# Plug and Play Unplugged

**Regularized inversion** = **snaha získat původní signál** nebo data na základě pozorovaných měření/neúplných dat. Tzn je to pojmenování problematiky

* U declipingu chci získat původní neclipnutý signál (resp. najít nejlepší řešení, které by mu odpovídalo)
* Inverzní problémy jsou často **ill-posed** = **nemají jednoznačné řešení** nebo řešení vůbec neexistuje
* **Regularizer** má za cíl stabilizovat řešení tím, že přidávají **penalizace**
  + Preference pro řešení s určitými vlastnostmi
    - Signál je hladký, řídký ve spektru, aj. fyzikální/statistické vlastnosti
  + L1 regularizace (penalizace řídkosti)
  + L2 regularizace (penalizace velikosti koeficientů)
  + TV regularizace (total variation)
  + **Regularizer vs prior/regularity component**
    - Regularizer je konkrétní funkce
    - Prior je informace/předpoklad použitý pro formování přístupu k hledání řešení
    - **Regularizer může být implementací prior component**
  + **Regularity component**
    - přidává dodatečné předpoklady (prior) o struktuře původního signálu
    - Příklady regularizing funkcí:
      * **Tikhonova regularizace** 🡪 zajišťuje, že řešení bude **hladké,** tzn nebude mít velké oscilace

****

* + - * + **L** může být třeba operátor derivace
      * **Total variation (TV regularizace) 🡪** obraz by měl být hladký a zároveň umožňovat ostré hrany



* + - * **L1 norm regularizace** 🡪 penalizuje počet nenulových prvků signálu (preferuje řídká řešení).



* + - * + Používáme to, když očekáváme, že signál má většinu svých komponent nulových (třeba ve frekvenční doméně)
  + **Data fidelity compnent**
    - Zajišťuje aby byl signál věrný pozorovaným datům = chci, aby řešení co nejlépe odpovídalo tomu, co je v datech (tomu, co bylo naměřeno = clipovanému signálu)
    - Příklady data fidelity funkcí:
      * **Squared error** 🡪 minimalizuju čtvercovou chybu mezi pozorovaným signálem x a minimalizovaným signálem y



* + - * **Absolute error** 🡪 minimalizuju rozdíl signálů

****

**Consensus Equilibrium (CE)** se iterativně snaží **nalézt rovnováhu** mezi těmito dvěma složkami

* **Agent** je každá jednotlivá metoda/algoritmus, kterou použiju (řídkost, neuronky)
  + Jeho součástí je **regularizer**
* CE kombinuje více regularizérů 🡪 každý přispívá k rekonstrukci
* **Cost function** je funkce, která měří kvalitu řešení nějakého problému, tedy vyjadřuje chybu/náklady spojené s řešením
  + V mém případě se cost function používá pro optimalizaci inverzního problému = rekonstrukci signálu
  + Její hodnotu se snažím **minimalizovat**



* + V mém příkladě zahrnuje cost funkce J(x) část **data fidelity (f)** a část pro **regularizaci (h)**

* **MAP (Maximum A Posteriori) odhad**
  + Přístup, který se používá ve statistice k **nalezení nejpravděpodobnějšího odhadu neznámého parametru** z dostupných dat a předchozích informací (tzv. priorů)
  + Bayesova věta (neznámý parametr x)

Obsah obrázku Písmo, text, bílé, řada/pruh

Popis byl vytvořen automaticky

* + - P(x|y) = posteriorní pravděpodobnost
    - P(y|x) = likelihood
    - P(x) = prior
      * Tedy jaké mám předchozí znalosti o parametru x
    - P(y) = evidence
      * Zajišťujeme že součet všech pravděpodobností bude 1
  + **MAP odhad** 🡪 nalezení takového x, které maximalizuje **P(x|y)** = hledáme nejpravděpodobnější **hodnotu x** na základě **dat y** a předchozích znalostí (**priorů**)
  + Obsah obrázku text, Písmo, bílé, typografie

    Popis byl vytvořen automaticky🡪 Obsah obrázku text, Písmo, bílé, typografie

    Popis byl vytvořen automaticky

**Proximal maps**

* Je to matematický konstrukt používaný v optimalizaci (často ve spojení s funkcemi co **nejsou hladké** nebo maj **složitou strukturu**)
* *Pro hladké funkce můžu použít gradientní metody, tzn proximal maps jsou náhradou gradientních metod pro nehladké fce*
* **Hladká funkce** = **má všechny derivace**, tedy nemají zuby a skoky
  + Vypadají třeba takto:
  +  🡪 je hladká funkce
  +  🡪 není hladká funkce

