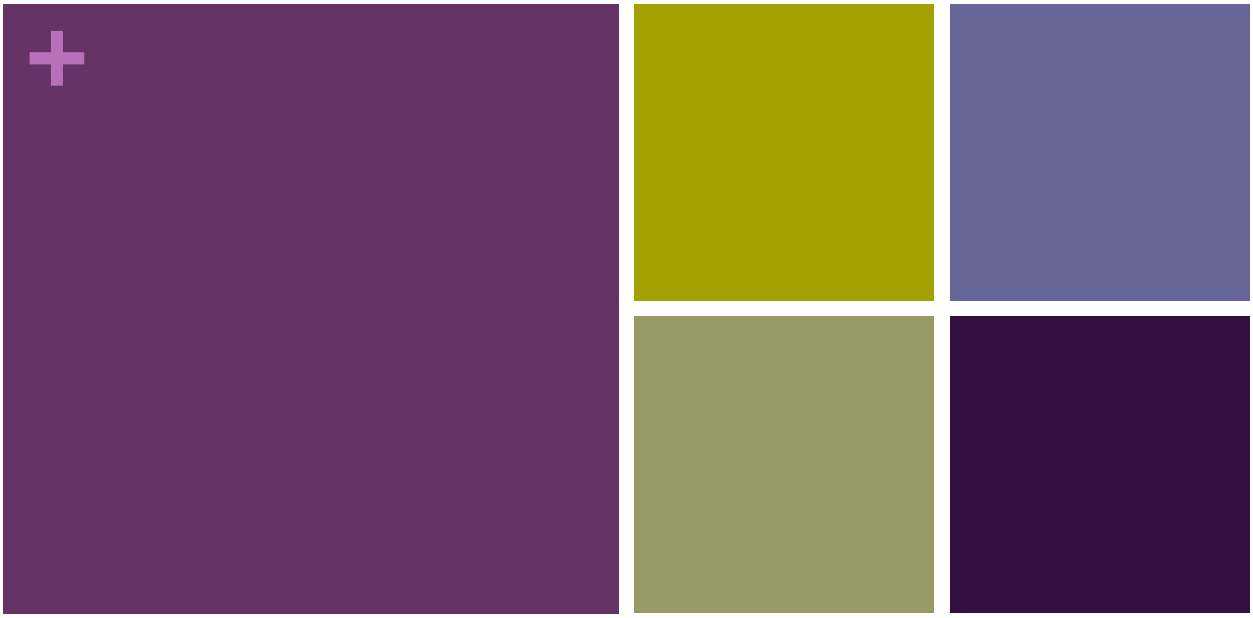


Quantitative methods in CompSci

Some observations from being a reviewer and an author

You and your readers

- Why do you want to write something?
  - You've been told to
  - Show you are clever
  - … etc
  - But also: **Because you have something important to tell**
- Why would a reader want you to write something?
  - **Because you have something important to tell**
  - Anything else?
- Why would a reader dislike your writing?
  - Show they are clever
  - "Envy" / You "block" an idea
  - … etc

YOU NEED TO "PROVE" YOU ARE WORTH THEIR TIME!

Show them you have something to say, that is in fact important …

How to prove you have something imnportant to tell?

- Depends on the type of publication
  - Review articles
  - Discussion/Comment-based articles
  - Application-papers
  - Etc etc.
  - Algorithm/method development:
    a. Prove analytically that your method is optimal or at least comparative to state-of-the-art methods
    b. Show empirically that your method is optimal or at least comparative to state-of-the-art methods

Note! Method (a) is preferable to Method (b), but in many cases only Method (b) is feasible

The term "Quantitative methods" in this presentation

- I will talk about quantitative methods used for validating a classification algorithm:
  - Classification accuracy generally accepted as a proxy for "algorithm goodness"
- So, I do not need to think that much about:
  - Hypothesis establishment
  - Data collection
  - Data validation
  - Etc.

Classifier learning

- How to make the optimal classifier given infinite data and unlimited compute is already well-known, but that method is unfeasible with limited data to learn from.
- New classifiers are made by **making assumptions** about the data (that we know are wrong!!) to make learning feasible.
- Remember that ***"essentially, all models are wrong, but some are useful"***. The trick is to find an **"almost true"** set of assumptions that make your models **useful**.
- **Success can only be shown empirically.**

Do-s and **Don't**-s in classifier evaluation

- **Evaluate all assumptions** you make (alternatively all assumptions you lift if starting from a standard algorithm).
- Use **standard datasets** (not your own fabricated/fake data) if you want to make general statements. Otherwise, adjust your conclusions accordingly.
- Use a **broad collection** of datasets, not only those that fit your purpose.
- Measure **statistical uncertainty**, and do not overdo your conclusions.
- Use a a broad class of baselines and **be fair**.
- Don't force it: A **negative result** is still a result and can – given proper treatment – still become a nice paper.

