Technische Universität München

Fakultät für Informatik

Master's Thesis in Informatik

An Email-Centered Approach to Intelligent Task Management Using Crowdsourcing and Natural Language Processing

John Doe

figures/infologo.jpg

Technische Universität München

Fakultät für Informatik

Master's Thesis in Informatik

An Email-Centered Approach to Intelligent Task Management Using Crowdsourcing and Natural Language Processing

Ein Email-basierter Ansatz für intelligente Aufgabenverwaltung mit Hilfe von Crowdsourcing und Natural Language Processing

Author: John Doe

Supervisor: Prof. Dr. Johann Schlichter

Advisor: Dr. Wolfgang Wörndl

Submission: DD.MM.YYYY

I assure the single handed composition of this master's thesis only supported by declared resources.

München, DD.MM.YYYY

(John Doe)

Abstract

English abstract.

Inhaltsangabe

Deutsches Abstract.

Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

1.2 Contributions

Chapter 2

Channel model

C:/Users/Kevin/Bachelarbeit/Bachelorarbeit/01_Bachelorarbeit_LaTex/02_Figures/Channelmo

Figure 2.1: Channel model for general Transmitter/Receiver Chain

First a short introduction of the system for all simulations will be given (Fig. 2.1). A few crucial parts of channel blocks are needed for every communication link and a few blocks are added for improvement in performance. All these blocks were chosen in direct benefit to a LDPC transmission of code words to make the simulations as simple and efficient as possible. The link is built up of three main blocks: The transmitter,channel and receiver. With the transmitter handling the creation of random code words, coding with low density parity check (LDPC) and mapping in different modulation schemes. The channel will simulate any incoming and existing noise, e.g., additive white gaussian noise (AWGN). In the end, our receiver will demap and decode our transmitted symbols and

compare the decoded bit stream with the initially created code word.
All single channel blocks will be explained shortly in the following sections.

2.1 Encoder/Decoder

There are many ways to make our transmission more stable and less error prone. A major role in this protection plays the encoder and its counterpart the decoder. Encoder/decoder come in many different forms taking form in hardware coder and also as software coder. They reach from simple linear block codes to more complex convolutional coding and so called turbo codes. It is also important to note, that codes working well in a AWGN-channel will often not have the same performance in a fading channel. A further look will be taken into LDPC codes and based on these, the WiMax code according to the standard IEEE 802.16e. **citation**. While LDPC was mainly ignored in the past, in the 1990's the introduction of turbo codes and an sharp increase in computing power helped the recognition of these forms of decoding.
LDPC-codes are linear block codes with a particular structure for their parity check matrix [$\mathbf{H}$]. In the case of LDPC-codes $\mathbf{H}$ has a small amount of nonzero entries, which means that there is a low density in the parity check matrix. Another important difference in LDPC to turbo codes is the complexity of encoding and decoding. While turbo codes have low complexity in encoding they have high complexity in decoding. The total opposite can be said about LDPC with high complexity in encoding and low complexity in decoding. Another advantage of LDPC is the ability of self correction after decoding with the help of decoding algorithms and the parity check matrix.
WiMax IEEE 802.16e is a standard code model used in small and medium distances in urban areas, which fits the simulations quite well. With WiMax there are different given blocksizes ranging from 576 codewords up to 2304 code words. Code rates are also set in WiMax, which are the following: 1/2, 2/3, 3/4, and 5/6. While there are also two different classes of encoding (A/B), only encoding class A will be used in this setup.

2.2 Bit interleave/Deinterleaver

While the above mentioned coder LDPC (Chapter 2.1) works really well for an AWGN channel this is not always the case in a fading channel. This is where the next important channel block comes into play. To guarantee a stable performance the method of interleaving will be introduced. Interleaving will handle a major problem in fading channels, the appearance of burst errors mainly caused by deep fading over a set time. While LDPC has the ability to correct single code errors it is usually not able to correct a stream of errors. With the interleaver code words will be shuffled into a new random Gaussian distributed code word, which will be passed through our channel. At the receiver a restoration of the shuffled codeword back into its initial state will take place.

