Advanced Machine Learning, Lecture 6

András Kornai

BME 2020 Oct 15

Homeworks, projects

- Some make-up work has been submitted
- We have only 15 project plans so far!
- Will discuss these individually
- Will take 2 minutes or less/person or team
- So far, exactly one team
- Only people with excessive courseloads (≥ 9) can form teams

Maximum entropy

- If you set up n random linear equations in n unknowns, there will be exactly one solution (with probability 1)
- If you have *more* equations than unknowns, you have no solutions (with probability 1)
- Gauss to the rescue! You will have a unique *best* solution (in the least squares sense)
- What to do when you have *fewer* equations than unknowns?
- There is an infinite number of solutions (with probability 1)
- E.T. Jaynes to the rescue!

Looking for a probability distribution

- We want to estimate $p_1, \ldots, p_n$ such that $\sum_i p_i = 1$
- We have some overall knowledge about events A_i e.g. that $f(A_i) = F_i$ and we know (e.g. from observation) that $\mathbb{E}(f) = c$
- Numerical example: $c = 1.75$ and

event	prob	f
A_1	p	1
A_2	q	2
A_3	r	3

- We have two equations, $p + q + r = 1$ and $p + 2q + 3r = 1.75$, what to do?
- $H = -p \log p - q \log q - r \log r$ is the entropy of the distribution: choose the one for which this is maximal!
- When there is exactly one less equation than unknown, this can be solved analytically

Looking for a pd (2)

- $q + 2r = 0.75$ so $q = 0.75 - 2r$ and $p = 1 - q - r = 0.25 + r$
- Maximize H, minimize -H:
  $(0.25 + r)\log(0.25 + r) + (0.75 - 2r)\log(0.75 - 2r) + r\log(r)$
- After differentiation, we have
  $\log(0.25 + r) - 2\log(0.75 - 2r) + \log(r) = 0$
- Gives rise to quadratic equation with root at $\frac{13-\sqrt{61}}{24} = 0.2162$
- Yes, but what do we do when there are far more equations than variables?
- What is the basic technique of constrained optimization?

Lagrange multipliers

- Our primary interest is not with modeling the distribution p as with joint modeling of distribution classes for best classification. We have a bunch of samples (n-dim vectors whose components are the direct measurements, plus one *class variable* c), and we can use any $\mathbb{R}^{n+1} \to \mathbb{R}$ function f_i as an *indicator* or *feature* to distinguish the classes
- When we have k classes, the ideal features $f_1, \ldots f_k$ would be the indicator functions that take 1 on class j and 0 elsewhere
- We are looking for a distribution with maximum entropy H among all distributions that satisfy constraints given to us in the form $p(f_i) = \tilde{p}(f_i)$ (p is the expected value, and $\tilde{p}$ is the measured value)
- The constraints are $\frac{1}{N}\sum_{d\in D} f_i(d, c(d)) = \frac{1}{N}\sum_{d\in D}\sum_c P(c|d) f_i(d, c)$
- Altogether, we want to maximize the Lagrangian $\Lambda(p, \lambda) = H(p) + \sum_i \lambda_i (p(f_i) - \tilde{p}(f_i))$

Looking for features

- There is a unique distribution p with maximum entropy, and it always has the form $P(c|d) = \frac{1}{Z(d)} \exp(\sum_i \lambda_i f_i(d, c))$
- Here $Z(d)$ is the *partition function* $\sum_c \exp(\sum_i \lambda_i f_i(d, c))$ (required to make sure probabilities sum to 1)
- The maxent feature weights were originally obtained by (Improved) Iterative Scaling, nowadays we use L-BFGS
- A key issue is feature selection, dropping features that are not helpful
- In *extrinsic filtering* we check e.g. correlations with features first
- In *outer loop* or *wrapper* methods we just recompute the model with some features dropped and see if it gets better
- In *embedded methods* we drop features that receive low λ_i weights

An early application

- English-French Machine Translation (Berger et al 1996)
- certain "N de N" phrases are inverted: bureau de poste – post office; compagnie d'assurance – insurance company; ...
- others stay put: pays d'origin – country of origin; somme d'argent – sum of money; ...
- predict which is which based on one of the words or both
- clear tendencies: système de X typically inverts, mois de X typically stays put
- But there are at minimum 50k nouns to be considered, potentially giving rise to 2.5g N de N constructions.

Berger et al (2)

- We can't collect enough data, and can't expect to memorize it!
- What we have is a small sample, maybe 10k pairs labeled for invert/stay
- Let's assume *all* words are relevant, in 1st place, 2nd, or that both words are relevant
- These all contribute to the model before feature selection
- But we keep less than 400
- Generalizes remarkably well to unseen data

A more recent application

- Digital language death (Kornai 2013)
- There are over 7k languages spoken today, 2.5k (35%) were considered endangered (in the traditional sense – 100 year time horizon). We investigated digital survival.
- Four classes: Thriving, Vital, Heritage, Still. Few dozen manually selected examples of each. E.g. French, Spanish, Chinese are T; Czech, Finnish are V; Latin, Classical Chinese are H; Rerau, Terik S.
- Measurements collected along 30+ dimensions: number of speakers, existence of spellchcker, size of wikipedia, ...
- 6 features kept. Result: over 95% of languages not just endangered, but already digitally dead. It's not "there will be an extinction", the extinction is done.