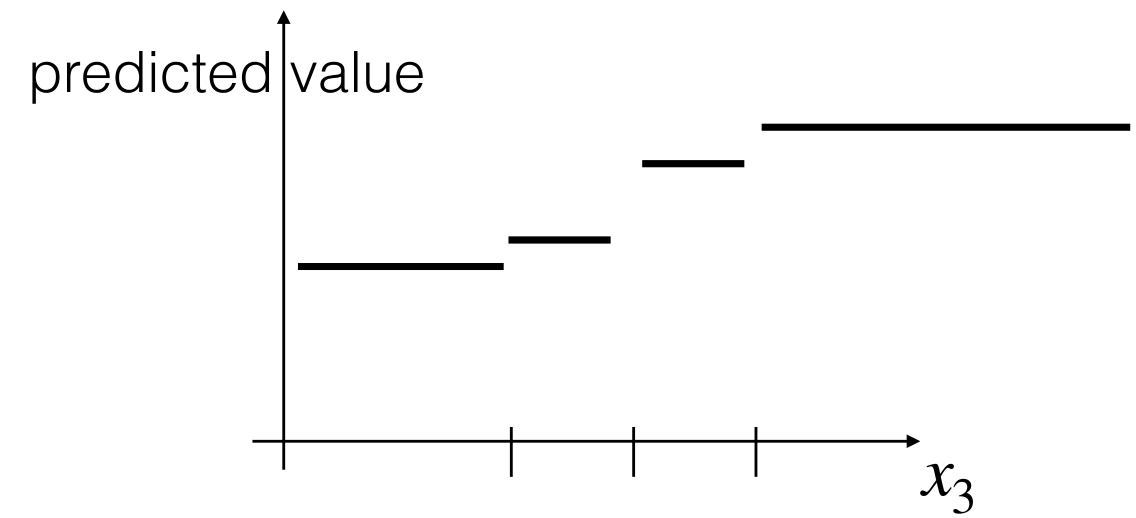
one-hot vector for nominal values
(of low cardinality)

(1) Strategies of feature encoding

- The information is typically not available in the form that should be used directly in (say) a linear regression model
- E.g., predicting house (selling) price
  - type of house x_1: 1,2,3,4
  - number of bathrooms x_2: 1,2,3,4 or more
  - **size in sqft:** x_3
  - etc.

(1) Strategies of feature encoding

- The information is typically not available in the form that should be used directly in (say) a linear regression model
- E.g., predicting house (selling) price
  - type of house x_1: 1,2,3,4
  - number of bathrooms x_2: 1,2,3,4 or more
  - **size in sqft:** x_3
  - etc.



step 1: e.g., divide x_3 into intervals such as

[0,1000),
[1000,1500),
[1500,2000),
[2000 +)

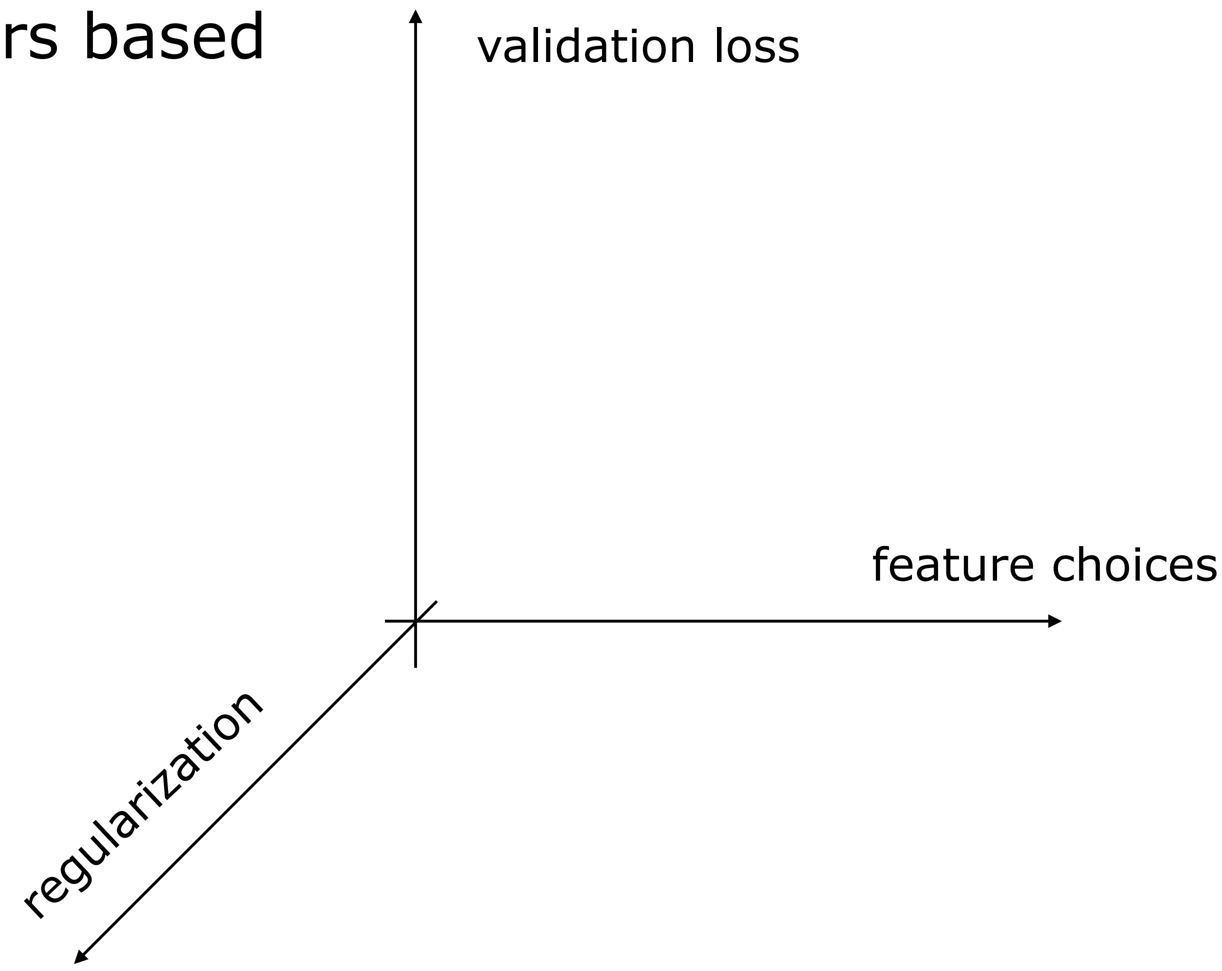
e.g., $\phi(x_3) = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$, if $x_3 \in [1000, 1500)$