`C:/Users/Kevin/Bachelarbeit/Bachelorarbeit/01_Bachelorarbeit_LaTex/02_Figure`

Figure 2.2: Example for interleaving

As clearly seen in figure 2.2 the interleaver will not remove any errors but will prevent or at least mitigate the presence of burst errors. Single errors can be corrected by the LDPC algorithm again. There are two main methods of interleaving today: symbol-interleaved coded modulation (SICM) will interleave our symbols after the modulator while bit-interleaved coded modulation (BICM) will interleave the single bits before modulation is done. BICM will also be used in this thesis for its dominant position in practical communication systems nowadays. **cite**

2.3 Mapper/Demapper

In the mapper, also called modulator, it is possible to compress the codeword into a set sequence of symbols. Group of bits are taken from the bit stream to combine them into specific constellation points. These constellation points or symbols are located in a real/imaginary plane, also called Inphase/Quadrature Plane (I/Q-Planes). The magnitude of the signal can be pinpointed by the distance from the constellation point from the axis nullpoint. Also the same can be done for calculating the phase shift of the constellation point.

add I/Q plane with description of magnitude and phase

There are many forms of modulation schemes, with the most common ones being M-phase shift keying (PSK), M-frequency shift keying (FSK), M-amplitude modulation (AM) and M-quadrature amplitude modulation (QAM). For the simulation a further look will be taken at quadrature phase shift keying (QPSK), 16-QAM and 64-QAM, which are depicted below (Fig. 2.3). All three modulations share the common fact of being differential, which means that the symbols are located in both the real and imaginary plane. One important aspect of differential modulation is the requirement of coherent demodulation, which means that the transmitter and receiver must have matched phase ϕ. In the simulation it will be assumed that perfect phase match is guaranteed between those two. If not, a phase recovery has to be done.

C:/Users/Kevin/Bachelarbeit/Bachelorarbeit/01_Bachelorarbeit_LaTex/02_Figures/Modulat

Figure 2.3: Modulation in I/Q planes for QPSK, 16-QAM and 64-QAM

With QPSK the symbols all share the same amplitude and only differ in their respective phase angle. With the information entropy $S = log_2(M)$**cite** the maximum number of bits can be identified which are assigned in every symbol, with M being the number of symbols in the modulation scheme. So for QPSK the number of bits per symbol amounts to 2.

With M-QAM the phase shift already implemented in QPSK will be used and also a differentiation by magnitude of the signal will be added. So for QAM, signals which differ in their phase shift and also their amplitude, will be send. For 16-QAM a maximum of 4 bits per symbols and for 64-QAM 6 bits per symbol can be achievied.

The modulation schemes make it possible to increase the rate/speed of transmission and are used for any kind of practical communication link. In a practical case modulation will protect the signal from outside noise and interferences, e.g., other mobile hand-held devices, GPS-signals or Wi-Fi signals. It can also increase the range of communication by transmitting over higher carrier frequencies.

2.4 Channel

The channel can be modified in many different ways. Various sources of noise or fading can be applied, which relate to real world interferences. Some interferences experienced in real life transmission are, e.g., thermal noise, distance fading, doppler effect and reflection of signals. To approach those kind of interferences there are many different channel models in simulations, like an AWGN-Channel or Rayleigh/Rician fading. A further look into the AWGN-Channel and the Rayleigh fading will be given. A small graphic will further illustrate the usual culprits for degradation of signal power and resulting loss in communication performance (Fig. 2.4).