Obsah obrázku řada/pruh, Vykreslený graf, diagram, svah

Popis byl vytvořen automaticky

* Proximal maps umožňuje **řešit** optimalizační **problémy**, které obsahují nespojitosti a složitosti, tedy například **regularizační funkce** používané při rekonstrukci signálů.
* Metoda CE je závislá proximálních mapách, protože **regularizační funkce jsou často nehladké** a já je potřebuju (jakožto součást cost function) **minimalizovat**
* Proximal map pro funkci h(y):

Obsah obrázku Písmo, řada/pruh, bílé, typografie

Popis byl vytvořen automaticky

* + Snažím se **najít hodnotu „y“**, která **minimalizuje** kombinaci funkce **h (regularizační funkce!!!)** a **square erroru (data fidelity component!!!)**, který penalizuje odchylku od **bodu** „**x“**
* **[https://math.stackexchange.com/questions/168159/what-is-the-motivation-of-proximal-mapping-proximal-operator](https://math.stackexchange.com/questions/168159/what-is-the-motivation-of-proximal-mapping-proximal-operator" \o "Výborný příklad na prox operátor (druhá odpověď))**
* Proximal maps používáme v CE. Máme N vektorových map. je jedním agentem řešícím problematiku (declipping, impainting). CE pro tyto mapy (agenty) je definováno jako jakékoliv řešení soustavy:

Obsah obrázku Písmo, text, řada/pruh, bílé

Popis byl vytvořen automaticky

* **u** je vektor vzniklý z vektorů a je vážený průměr definovaný jako:

Obsah obrázku Písmo, typografie, bílé, text

Popis byl vytvořen automaticky

* tzn „účka“ vyvažují muj vstupní signál a postupně se mění v jednotlivých iteracích algoritmu CE
* průměr těchto vyvažovacích signálů by měl být 0
* nyní definujeme jako proximální mapu 🡪 předpokládáme že regularizační/data fidelity funkce jsou uzavřené konvexní funkce:

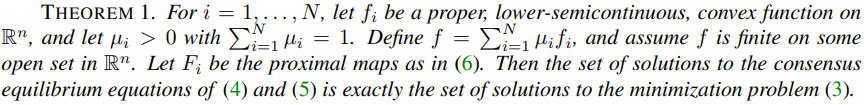
Obsah obrázku Písmo, text, řada/pruh, bílé

Popis byl vytvořen automaticky

* To znamená, že je **hladká**, tedy **má spojité derivace**
  + Platí, že , tzn je funkce, která na vstupu přijme vektor (vstupní signál) a vrátí nějaké číslo
* V článku se popisuje, že **pokud všechna jsou proximálními mapami**, tak řešení metody CE jsou řešeními pro consensus optimization problem definovaný v rovnici:

Obsah obrázku text, Písmo, snímek obrazovky, bílé

Popis byl vytvořen automaticky



* Tohle defakto tvrdí, že řešení soustavy proximal maps výš je řešením minimalizačního problému consensus optimization

**Plug and Play** je libovolná metoda, která kombinuje různé komponenty pro regularizaci bez nutnosti definovat regularizační funkci.

* Tzn 🡪 kombinuju prostě nějaký metody pro zpracování signálů a explicitně neuvádim proč to funguje?
* V klasických inverzních problémech optimalizuju funkci:



* f(x) je věrnost dat, tzn jak moc odpovida vyslednej signal tomu poškozenýmu, kterým to krmim
* h(x) je regularizátor, kterej ukládá jak musí signál vypadat (hladkost, řídkost ve spektru)
  + v Plug and Play ale h(x) **nemusí být explicitně definovaná funkce**. Místo definice se využívá **proximal map** nebo nějaká regularizační metoda, která nemusí mít známou analytickou formu, ale lze ji aplikovat prostřednictvím iterativního denoiseru
* **Consensus equilibrium je jedním z přístupů, který může být použit v Plug-and-Play frameworku pro kombinaci více agentů (declipperů, denoiserů).**🡪 tzn. takto to souvisí dohromady…
* V CE frameworku každý agent nebo komponenta odpovídá určitému regularizéru nebo řeší část problému
* Zadaný set agentů může vést k několika možným CE řešení. To se může stát i v ?optimalizačním frameworku? (to skoro vypadá, jak kdyby to byla oddělená část), kde nejsou striktně konvexní a mohou mít několik lokálních minim
  + *Může být použita objective function, která z nalezených lokálních minim vybere 🡨 tohle dovysvětlit*
* **Analogické řešení pro CE** může být nalezení řešení, které minimalizuje velikost nebo L1/L2 normy
  + To se dá chápat i tak, že se snažim minimalizovat rozdíly mezi různými výsledky jednotlivých agentů, abych dosáhl shody = equilibria
* Fixed point problém 🡪 hledám body/hodnoty, které se funkce/operátory zobrazí samy na sebe
  + Pokud mám , pak hledám takový bod , pro který platí:

**Řešení rovnic CE**

* Článek se dál věnuje alternativní reprezentaci pro CE framework 🡪 všechno se předefinuje pomocí matic, aby šlo efektivnějš (=míň polopaticky a víc krypticky) popsat, jak spolu agenti interagují a jak se stabilizuje dosažené řešení
* Předefinování systému rovnic viz výš:

Obsah obrázku Písmo, text, řada/pruh, diagram

Popis byl vytvořen automaticky



* slouží k popisu vztahů mezi agenty. Popisuje jak se jednotlivá řešení agentů navzájem liší. **Zavedení téhle matice má za cíl popsat to, jak dochází ke stabilizaci mezi agenty**
* „**v**“ jsou všechna řešení mých agentů pod sebou a ta když zprůměruju**, dostanu to řešení, které chci.** Narzdíl od , který popisovalo dílčí řešení jednotlivých agentů

Obsah obrázku text, Písmo, snímek obrazovky, řada/pruh

Popis byl vytvořen automaticky

* Tenhle theorem jen formalizuje podmínky, za kterých řešení pro CE existuje a je jednoznačné
* Vyplývá z něj to, že pro jednu konfiguraci vstupních oporátorů a parametrů **existuje pouze jedno konsenzuální řešení**, ke kterému systém konverguje

Obsah obrázku text, Písmo, snímek obrazovky, algebra

Popis byl vytvořen automaticky

* Pokud používam neexpanzivní operátory, bude metoda založená na CE frameworku konvergovat ke stabilnímu řešení

Obsah obrázku text, Písmo, snímek obrazovky, řada/pruh

Popis byl vytvořen automaticky

* **I** je **jednotková matice**, neboli matice identity
* \* 🡪 hvězdička označuje optimální řešení minimalizačního problému
* ^ 🡪 stříška označuje odhad nebo optimalizaci proměnné

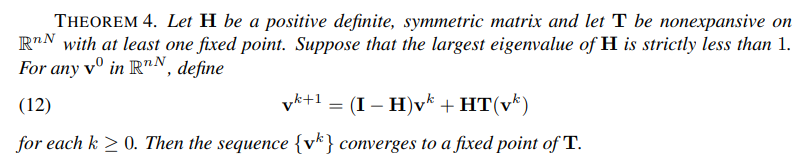
Dále definujeme T jako neexpanzivní a má pevný bod (tzn bod, který se zobrazí sám na sebe):

Pro řešení tohohle problému (=nalezení pevného bodu T) používáme Mann iteraci. K tomu definujeme:





* Strana 6 je prostě mess… wtf co se tam děje…
* Něco se děje a najednou předefinováváme Mann iteraci, kde místo najednou používáme matici (lineární mapu) **H**



* měnim na H proto, abych mohl váhovat každou složku vektoru jiným číslem. Jinak bych to vše váhoval stejnou skalární hodnotou?

**Beyond nonexpansive maps**

* Mam zaručeno, že iterativní algoritmy s **w** a **v** z rovnic (11) a (12) konvergují, když T je nonexpanzivní a rho/H splňují nějaké podmínky
* V týhle části se řeší **co dělat když operátory nejsou nutně neexpanzivní 🡪 vlastně tu zavádíme myšlenku, že je to možné…**
* Říká se tu, že i expanzivní operátory mohou obohatit řešení a jde pomocí nějakých praktik docílit toho, aby řešení konvergovalo i když některé mapy budou expanzivní

**Newton’s Method**

* Když CE formuluju jako řešení rovnic , tak na hledání řešení/kořenů můžu použít různé metody, nabízí se třeba Newtonova metoda (viz MA3)
* Reformulace:  proč?