Remember: If the reviewer cannot reproduce your empirical results, the whole validation argument (and sometimes also the whole paper) will be seen as useless!

Empirical investigation of classifiers and threats to validity

- Are the conclusions drawn **valid**?
  - Statistical **variations** neglected?
  - Are the **conclusions *actually*** tested?
  - Are new results compared to reasonable **baselines**?
- Are the conclusions **general**?
  - Too **few** datasets used for testing?
  - Too **obscure** datasets used for testing?
- As a reviewer I see **many** papers that make unjustified claims (validity threatened as above). **All of them have been rejected**.

QUIZ: Er dette lov/meningsfullt?

- Jeg har laget en ny klassifikasjonsalgoritme, som heter «**Helges Vidunderlige Klassifikator**» (a.k.a. HVK)
- Når jeg lærer/velger ut/bygger modellen bruker jeg et datasett for å optimere nøyaktigheten (accuracy) HVK oppnår.
- Når jeg er ferdig rapporterer jeg den accuracy'en jeg har beregnet.

SVAR

- **NEI!!!!**
- Når jeg har brukt data til å lære er disse "oppbrukt", og jeg kan ikke bruke de samme til å finne ut hvor bra jeg har blitt.
- Tenk på det sånn:
  - Lærealgoritmen er ekstra god på de dataene den har lært med
  - Du vil være for optimistisk dersom du rapporterer dette tallet.
- **HOLD AV ET EGET TEST-SETT** (eller bruk kryssvalidering)

QUIZ: Er DETTE lov/meningsfullt?

- Jeg tester HVK på 25 forskjellige standard datasett mot en state-of-the art classifier.
- På datasettet `wine` er HVK best (hypotesetest på 5% nivå).
- Jeg konkluderer med at HVK er den beste på `wine`, og bruker dette datasettet i rapporten min. Jeg døper om HVK til HVWK – «Helges Vidunderlige `Wine` Klassifikator».

SVAR

- **NEI!!**
- Har 25 data-sett, og for hver av dem er det 5% sjanse for feil konklusjon (det er dette 5%-nivå i testen betyr).
  → Må forvente at 25*5%=1.25 av de 25 går galt, og det er dermed (kanskje) bare tilfeldig at HVK vinner på `wine`…
- Jeg skal ikke gjemme bort de datasettene der HVK ikke vinner, og eneste muligheten for å bare fokusere på et subset av datasettene er hvis jeg kan begrunne at disse på noe vis er «spesielle». **Alt annet er juks og fanteri**!
- **HVIS** jeg kan si noe om hva som er spesielt med `wine` og som HVK utnytter kan jeg kanskje komme med en god hypotese om at HVK er bra på **slike datasett**, men da må hypotesen **testes videre**.

QUIZ: Hva med dette?

- Jeg genererer HVK ved å kjøre noen genetiske algoritmer. Jeg har gjort 100 repetisjoner, der jeg for hver gang har brukt 250 individer over 50 generasjoner.
- Jeg har latt hver av de 100 GA'ene velge en "vinner". Fitness-evaluering gjøres på **TRENINGS-DATA** (dvs. jeg har holdt **TEST-DATA** borte fra GA'en)
- Jeg beregner accuracy på **TEST-DATA** for hver av de 100 vinnerne, og rapporterer den beste av disse verdiene som HVK's accuracy.

SVAR

- **NEI!!**
- Hvis jeg har en metode der jeg (FØR jeg beregner nøyaktighet på testsettet) kan velge hvilken av de 100 GA-vinnerne jeg vil sette som **DEN UTVALGTE** kan jeg rapportere accuracy'en **den** har på testsettet.
- Standard løsning vil være å bruke **VALIDERINGS-DATA** for å finne «the one».
- Hvis jeg ikke har noen måte å velge mellom de 100 **må** jeg rapportere hele range'en av verdier, f.ex. ved å angi en av disse:
  - min/median/max
  - mean/stdev
  - Empirisk konfidensintervall

QUIZ: NÅ da???