C:/Users/Kevin/Bachelarbeit/Bachelorarbeit/01_Bachelorarbeit_LaTex/02_Figures/reflect

Figure 2.4: Interferences in a normal transmission between two devices

Ausführen, beispiele, bilder

2.4.1 AWGN-Channel

Lookup math

The easiest kind of channel manipulation is to add random Gaussian noise to the channel, also commonly known as an AWGN-Channel. Like the name says we will add noise, which is a random Gaussian distribution with flat spectral density, to an existing transmitted signal. Our channel will receive a signal like this:

$$Y = X + N, \tag{2.1}$$

with Y being the received signal, X the send codeword and N the AWGN noise.
The probability density function is defined as follows:

$$f(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} * e^{-\frac{(x-\mu)^2}{2\mu}}, \tag{2.2}$$

with x being the acquired point, μ being the mean or expectation of the distribution and σ^2 the variance of the distribution.
More or less every communication link will have some kind of gaussian noise interference, so the AWGN-channel will be added to every simulation run. Below (Fig. 2.5) a depiction of the spectral power distribution of AWGN. It can be clearly seen that it is flat and spread evenly over the whole spectrum.

C:/Users/Kevin/Bachelarbeit/Bachelorarbeit/01_Bachelorarbeit_LaTex/02_Figures/AWGN.png

Figure 2.5: Power spectral density for a AWGN channel

Add picture of AWGN, pdf or distribution

2.4.2 Rayleigh-Channel

Lookup math

Another common channel model used in communication theory is Rayleigh fading. Rayleigh fading is used to simulate multi-path reception, which means that for a receiver antenna in a wireless link there are many reflected and scattered signals reaching it (Fig. 2.4). These kind of reflections are common for high density urban areas. This results into construction or destruction of waves. Rayleigh distribution can be defined like this:

$$H = \sqrt{X^2 + Y^2}, \tag{2.3}$$

with X and Y being two independent Gaussian distributed random variables. This leads our channel to look like this:

$$Y = H * X + N, \tag{2.4}$$

with the newly added fading coefficient H.
Further calculations will lead to the following pdf:

$$f(x\sigma) = \frac{1}{\sigma^2} e^{-\frac{x^2}{2\sigma^2}}, \tag{2.5}$$

this time only with sigma being the variance of the pdf. The graphic (Fig. 2.6) shows the power distribution over 12000 samples. Being Gaussian randomly distributed there are now these so called "deep fadings" where the power of the fading drops, which will also decrease the signal power of the received signal drop significantly. This results in the so called burst errors, which was mentioned in chapter 2.2.

C:/Users/Kevin/Bachelarbeit/Bachelorarbeit/01_Bachelorarbeit_LaTex/02_Figures/rayleigh

Figure 2.6: Power spectral density for a rayleigh channel

Add pictures, spreading, reflection...

Chapter 3

Capacity in an AWGN channel

In this chapter simulation will be done to calculate the capacity of a communication link with the added noise of AWGN (Ch. 2.4).

3.1 Capacity and Monte-Carlo-Simulation

In general capacity C can be defined as the rate R at which information can be reliably transmitted over a channel, which means as long $R \leq$ C we can achieve a transmission without errors even with noise. All the capacities looked at will be for complex channel models.

For a AWGN-Channel a simple channel model will be used defined by:

$$Y = X + N, \tag{3.1}$$

with X $\sim$ $N(0,{\sigma_{\mathrm{T}}}^2)$ and N $\sim$ $N(0,1)$. The received signal Y will have a distribution of Y $\sim$ $N(0,sigma^2{+}1)$ under the condition that X and N being independently distributed. We will calculate the capacity as a maximum of mutual information I between X and Y:

$$C = max(I(X;Y)), \tag{3.2}$$

with X and Y being to independent randomly normal distributed variables. With the maximum mutual information we calculate the maximum information we can achieve with the given parameters, like modulation, encoding, channel.
For the mutual information further calculations will lead to the differential entropy:

$$I(X;Y) = h(Y) - h(Y|X) \tag{3.3}$$

With differential entropy being defined as:

$$h(Y) = \int p(y) * [-log(p(y)]\mathrm{d}x \tag{3.4}$$

We will now apply the Monte-Carlo-Simulation (MC) to turn our integral into an addition. The Monte-Carlo-simulation will be further explained in the following chapter (Ch. 3.1.2).

$$h(Y) = h(X+N) = log(\pi * e^{\sigma^2+1}) \quad \text{and} \quad h(Y|X) = h(N) = log(\pi * e^1) \tag{3.5}$$

Further calculations will lead us to the final equation for the capacity in an AWGN-channel:

$$C = log(1 + \frac{\sigma^2}{N}) \tag{3.6}$$

With this approach a good approximation of values for further calculations with added modulation schemes has been given. It is stated that only for AWGN the capacity is at his maximum, there should be no capacity value over the calculated one here.