# Audio inpainting revisited and reweighted

* Ve článku se řeší problém obnovy chybějících vzorků v digitálním signálu
* Jedny z prvních metod jsou založeny na autoregresivním modelování (AR)
  + AR se probíralo v signálech na magistru…
  + Chybějící vzorky se doplňují na základě lineární predikce vzorků z okolí
  + Pro dlouhé mezery nad 100ms selhávají a nahrazují je sinusoidální modely nebo neuronové sítě
* Článek se zaměřuje hlavně na mezeru do 50ms a na problémy ztráty energie signálu při doplňování
* **Autogregresivní model**
  + Předpovídá hodnotu signálu na základě jeho minulých hodnot
  + V inpaintingu tedy vytváří chybějící vzorky na základě vzorků, které obklopují tu chybějící část
* **“Weighting signal coeficients”**
  + Tim se myslí to, že sinál má různý části 🡪 na některý se dá spolehnou a na jiný ne
  + Já proto ty míň spolehlivý penalizuju – *jako třeba tranzientní a ruchový části? Nebo se to dělá nějak analýzou spektra?*
* **Analytický model**
  + Nejprve se analyzuje přímo signál 🡪 například pomocí nějaké transformace (Fourierky) a pak se modeluje tak, tak aby měl co nejmíň nenulových koeficientů
  + Místo rekonstrukce koeficientů se tedy hledá signál, jehož koeficienty splňují nějaká kritéria (třeba tu stokrát omýlanou řídkost)
* **Evaluační studie**
  + Je výzkum, který nevyvíjí nový model, ale hodnotí a srovnává různé metody nebo modely – v tomhle případě ty používané pro impanting
* ***V tomto článku se tvrdí, že neexistuje žádný článek, který by používal analytický model pro audio inpainting***
* Gaborova transformace = STFT
* V tomto článku se používá časové okno, které je frekvenčně modulované, tzn to asi nebude klasickej Hamming nebo Hann…
* **Model-based**
  + Znamená, že metoda je je založená na nějakým matematickým modelu (vzorci) k popisu a řešení
  + Může to bejt třeba autoregrese
  + Alternativou k model based jsou data-driven metody, které se nespoléhají na předem zadaný matematický model, ale využívají data k učení
    - Machine Learning a Deep Learning
    - Empirické metody
    - Non-parametrické metody
  + Model-based metody jsou úspěšný pro malý úseky data
* Na základě model-based metod vzniká optimizační problém, který má data fitting term a regularizer.
* V článku popisují, že největší problém je pokles energie v místě, kde se vzorky doplňují

***Obsah obrázku text, řada/pruh

Popis byl vytvořen automaticky***

* Nejsem si jistej, jestli frekvenční modulace v článku popisovaná je prostě další krok navíc nebo se jen prostě obšírně okecává Fourierka
* Neni tohle náhodou zaručeno vžycky?



* **g** je moje okno a  z něj vznikne tak, že z g **odstranim nulové vzorky**
  + nefunguje to tak, že si dělam ze signálu samotnýho slovník? A nulový části odstraňuju abych vyhodil ticho?
* Obsah obrázku text, Písmo, snímek obrazovky, řada/pruh

  Popis byl vytvořen automaticky
  + 🡪 důvod, proč dochází k poklesu amplitudy
* AIa (analyzation) 🡪 výstupem je přímo optimalizovaný signál
* AIs (synthesis) 🡪 výstpem je signál získaný přímo aplikací D
* Gabor atoms
  + Jsou to pravděpodobně jednotlivá okna frekvenčně modulovaná
  + Okna, který jsou blízko/překrývaj se s ,,dírou” mají menší váhu
* Fourierova báze
  + Lze to chápat jako obdobu báze z lineární algebry
  + Jsou to sinusové vlny o různé frekvenci, ze kterých lze složit výsledný signál
  + Tzn když mam Fvz=44100Hz a signál dlouhý 1024 vzorků, pak moje Fourierka má rozlišení 43,06 Hz bázi o frekvenčních složkách 0; 43,06; 86,13Hz…
* Gaborova báze
  + Vychází ze STFT
  + Kromě frekvenční složky mam informaci i o rozložení v čase, protože se každý úsek dělí po časových oknech
  + Ve článku se píše o frekvenční modulaci… proč?