- **DEL 1:** HVK har masse hyper-parametre som settes eksternt (learning-rate, batch-size, loss-regularisering, …).
- Jeg tester ut forskjellige hyper-param combo's ved
  - … å lære modell med gitt hyper-parametre fra **trenings-data**
  - … beregner accuracy med **validerings-data**
- Til slutt velger jeg modellen med best oppførsel (på valideringsdata), sjekker accuracy for denne på **test-data**, og rapporterer dette tallet som HVKs resultat.
- **DEL 2:** Det viser seg at random-seed'et er viktig for hva som læres. Jeg inkluderer derfor random seed i hyper-parameter-søket, og velger det beste seed'et på samme vis som over.

SVAR

- Del 1: **JA!!** Endelig! Det er slik det skal gjøres…
- Del 2: Jeg synes **NEI**, selv om dette ikke er formelt galt.
  - Hva sier det om robusthet / brukbarhet / verdi av metoden din dersom random seed er en viktig «parameter»?
  - Du bør heller repetere kjøringen med flere seeds, og rapportere effekten i form av, f.ex., mean + st.dev. av accuracy'ene.

Going all in on statistics: Example – Calculating accuracy

- A classifier correctly classifies 80 of 100 examples.
- Estimated classifier accuracy: 80 out of 100
  → Acc. = 80/100 = 0.8.
- How to think about this:
  - The classifier has some **ability**; a probability *p* to classify new examples correctly.
  - When we estimate accuracy from *n* examples, it is like *"flipping a coin"* with P(classify example correctly) = *p* each time.
  - Accuracy estimate: ρ= *#successes / #trials*.
  - **Note!** ρ is **an estimator** of *p*, it is not (necessarily) **equal** to *p*.
  - We'd like to say something about how "**close**" *p* to ρ; that way we can say something "general-ish" about *p* based on the estimate ρ.

Statistics – Robustness of accuracy

- I calculate the accuracy of a classifier $\boldsymbol{L}$ on a test-set $x_1, x_2, \ldots, x_n$ simply by checking for each example x_i if $\boldsymbol{L}$ guesses right.
- ρ is simply the observed fraction of successes. **Uncertainty?**
- The higher the n, the **more certain** can the estimator be. Quantify by giving uncertainty bounds, **approximated** by

$$\rho \pm z_{1-\alpha/2,n-1}\sqrt{\frac{\rho\cdot(1-\rho)}{n}}$$

  - $z_{1-\alpha/2,n-1}$ is s.t. $P(Z_{n-1} \le z_{1-\alpha/2,n-1}) = 1-\alpha/2$ when Z_{n-1} is t-distributed with $n-1$ degrees of freedom (typically approximately 2.0 for $\alpha=5\%$ and relatively large n).
- **Example:** 80 of 100 successful gives
  - $\rho = 0.8$
  - Interval: (0.7, 0.9) – approximately…

Statistics – Comparing PAIRS

- I have two sets of observations $x_1, x_2, \ldots, x_n$ and $y_1, y_2, \ldots, y_n$, where x_i and y_i are pairs exposed to different "treatments" (e.g., correctness of two algorithms $\boldsymbol{L_x}$ and $\boldsymbol{L_y}$ on samples $D_1, D_2, \ldots, D_n$, and I want to conclude that $\boldsymbol{L_x}$ is "better" than $\boldsymbol{L_y}$).
- **Assumptions**:
  - AVERAGE(X-Y) ≈ Gaussian($\mu, \sigma^2/n$) – That this is approximately true follows from "The Standard Limit Theorem" when n is large enough.
  - We do not know the parameters (μ, σ), but want to conclude $\mu > 0$.
- Generate a statistical hypothesis test:
  - $H_0: \mu \le 0$
  - $H_1: \mu > 0$ Note: H_1 is always what we hope to conclude (here $\boldsymbol{L_x} > \boldsymbol{L_y}$)
- Under H_0, AVERAGE(X-Y) = *AVERAGE(X) – AVERAGE(Y) follows a t-*distribution with $\mu = 0$ and estimated standard deviation.
  ➔ Can do "Paired t-test". (If n is small, there are other alternatives…)