3.1.1 Approach in Matlab

The above mentioned formula 2.4 will be simply implemented in MATLAB. With our noise being randomly distributed around 1 our formula simplifies even more into:

$$C_{\mathrm{AWGN}} = log(1 + SNR) \tag{3.7}$$

The SNR here must be transformed into power and not in decibel. It should be noted that the 0.5 factor in the equation falls away, because of the channel being complex.

3.1.2 Monte-Carlo-Simulation

Monte Carlo Simulation is widely used in stochastic to get solutions for random experiments. It is used to solve analytically unsolvable problems numerically. MC is based upon the law of large numbers, which says that a large number of performing the same experiment will lead the average of the results close to the expected value. We take this as a base to get reliable results. The Monte Carlo simulations will be used for two calculations, once already used above for calculating the differential entropy and later once to calculate a theoretical Rayleigh fading curve out of AWGN.

3.2 Capacity for QPSK and M-QAM

Now the capacity will be calculated for the three above mentioned modulation schemes (Ch. section 2.3). We will implement these schemes into our capacity calculations in an AWGN channel.

3.2.1 QPSK

For QPSK we will have 4 symbols and resulting 2 bits per symbol. Before any simulation or calculation were run it can already be guaranteed, that we will not pass the upper bound of 2 bits/Symbol. So the plot will approach the 2 bits/Symbol for high SNR. After creating a random codeword modulated with the fitting modulation scheme, noise is added to the signal, which is than received as the bit stream Y. The next step to calculate the capacity differs from above.
It is known that the signal is normal random distributed variables and the differential entropy for h(Y) has to be calculated, which is:

$$h(Y) = \sum_{n=0}^{N} (-log(p(y_{\mathrm{n}}))) * \frac{1}{N} \tag{3.8}$$

with p(y) being the probability of y for a normal distributed variable and N the code length.

$$p(y) = \frac{1}{n * \pi} * \sum_{i=1}^{n} (e^{y - x_{\mathrm{i}}}) \tag{3.9}$$

In this equation the number of n needs to be carefully selected. For QPSK we have a n = 4, 16-QAM n = 16 and 64-QAM n = 64.

With the above established equation 3.9 the simulations for QAM are done the same as QPSK, only differing in the number of symbols n.

3.2.2 Results

`C:/Users/Kevin/Bachelarbeit/Bachelorarbeit/01_Bachelorarbeit_LaTex/02_Figure`

Figure 3.1: Capacity plot for general AWGN-channel, QPSK, 16-QAM and 64-QAM

The results of the calculation in MATLAB can be seen above. We can clearly see the modulated channels approach the desired bit/symbol in a good SNR to bit/symbol rate. The Gaussian channel clearly outperforms the modulated channels, clearly seen after 0 dB SNR.

Compare with book capactiy!

Chapter 4

Transmitter Receiver Chain in MATLAB

We will now focus in creating a functioning Transmitter-Receiver chain to simulate a wireless communication as real as possible. The blocks for the communication link were shortly introduced in the beginning, but will be explained further in the following chapters. With LDPC WiMax we use a common communication protocol, which simulates a real channel quite well. Furthermore we will use soft mapping to reconstruct our symbols not hard decoding. Later on I will explain my reasoning behind it.