step 2: one-hot vector for interval membership

Question: How should we go about making these "choices"?

(1) Hyper-parameter optimization

- We can optimize also over possible feature transformations as hyper-parameters based on validation performance



Costly choice, can be partially automated (AutoML) though with limited success.

(2) Partially missing features

- Much of the data we have is heterogeneous and incomplete in some ways
- There are many scenarios/ways for data to be missing; these “mechanisms” matter
- E.g., risk of recurrence prediction

age	ethnicity	grade	size
43	1	3	11
55	2		
	3		5
70		2	9
	2		15
65	3		
30	1	3	

(2) Partially missing features

- Much of the data we have is heterogeneous and incomplete in some ways
- There are many scenarios/ways for data to be missing; these "mechanisms" matter
- E.g., risk of recurrence prediction
- Missingness mechanisms
  A. "missing completely at random"
  B. "missing at random"
  C. "not missing at random" (e.g., censoring)

age	ethnicity	grade	size
43	1	3	11
55	2		
	3		5
70		2	9
	2		15
65	3		
30	1	3	

(2) Partially missing features

- Much of the data we have is heterogeneous and incomplete in some ways
- There are many scenarios/ways for data to be missing; these "mechanisms" matter
- E.g., risk of recurrence prediction
- Missingness mechanisms
  A. "missing completely at random"
  B. "missing at random"
  ~~C. "not missing at random" (e.g., censoring)~~

- Algorithms for filling-in or imputing missing data
  - mean/median imputation
  - dedicated symbol + learned coefficient
  - nearest neighbor imputation
  - multiple imputation by chained equations
  - etc.

age	ethnicity	grade	size
43	1	3	11
55	2		
	3		5
70		2	9
	2		15
65	3		
30	1	3	

Multiple Imputation by Chained Eqs

- Step 1: fill in all the missing values (e.g., randomly)
- Step 2: estimate a model to predict each feature value using only real targets
- Step 3: rewrite missing values using model predictions
- Goto step 2

age	ethnicity	grade	size
43	1	3	11
55	2		
	3		5
70		2	9
	2		15
65	3		
30	1	3	

Multiple Imputation by Chained Eqs

- Step 1: fill in all the missing values (e.g., randomly)
- Step 2: estimate a model to predict each feature value using only real targets
- Step 3: rewrite missing values using model predictions
- Goto step 2

age	ethnicity	grade	size
43	1	3	11
55	2	1	9
30	3	3	5
70	1	2	9
55	2	1	15
65	3	2	5
30	1	3	15

Multiple Imputation by Chained Eqs

- Step 1: fill in all the missing values (e.g., randomly)
- Step 2: estimate a model to predict each feature value using only real targets
- Step 3: rewrite missing values using model predictions
- Goto step 2

age	ethnicity	grade	size
43	1	3	11
55	2	1	9
30	3	3	5
70	1	2	9
55	2	1	15
65	3	2	5
30	1	3	15
x_1	x_2	$\mathbf{y}$	x_4

Multiple Imputation by Chained Eqs

- Step 1: fill in all the missing values (e.g., randomly)
- Step 2: estimate a model to predict each feature value using only real targets
- Step 3: rewrite missing values using model predictions
- Goto step 2

age	ethnicity	grade	size
43	1	3	11
55	2	1	9
30	3	2	5
70	1	2	9
55	2	3	15
65	3	2	5
30	1	3	15
x_1	x_2	$\hat{y}$	x_4

Multiple Imputation by Chained Eqs

- Step 1: fill in all the missing values (e.g., randomly)
- Step 2: estimate a model to predict each feature value using only real targets
- Step 3: rewrite missing values using model predictions
- Goto step 2

age	ethnicity	grade	size
43	1	3	11
55	2	1	9
30	3	2	5
70	1	2	9
55	2	3	15
65	3	2	5
30	1	3	15
x_1	x_2	x_3	$\mathbf{y}$

What we are not solving (yet)

- Different techniques needed when the data are predominantly missing as in recommender problems

- Computational issues:
  - a typical matrix is very large,
- Statistical issues:
  - the matrix is very sparse, e.g., 1% known ratings
  - ratings may be diverse and under-sampled (?)
- Formulation issues:
  - many interpretations for missing entries

m movies

n users

5	5						5		
		3	5	1	3	4	4		4
	4	2			2				
		5							5
4	5							4	
4							4		
5		4	5	1		4			
	4								
5				4					
5						4			
		5				5		3	

Thoughts: What about other approaches to missing data?