Statistics – Comparing GROUPS

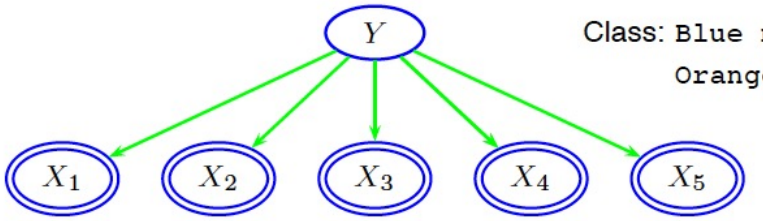
- I have two sets of observations $x_1, x_2, \ldots, x_n$ and $y_1, y_2, \ldots, y_m$. I want to conclude that the **x**-group is "better" than the other
- **Assumptions**:
  - $X \sim N(\mu_X, \sigma_X^2)$, $Y \sim N(\mu_Y, \sigma_Y^2)$.
  - We do not know the parameters (μ_X, μ_Y, σ_X, σ_Y).
  - We are interested in $\delta = \mu_X - \mu_Y$ and want to conclude that $\delta > 0$.
- Generate a statistical hypothesis test:
  - H_0: $\delta \leq 0$
  - H_1: $\delta > 0$
- Under H_0, δ that is estimated by *AVERAGE(X) – AVERAGE(Y)* *approx. follows a t*-distribution; expectation 0 and estimated *standard dev + df.'s.*
  ➔ Can do "independent two-sample t-test".

An example of how to do empirical evaluation (I hope…):

- Helge Langseth and Thomas D. Nielsen (2005):
  Latent Classification Models
  Machine Learning **59**(3), pp. 237-265, 2005.
- Starting from a well-known classifier (Naïve Bayes), we develop a new classification algorithm.

The Naïve Bayes classifier: an example

A Naïve Bayes classifier for the crabs domain:



Class: `Blue male, Blue female`
`Orange male, Orange female`

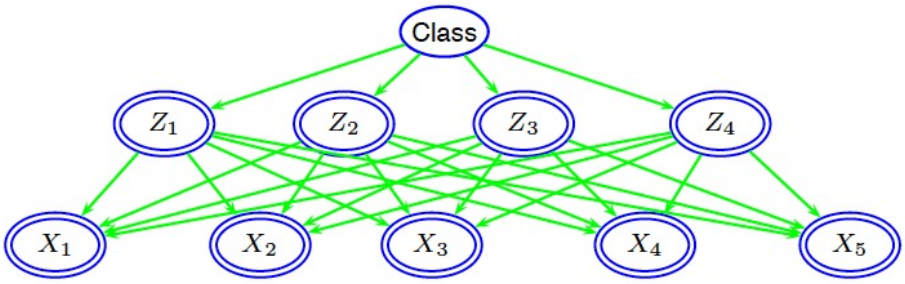
X_1 - `Width of frontal lib`
X_2 - `Length along the midline`
X_5 - `Body length`
X_4 - `Rear width`
X_5 - `Maximum width of the carapace`

Step 0: Define starting position and work-plan
- Define a base-line.
- Analyze the base-line to suggest useful extensions.

The two assumptions:

- The attributes are conditionally independent given the class.
- The continuous attributes are generated by a specific parametric family of distributions.

Latent Classification Models: The linear case



For the quantitative part we have:

- Conditionally on $Y = j$ the latent variables, $\boldsymbol{Z}$, follow a Gaussian distribution with $\mathbb{E}[\boldsymbol{Z} \mid Y = j] = \boldsymbol{\mu}_j$ and $\text{Cov}(\boldsymbol{Z} \mid Y = j) = \boldsymbol{\Gamma}_j$.
- Conditionally on $\boldsymbol{Z}$ $\mathbb{E}(\boldsymbol{X}|\boldsymbol{Z} = \boldsymbol{z}) = \boldsymbol{L}\boldsymbol{z}$

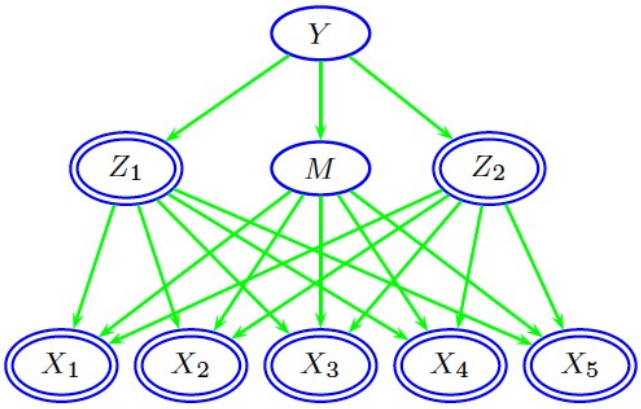
Note that:

- Both $\boldsymbol{\Gamma}_j$ and Θ mus
- As opposed to the g $\mathbb{E}(\boldsymbol{Z}|Y) = \boldsymbol{0}$ and C

Step 1: Generalization (1)
- Define a new class of classifiers that lift the first assumption (conditional independence).
- **Analytically prove** that all classifiers with the remaining assumption (distribution) can be represented. Also consider effects of **learning bias**.
- Give the class a name for **later empirical testing**

The complexity is not that bad...