4.1 LDPC and the CML Library

With a given codeword x of length n and a generator matrix $G = [I^T|P]$. The parity check matrix $\mathbf{H}$ can now be derived as $\mathbf{H} = [-P^T|I_{\text{n-k}}]$. With the parity check matrix $\mathbf{H}$ and a code $\mathbf{C} = xG$ the condition for $c\mathbf{H}^T = 0$ must be fulfilled for the codeword to be valid.
Also with a parity check matrix error correction can be done, that means for single errors we get in our codeword the parity check matrix can selfcorrect our code. This whole process in MATLAB can be computed with the help of the Coded Modulation Library (CML). For this we have the given function "*InitializeWiMaxLDPC*" to create the parity-check.matrix, "*LdpcEncode*" and "*MpDecode*" to encode and decode our codeword. We decided on a length of 2304, the maximum length that can be send, and self correcting for 50 iterations to be sure to correct as many errors as possible that we receive at the end of our communication chain.

4.2 Soft-demapping vs. Hard-demapping

These two approaches will result in rather different result in any kind of simulation. We will have a look in both approaches and will compare their unique advantages and disadvantages. For hard-demapping a received symbol is compared to a given fixed threshhold. At every sampling instant the receiver will decide the state of the bit, either "0" or "1". Hard-demapping uses the minumum Hamming distance to make a decision, which means that bitrow from our receiver is compared to every available constellation point. For every bitdifference between bitrow and constellation point will add to the Hamming distance. In the end the receiver will make a decision by taking the constellation symbol which compared to the created bitrow resembles the most, that means the one with the lowest Hamming distance.
Major difference to hard-demapping the soft-demapping will use the euclidean distance to make a decision. It will use additional informations supplied by us to make a decision. While hard demapping has no info about the reliability of the receivers decision, soft demapping will gives us exactly this. With the eucladian distance we calculate the distance between received symbol to every constellation point. Furthermore we will use the loglikelihood ratio to calculate the reliability with the euclidian distance. While hard-demapping is fast and easy to implement in a system it gives us no reliability and as good of performance as soft demapping. In the end it is a decision based on a balance of computing complexity and performance gain. With many modern systems achieving great computing capability and our desire to create a channel as good as possible we will decide to use soft-demapping. In the next section we will have a further look into soft-demapping and LLR based on a QPSK example.

`C:/Users/Kevin/Bachelarbeit/Bachelorarbeit/01_Bachelorarbeit_LaTex/02_Figure`

Figure 4.1: Capacity plot for general AWGN-channel, QPSK, 16-QAM and 64-QAM

With QPSK we have 4 different symbol constellation also depicted above: (0,0),(0,1),(1,1) and (1,0). The loglikelihood ratio is defined as below

$$L^n = log\frac{P(Y|B_1 = 0)}{P(Y|B_1 = 1)} = log\frac{P(Y|X_1) + P(Y|X_2)}{P(Y|X_3) + P(Y|X_4)} \tag{4.1}$$

add figure of constellation points with differentiation

C:/Users/Kevin/Bachelarbeit/Bachelorarbeit/01_Bachelorarbeit_LaTex/02_Figures/llr.png

Figure 4.2: Figure of QPSK constellation with decision bounds

In the figure 4.2 above it can clearly seen that one symbol consists of two bits. Every two bit constellation possible: (00,01,10,11) are assigned a specific constellation point. For assigning the received points back to a symbol and the associated bits decision bounds and the euclidian distance is needed. We will start with the equation:

$$L(n) = log\frac{P(B1 = 0|Y)}{P(B1 = 1|Y)}, \tag{4.2}$$

which simplifies to the following:

$$L(n) = log\frac{P(Y|X1) + P(Y|X2)}{P(Y|X3) + P(Y|X4)} \tag{4.3}$$

With the log likelihood ratio the most likely constellation point can be pinpointed down.

4.2.1 Results

After receiving our demodulated symbols we can compare those to our codeword we initially send. With this we will determine the frame errors we got for the whole transmission. A frame is defined as a whole codeword length, that means for us it is 2304 bits sent. We have to simulate at least 100000 of those codewords to receive a reliable error rate. Our error rate $= \frac{\text{frame errors}}{\text{number of frames sent}}$. With 100 errors being a reliable number we can also prematurely interrupt our simulation after 100 errors to save simulation time.