Latent Classification Models: the non-linear case



For the quantitative part everything is identical to the linear LCM except that:

- The mixture variable
- Conditionally on $\{Z$
  - $\mathbb{E}(\boldsymbol{X}|\boldsymbol{z},m) = \boldsymbol{\mu}$
  - $\text{Cov}(\boldsymbol{X}|\boldsymbol{z},m) =$

Step 2: Generalization (2)

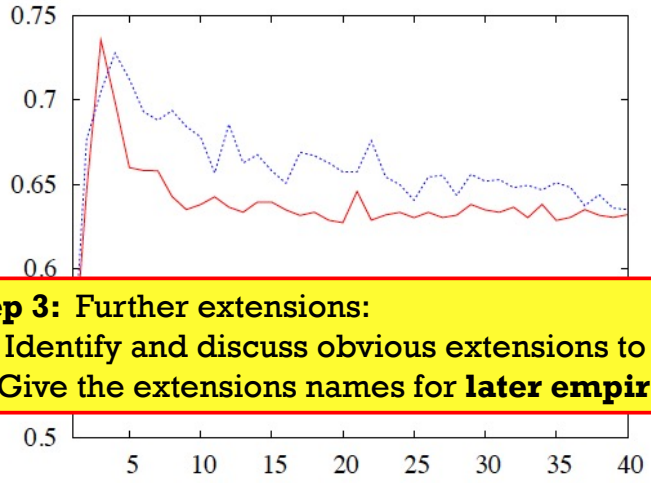
- Enhance this class by defining a new class of classifiers that lift the **second** assumption (distribution).
- **Analytically prove** that all classifiers can be represented. Also consider effects of **learning bias**.
- Give the class a name for **later empirical testing**

Learning the edge set

First of all, note that:

- We will only consider the edge-set between $\boldsymbol{Z}$ and $\boldsymbol{X}$.
- All these edges must be directed from $\boldsymbol{Z}$ to $\boldsymbol{X}$.

The reason for considering this set is that even though the number of parameters is only proportional to the number of arcs in the model we still risk the problem of overfitting:



Step 3: Further extensions:

- Identify and discuss obvious extensions to the model(s).
- Give the extensions names for **later empirical testing**

Experimental results: Data sets

We have con

D

ba

br

Experimental results: LCM setting

For linear LCMs:

Experimental results: Other classifiers

We have compared the LCM classifiers with 9 other classifiers in order to:

- Evaluate the accuracy of the LCM classifier.
- Ex

The class

NB:

NB/M:

FA/BIC

PCA/λ

PCA/n

CW/PCA/n: One PCA/n fitted per class; classification by Bayes rule.

CG/PCA/n: Unsuper. clust. #clusters decided by BIC; One PCA/n per cluster; classification by voting.

well

boun

ximu

0^{-3}

Step 4: Define the testing protocol:

- Datasets
- Which baselines are used (and why)
- Model parameters (new model + baselines). **BE FAIR!**
- Make sure the experiments are meaningful and relevant:
  Can they enable you to make interesting claims?
  What would it take of the result for those claims to be valid?...

Experimental results

Uncertainties

Hyp. tests

Database	NB	NB/M	FA/BIC	PCA/λ	PCA/n	CW/PCA/n	CG/PCA/n
balance-scale	$^{-}86.9 +/- 1.4$	$^{*}53.9 +/- 2.0$	$^{*}53.3 +/- 2.0$	$^{-}86.9 +/- 1.4$	$^{-}86.9 +/- 1.4$	$\mathbf{90.9 +/- 1.2}$	$^{-}76.3 +/- 1.7$
Average	74.9	75.6	81.3	80.9	83.8	83.2	81.7

Step 5: Analysis

"Everyone" can produce numbers, the trick is to understand what they tell.

- Give quantitative results – Remember uncertainty measurements
- **Give your interpretation of the results**.
- Give a **stepwise** evaluation – aka ablation study.
  In this case:
  - The NB
  - NB with Assumption 1 lifted, keeping Assumption 2
  - NB With Assumption 2 lifted, keeping Assumption 1
  - NB with both lifted
  - Final tweaks

  What was helpful, what was not. **Why?**