4.3 FER and comparison with capacity plots

Add FER points with respective capacity plots

Chapter 5

Communication link for Rayleigh fading channels

Mention slowfading We will do the same as before but also add the fading coefficient H to our channel. The fading coefficient is represented by rayleigh fading, which was introduced in chapter 2.1... For fading our received signal changes in this way:

$$Y = \sqrt{\sigma^2} * H * X + N \tag{5.1}$$

With the fading being unknown to our receiver we need a way to extract or estimate the fading coefficient in the channel. An efficient and easy approach to this it to insert a pilot symbol $\mathrm{X_p}$ before the transmission. We also will divide our whole codeword in single blocks T which will range for blocksize equal to one symbol up to the whole codeword being one codeword. For every block we will insert one pilot symbol at the beginning.

Graphic for block + pilot

Our pilot symbol will have the default value of 1, which is also known at the receiver side. This means to estimate the fading we will do this:

$$Y_p = \sqrt{\sigma^2} * H * X_p + N \tag{5.2}$$

which leads to:

$$H_{\text{est}} = \frac{Y_p + N}{\sqrt{\sigma^2} * X_p} \tag{5.3}$$

With this we get a proper estimation for the fading coefficient, but its estimation is highly dependable of the strength of fading and SNR. With higher SNR we receive better estimation not disturbed by the noise as much. And with lesser fading, close to 1, we do not receive a weakened signal, which is hard to distinguish from the noise.

Maybe add graphics which shows the single scenarios

With the estimated fading coefficient the symbols can be reconstructed.

$$Y_{\text{est}} = \frac{H}{H_{\text{est}}} * \sqrt{\sigma^2} * X + \frac{N}{H_{\text{est}}} \tag{5.4}$$

and with H_{est} being close to H we get

$$Y_{\text{est}} = \sqrt{\sigma^2} * X + \frac{N}{H_{\text{est}}} \tag{5.5}$$

which can be used to calculate the log likelihood ratio.

Maybe add the scatterplot with and w/ rayleigh fading

5.1 Theoratical rayleigh fading FER constructed out of AWGN-Channel

For a proper simulation we will need a reference to compare our simulation results to. For this we will construct a theoratical FER plot for rayleigh fading out of the AWGN-channel. First step is the simulate the AWGN channel with the desired codelength over our SNR. In the previous chapters we have already proven that our simulations are match the theoratical curves. With the simulation for AWGN finished we can now start creating many random value (here n = 10000) rayleigh fading coeffiecients. For every single step of SNR, one step being one SNR, we will compute the SNR after rayleigh fading is created, that means $\text{SNR}_{=}$ SNR * H. The SNR for the fading has a corresponding FER-value which we will be add up and divide by the number of fading coefficients.

$$FER - with - SNRi = \sum_{k=0}^{n} FER(SNRi * H(k)) \tag{5.6}$$

Add plot from AWGN

It can seen that for the AWGN channel we will reach the error floor really fast at around 3 SNR. For any snr-value which was not simulated for, here only for 0 - 5 SNR, we will add virtual value. While the frame error rate of the AWGN channel reaches a minumum value for high SNR it will never reach 0. In our case we will calculate the theoratical rayleigh FER with error floor 0 and once with error floor 10^{-6}, just to show the drastic difference in performance with different error floors. Later on we will also prove that the assumend error floor of 10^{-6} comes close to the real error floor.

Plot for FER with error floor 0 and 10^{-6}

5.2 Rayleigh fading FER with AWGN channel

Add different plots 1. Simulation with perfect channel knowledge 2. Simulation with estimated coefficient 3. Different blocksizes T=N/2, T=N/16 Explain difference and why? We can clearly see a distinct performance difference between the two error floors. OH WOW! SURPRISE!

5.3 Increase power of pilot symbol

5.4 Increase of pilot symbols in one block

5.5 Results and comparison with AWGN channel

Chapter 6

Conclusion

6.1 Comparison between fading and AWGN channel

6.2 Fazit